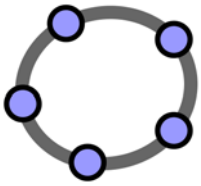
GeoGebra Help

Official Manual 3.2



Markus Hohenwarter and Judith Hohenwarter
www.geogebra.org

GeoGebra Help 3.2

Last modified: April 22, 2009

Authors
Markus Hohenwarter, markus@geogebra.org
Judith Hohenwarter, judith@geogebra.org

GeoGebra Online: http://www.geogebra.org

Contents

1. WHAT IS GEOGEBRA? 6

1.1. Multiple Views for Mathematical Objects 6
1.1.1. Graphics View 6
1.1.2. Algebra View 7
1.1.3. Spreadsheet View 8

1.2. GeoGebra as a Tool for Teaching and Learning Mathematics 8
1.2.1. Customizing the User Interface 8
1.2.2. The Properties Dialog 10
1.2.3. The Context Menu 10

1.3. GeoGebra as a Presentation Tool 11
1.3.1. The Navigation Bar 11
1.3.2. The Construction Protocol 11
1.3.3. Customize the Settings 12

1.4. GeoGebra as an Authoring Tool 13
1.4.1. Printing Options 13
1.4.2. Creating Pictures of the *Graphics View* 13
1.4.3. Creating Interactive Webpages 14

2. GEOMETRIC INPUT 16

2.1. General Notes 16

2.2. Construction Tools 16
2.2.1. General Tools 17
2.2.2. Point Tools 18
2.2.3. Vector Tools 19
2.2.4. Segment Tools 19
2.2.5. Ray Tool 20
2.2.6. Polygon Tools 20
2.2.7. Line Tools 20
2.2.8. Conic Section Tools 21
2.2.9. Arc and Sector Tools 22
2.2.10. Number and Angle Tools 23
2.2.11. Boolean Variable Tool 25
2.2.12. Locus Tool 25
2.2.13. Geometric Transformation Tools 25
2.2.14. Text Tool 26
2.2.15. Image Tool 28

3. ALGEBRAIC INPUT 30

3.1. General Notes 30

3.2. Direct Input 32

3.2.1. Numbers and Angles 32
3.2.2. Points and Vectors 33
3.2.3. Lines and Axes 33
3.2.4. Conic Sections 34
3.2.5. Functions of *x* 34
3.2.6. Pre-defined Functions and Operations 35
3.2.7. Boolean Variables and Operations 36
3.2.8. List Objects and Operations 37
3.2.9. Matrix Objects and Operations 38
3.2.10. Complex Numbers and Operations 38

3.3. Commands 39
3.3.1. General Commands 40
3.3.2. Boolean Commands 40
3.3.3. Number Commands 41
3.3.4. Angle Command 45
3.3.5. Point Commands 45
3.3.6. Vector Commands 47
3.3.7. Segment Command 48
3.3.8. Ray Command 48
3.3.9. Polygon Command 49
3.3.10. Line Commands 49
3.3.11. Conic Section Commands 51
3.3.12. Function Commands 52
3.3.13. Parametric Curve Command 53
3.3.14. Arc and Sector Commands 54
3.3.15. Text Commands 55
3.3.16. Locus Command 58
3.3.17. List and Sequence Commands 58
3.3.18. Geometric Transformation Commands 61
3.3.19. Statistics Commands 63
3.3.20. Spreadsheet Commands 67
3.3.21. Matrix Commands 67

4. MENU ITEMS 69

4.1. File Menu 69

4.2. Edit Menu 71

4.3. View Menu 73

4.4. Options Menu 74

4.5. Tools Menu 76

4.6. Window Menu 77

4.7. Help Menu 77

5. SPECIAL GEOGEBRA FEATURES 79

5.1. Animation 79
5.1.1. Automatic Animation 79
5.1.2. Manual Animation 79

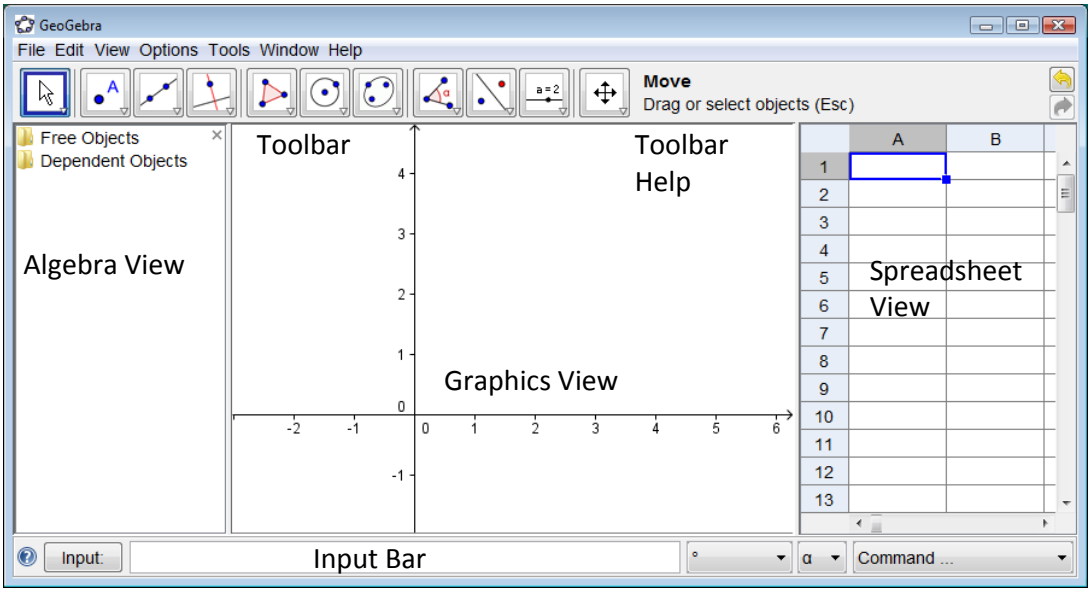
5.2. Conditional Visibility 80
5.3. User Defined Tools 81
5.4. Dynamic Colors 82
5.5. JavaScript Interface 82
5.6. Keyboard Shortcuts 83
5.7. Labels and Captions 87
5.8. Layers 87
5.9. Redefine 88
5.10. Trace and Locus 88
6. INDEX 90

1. What is GeoGebra?

GeoGebra is dynamic mathematics software that joins geometry, algebra and calculus. It is developed for learning and teaching mathematics in schools by Markus Hohenwarter and an international team of programmers.

1.1. Multiple Views for Mathematical Objects

GeoGebra provides three different views of mathematical objects: a *Graphics View*, a, numeric *Algebra View*, and a *Spreadsheet View*. They allow you to display mathematical objects in three different representations: graphically (e. g., points, function graphs), algebraically (e. g., coordinates of points, equations), and in spreadsheet cells. Thereby, all representations of the same object are linked dynamically and adapt automatically to changes made to any of the representations, no matter how they were initially created.



1.1.1. Graphics View

Using the *construction tools* available in the *Toolbar* you can do geometric constructions in the *Graphics View* with the mouse. Select any construction tool from the *Toolbar* and read the *Toolbar Help* (next to the *Toolbar*) in order to find out how to use the selected tool. Any object you create in the *Graphics View* also has an algebraic representation in the *Algebra View*.

Note: After activating the tool *Move* you are able to *move objects in the Graphics View* by dragging them with the mouse. At the same time, their algebraic representations are dynamically updated in the *Algebra View*.

Every icon in the *Toolbar* represents a toolbox that contains a selection of similar construction tools. In order to open a toolbox, you need to click on the small arrow in the lower right corner of the *Toolbar* icon.

Hint: Construction tools are organized by the nature of resulting objects or the functionality of the tools. You will find tools that create different types of points in the *Point Toolbox* (default icon) and tools that allow you to apply geometric transformations in the *Transformation Toolbox* (default icon).

1.1.2. Algebra View

Using the *Input Bar* you can directly enter algebraic expressions in GeoGebra. After hitting the *Enter*-key your algebraic input appears in the *Algebra View* while its graphical representation is automatically displayed in the *Graphics View*.
Example: The input `f(x) = x^2` gives you the function *f* in the *Algebra View* and its function graph in the *Graphics View*.

In the *Algebra View*, mathematical objects are organized as *free* and *dependent objects*. If you create a new object without using any other existing objects, it is classified as a *free object*. If your newly created object was created by using other existing objects, it is classified as a *dependent object*.
Hint: If you want to hide the algebraic representation of an object in the *Algebra View*, you may specify the object as an *auxiliary object*: Right click (Mac OS: *Ctrl*-click) on the corresponding object in the *Algebra View* and select *Properties* from the appearing *Context Menu*. On tab *Basic* of the *Properties Dialog* you may specify the object as an *Auxiliary Object*. By default, *auxiliary objects* are not shown in the *Algebra View*, but you can change this setting by selecting the item *Auxiliary Objects* from the *View* menu.

Note that you are able to modify objects in the *Algebra View* as well: Make sure that you activate the *Move* tool before you double click on a *free object* in the *Algebra View*. In the appearing text box you can directly edit the algebraic representation of the object. After hitting the *Enter*-key, the graphical representation of the object will automatically adapt to your changes.
If you double click on a *dependent object* in the Algebra View, a dialog window appears allowing you to *Redefine* the object.

GeoGebra also offers a wide range of *commands* that can be entered into the *Input Bar*. You can open the *list of commands* in the right corner of the *Input Bar* by clicking on the button *Command*. After selecting a command from this list (or typing its name directly into the *Input Bar*) you can press the *F1*-key to get information about the syntax and arguments required to apply the corresponding command.

1.1.3. Spreadsheet View

In GeoGebra's *Spreadsheet View* every cell has a specific name that allows you to directly address each cell. For example, the cell in column *A* and row 1 is named *A1*.
Note: These cell names can be used in expressions and commands in order to address the content of the corresponding cell.

In the *spreadsheet cells* you can enter not only numbers, but all types of mathematical objects that are supported by GeoGebra (e. g., coordinates of points, functions, commands). If possible, GeoGebra immediately displays the graphical representation of the object you entered in a *spreadsheet cell* in the *Graphics View* as well. Thereby, the name of the object matches the name of the *spreadsheet cell* used to initially create it (e. g., *A5*, *C1*).

Note: By default, spreadsheet objects are classified as *auxiliary objects* in the *Algebra View*. You can show or hide these *auxiliary objects* by selecting *Auxiliary Objects* from the *View* menu.

1.2. GeoGebra as a Tool for Teaching and Learning Mathematics

1.2.1. Customizing the User Interface

The user interface of GeoGebra can be customized by using the *View* menu. For example, you can hide different parts of the interface (e. g., the *Algebra View*, *Spreadsheet View*, or *Input Bar*) by checking or un-checking the corresponding menu item in the *View* menu.

Showing and Hiding Objects

You may show or hide objects in the *Graphics View* in different ways.

- You may use tool *Show/Hide Object* to show or hide objects.
- Open the *Context Menu* and select item *Show Object* to change the visibility status of the selected object.
- In the *Algebra View*, the icon to the left of every object shows its current visibility state (*shown* or *hidden*). You may directly click on the little marble icon in order to change the visibility status of an object.
- You can also use the tool *Check Box to Show/Hide Objects* in order to show or hide one or several objects.

Customizing the *Graphics View*

In order to adjust the visible part of the *drawing pad* in the *Graphics View*, you can drag the drawing pad by using tool *Move Drawing Pad* and use the following ways of zooming:

- You may use the *tools* *Zoom In* and *Zoom Out* in order to zoom in the *Graphics View*.
  Note: The position of your click determines the center of zoom.
- You may use the *scroll wheel* of your mouse in order to zoom in the *Graphics View*.

- You may use *keyboard shortcuts* to zoom in (Ctrl +) and to zoom out (Ctrl -).
- After right clicking (Mac OS: *Ctrl*-click) on an empty spot on the drawing pad a *Context Menu* appears which allows you to *Zoom*.
- You may specify a *zoom rectangle* by right clicking (Mac OS: *Cmd* - click) on an empty spot in the *Graphics View* and dragging the mouse to the opposite corner of your desired *zoom rectangle*. Release the mouse button in order to finish the *zoom rectangle*, which will then automatically adjust to fill all the space in the *Graphics View*.

You can also show or hide the *coordinate axes* and a *coordinate grid* in the *Graphics View* by using the *View* menu.
Note: Another way of showing or hiding the *axes* and the *grid* is by right clicking (Mac OS: *Ctrl*-click) on the *drawing pad* and selecting the corresponding items *Axes* or *Grid* from the appearing *Context Menu*.

Customizing Coordinate Axes and Grid

The *coordinate axes* and *grid* can be customized using the *Properties Dialog of the Graphics View*. After right clicking (Mac OS: *Ctrl*-click) on the *drawing pad*, you can open this dialog window by selecting *Properties* from the appearing *Context Menu* of the *Graphics View*.

- On **tab *Axes***, you can, for example, change the line style and units of the coordinate axes, and set the distance of the tickmarks to a certain value. Note that you can customize both axes individually, by clicking on tabs *xAxis* or *yAxis*. Furthermore, you can also change the ratio between the axes and hide or show the axes individually.
- On **tab *Grid***, you can, for example, change the color and line style of the coordinate grid, and set the distance for grid lines to a certain value. In addition, you may also set the grid to be *Isometric*.

Note: Scaling the axes is possible in every mode by pressing and holding the *Shift*-key (PC: also *Ctrl*-key) while dragging the axis.

Note: The *Properties Dialog* of the *Graphics View* is different from the *Properties Dialog* for objects.

Customizing the *Toolbar*

The *Toolbar* can be customized by selecting *Customize Toolbar…* from the *Tools* menu. Select the *tool* or *toolbox* you want to remove from the *Toolbar* in the list on the left hand side of the appearing dialog window and click button *Remove >* in order to remove the tool/toolbox from the *Toolbar*.

Note: You can restore the default *Toolbar* by clicking on the button *Restore Default Toolbar* in the left lower corner of the dialog window.

1.2.2. The Properties Dialog

The *Properties Dialog* allows you to modify properties of objects (e. g., size, color, filling, line style, line thickness, visibility).

You can open the *Properties Dialog* in several ways:

- Right click (Mac OS: *Ctrl*-click) on an object and select *Properties…* from the appearing *Context Menu*.
- Select item *Properties* from the *Edit* menu.
- Select the *Move* tool and double click on an object in the *Graphics View*. In the appearing Redefine dialog window, click on the button *Properties…*.

In the *Properties Dialog* objects are organized by types (e. g., points, lines, circles) in the list on the left hand side, which makes it easier to handle large numbers of objects. You need to select one or more objects from this list in order to change its/their properties.
Note: By clicking on a heading in the list of objects (e. g., *Point*) you can select all objects of this type and therefore, quickly change the properties for all these objects.

You can modify the properties of selected objects using the **tabs** on the right hand side (e. g., *Basic*, *Color*, *Style*, *Advanced*).
Note: Depending on the selection of objects in the list, a different set of tabs may be available.

Close the *Properties Dialog* when you are done with changing properties of objects.

1.2.3. The Context Menu

The *Context Menu* provides a quick way to change the behavior or advanced properties of an object. Right click (Mac OS: *Ctrl*-click) on an object in order to open its *Context Menu.* For example, it allows you to change the object's algebraic notation (e. g., polar or Cartesian coordinates, implicit or explicit equation) and to directly access features like *Rename*, *Delete*, *Trace On*, *Animation On*, or *Copy to Input Bar*.

Note: If you open the *Context Menu* for a point in the *Graphics View*, it gives you the option *Trace to Spreadsheet* (only if the *Spreadsheet View* is active). Once selected, this feature allows you to record the coordinates of the point in the *Spreadsheet View* if it is moved.

Note: Selecting *Properties…* in the *Context Menu* opens the *Properties Dialog*, where you can change the properties of all objects used.

1.3. GeoGebra as a Presentation Tool

1.3.1. The Navigation Bar

GeoGebra offers a *Navigation Bar* that allows you to navigate through the construction steps of a prepared GeoGebra file. Select item *Navigation Bar for Construction Steps* in the *View* menu in order to display the *Navigation Bar* at the bottom of the *Graphics View*.

The *Navigation Bar* provides a set of navigation buttons and displays the number of construction steps (e. g., *2 / 7* means that currently the second step of a total of 7 construction steps is displayed):

- button: go back to step 1
- button: go back step by step
- button: go forward step by step
- button: go to the last step
- *Play*: automatically play the construction step by step
  Note: You may change the speed of this automatic play feature using the text box to the right of the *Play* button.
- *Pause*: pause the automatic play feature
  Note: This button only appears after you click on the *Play* button.
- button: This button opens the *Construction Protocol*.

1.3.2. The Construction Protocol

You can access the interactive Construction Protocol by selecting item *Construction Protocol* from the *View* menu. It is a table that shows all construction steps. The Construction Protocol allows you to redo a prepared construction step by step using the *Navigation Bar* at the bottom of the Construction Protocol dialog.

Navigating and Modifying the *Construction Protocol*

You may use the keyboard to navigate in the *Construction Protocol*:

- Use the ↑ *up arrow* of your keyboard to go to the previous construction step.
- Use the ↓ *down arrow* of you keyboard to go to the next construction step.
- Use the *Home* key to go to the beginning of the *Construction Protocol*.
- Use the *End* key to go to the end of the *Construction Protocol*.
- Use the *Delete* key in order to delete the selected construction step.
  Note: This may also affect other objects that depend on the selected object/construction step.

You may also use the mouse in order to navigate in the *Construction Protocol*:

- Double click a row in order to select a construction step.
- Double click the header of any column in order to go to the start of the *Construction Protocol*.

- Drag and drop a row in order to move a construction step to another position in the *Construction Protocol*.
  Note: This is not always possible due to the dependencies between different objects.
- Right click a row in order to open the *Context Menu* for the object of this construction step.

Note: You can insert construction steps at any position. Select the construction step below you would like to insert a new construction step. Leave the *Construction Protocol* window open while you create a new object. This new construction step is immediately inserted into the selected position of the *Construction Protocol*.

Using the column *Breakpoint* in the *View* menu of the *Construction Protocol* window, you are able to define certain construction steps as *Breakpoints*. This allows you to group several objects together. When navigating through your construction using the *Navigation Bar*, groups of objects are shown at the same time.

Note: You may switch the different columns of the *Construction Protocol* on and off by using the *View* menu of the *Construction Protocol* window.

Exporting the *Construction Protocol* as a Webpage

GeoGebra allows you to export the *Construction Protocol* as a webpage. First, you need to open the *Construction Protocol* using the *View* menu. Then, you can open the *File* menu of the appearing *Construction Protocol* window and select item *Export as Webpage.*

In the export window of the *Construction Protocol* you can enter *Title*, *Author*, and a *Date* for the construction and choose whether or not you want to include a picture of the *Graphics View* and the *Algebra View*. In addition, you can also choose to export a *Colorful Construction Protocol*. This means that objects in the *Construction Protocol* will match the color of the corresponding objects in the construction.

Note: The exported HTML file can be viewed with any Internet browser (e. g. Firefox, Internet Explorer) and edited with many text processing systems (e. g. OpenOffice Writer).

1.3.3. Customize the Settings

GeoGebra allows you to change and save settings using the *Options* menu. For example, you may change the *Angle Unit* from *Degree* to *Radians*, or change the *Point Style*, *Checkbox Size*, and *Right Angle Style*. In addition, you may change how *Coordinates* are displayed on screen and which objects are labeled (*Labeling*).
Please see the section about the *Options* menu for more information.

You can save your customized settings by selecting item *Save Settings* from the *Options* menu. After doing so, GeoGebra will remember your customized settings and use them for every new GeoGebra file you create.
Note: You may restore the default settings by selecting *Restore Default Settings* from the *Options* menu.

Note: If you use GeoGebra as a presentation tool, you might want to increase the *Font Size* (*Options* menu) so your audience can easily read text and labels of objects.

1.4. GeoGebra as an Authoring Tool

1.4.1. Printing Options

Printing the *Graphics View*

GeoGebra allows you to print the *Graphics View* of your constructions. You can find the corresponding item *Print Preview* in the *File* menu. In the appearing *Print Preview* dialog window, you can specify the *Title*, *Author*, and a *Date* for the construction. In addition, you can set the *Scale* of your printout (in cm) and change the *Orientation* of the paper used (portrait or landscape).

Note: In order to update the *Print Preview* after you made changes to the text or layout of the printout, you need to press the *Enter*-key.

Printing the Construction Protocol

If you want to print the *Construction Protocol*, you first need to open the *Construction Protocol* dialog window by using the *View* menu. Then, you can open the *Print Preview* window of the *Construction Protocol* from the *File* menu of this new window.
Again, you may enter Title, Author, and a Date or change the Scale or paper Orientation before printing your *Construction Protocol*.

Note: You may switch the different columns *Name*, *Definition*, *Command*, *Algebra*, and *Breakpoint* of the *Construction Protocol* on and off by using the *View* menu of the *Construction Protocol* dialog window.

1.4.2. Creating Pictures of the *Graphics View*

Saving the *Graphics View* as a Picture

You can save the *Graphics View* of your constructions as a picture file on your computer.
Note: The full *Graphics View* will be saved as a picture. If your construction does not use all the available space in the *Graphics View*, you might want to…

- …use tools *Move Drawing Pad*, *Zoom In*, and/or *Zoom Out* in order to place your construction in the upper left corner of the *Graphics View*. Afterwards, you may reduce the size of the GeoGebra window by dragging one of its corners with the mouse.
- … use the *selection rectangle* in order to specify which part of the *Graphics View* should be exported and saved as a picture.

- You may create points called *Export_1* and *Export_2*, which will be used to define diagonally opposite corners of the *export rectangle*.
  Note: Points $Export_1$ and $Export_2$ must be within the visible area of the *Graphics View*.

In the *File* menu, select item *Export* before clicking on item *Graphics View as Picture*. In the appearing dialog window you may specify the *Format*, *Scale* (in cm), and the *Resolution* (in dpi) of the output picture file.
Note: The true size of the exported image is shown at the bottom of the export window just above the buttons, both in centimeters and pixel.

Please find more information about the different picture files available in section Export - *Graphics View* as Picture (png, eps)….

Copying the *Graphics View* to Clipboard

There are different ways of copying the *Graphics View* to your computer's clipboard:

- In the *Edit* menu, you may select item *Graphics View to Clipboard*.
- In the *File* menu, you first need to select item *Export*, before you can click on item *Graphics View to Clipboard*.
- In the *Export Graphics View as Picture* dialog window (menu *File* – *Export* – *Graphics View as Picture (png, eps)…*) you may click on the button *Clipboard*.

This feature copies a screenshot of the *Graphics View* to your system's clipboard as a PNG (see PNG format) picture. This picture can be pasted into other documents (e. g. a word processing document).

Note: In Order to export your construction at a certain scale (in cm) please use the menu item *Graphics View as Picture* in the *File* menu, *Export*.

1.4.3. Creating Interactive Webpages

GeoGebra allows you to create *interactive webpages*, so called *Dynamic Worksheets*, from your files. In the *File* menu, you need to select item *Export* before you can click on item *Dynamic Worksheet as Webpage (html)*. This opens the export dialog window for *Dynamic Worksheets*:

- At the top of the export window you can enter the *Title*, *Author*, and a *Date* for your *Dynamic Worksheet*.
- Tab *General* allows you to add some text above and below the dynamic construction (e. g., a description of the construction and some tasks). You can also determine if the construction itself may be included directly into the webpage or if it can be opened by clicking on a button.
- Tab *Advanced* allows you to change the functionality of the dynamic construction (e. g., show a reset icon, double click should open the GeoGebra application window) as well as to modify the user interface shown in the interactive applet (e. g., show the *Toolbar*, modify height and width).

Note: If the size of your applet is too big to fit on a computer screen with standard resolution (1024 x 768), you may want to resize it before the actual export as a *Dynamic Worksheet*.

Note: Several files are created when you export a *Dynamic Worksheet*:

- *HTML* file (e. g. *circle.html*) – this file includes the worksheet itself
- *GGB* file (e. g. *circle.ggb*) – this file includes your GeoGebra construction
- *JAR* (several files) – these files include GeoGebra and make your worksheet interactive

All these files (e. g. *circle.html*, *circle.ggb* and the *geogebra.jar* files) have to be in one folder (directory) to let the dynamic construction work.

The exported HTML file (e. g. *circle.html*) can be viewed with any Internet browser (e. g. Mozilla, Internet Explorer, Safari). In order to let the dynamic construction work, Java has to be installed on the computer. You can get Java from http://www.java.com without charge. If you want to use your *Dynamic Worksheet* in your school's computer network, ask your local network administrator to install Java on the computers.

Note: You can edit the *Dynamic Worksheet*'s text with many word processing systems (e. g. FrontPage, OpenOffice Writer) by opening the exported HTML file. You may also edit the *Dynamic Worksheet*'s applet by opening the GGB file in GeoGebra and saving it with the same name afterwards.

2. Geometric Input

2.1. General Notes

The *Graphics View* shows the graphical representation of mathematical objects (e. g., points, vectors, segments, polygons, functions, curves, straight lines, conic sections). Whenever the mouse is moved over one of these objects a description appears as a roll-over text and the object is highlighted.

There are several tools/modes to tell GeoGebra how it should react to mouse input in the *Graphics View* (see section Construction Tools). For example, clicking on the drawing pad can create a new point (see tool *New Point*), intersect two objects (see tool *Intersect Two Objects*), or create a circle (see *Circle* tools).

2.2. Construction Tools

The following *construction tools* or modes can be activated by clicking on the buttons of the *Toolbar*. You can click on the small arrow in the lower right corner of an icon to open a T*oolbox* with similar other tools.
Note: With most construction tools you can easily create new points by clicking on empty spaces on the drawing pad.

Selecting Objects

To *select an object* means to click on it with the mouse after selecting the *Move* tool.

If you want to select several objects at the same time, you could draw a *selection rectangle*: Select the *Move* tool and click on the position of the first corner of your desired *selection rectangle*. Hold the left mouse key pressed down and move the pointer to the position of the diagonally opposite corner of your *desired selection* rectangle. After releasing the mouse button, all objects within the *selection rectangle* are selected.

Note: By holding the *Ctrl*-key (Mac OS: *Cmd*-key) while clicking on different objects, you can select several objects at the same time.

Fast Renaming of Objects

To quickly rename a selected or newly created object just start typing to open the *Rename* dialog for this object. Then, type in the new name of the selected object and click on the *OK* button.

2.2.1. General Tools

Copy Visual Style

This tool allows you to copy visual properties (e. g., color, size, line style) from one object to one or more other objects. To do so, first select the object whose properties you want to copy. Then, click on all other objects that should adopt these properties.

Delete Object

Click on any object you want to delete (also see command *Delete*).
Note: You can use the *Undo* button if you accidentally deleted the wrong object.

Move

Drag and drop free objects with the mouse. If you select an object by clicking on it in *Move* mode, you may…

- … delete the object by pressing the *Delete*-key
- … move the object by using the arrow keys (see section Manual Animation)

Note: You can quickly activate the *Move* tool by pressing the *Esc*-key of your keyboard.

Move Drawing Pad

Drag and drop the drawing pad in the *Graphics View* to change its visible area.
Note:

- You can also move the *drawing pad* by pressing the *Shift*-key (MS Windows: also *Ctrl*-key) and dragging it with the mouse in any mode.
- In this mode you can also scale each of the axes by dragging it with the mouse.

Record to Spreadsheet

This tool allows you to move an object and to record a sequence of its values in the *Spreadsheet View*. This tool works for numbers, points, and vectors.
Note: GeoGebra will use the first two empty columns of the *Spreadsheet View* to record the values of the selected objects.

Relation

Select two objects to get information about their relation in a pop-up window (also see command *Relation*).

Rotate around Point

Select the center point of the rotation first. Then, you may rotate *free objects* around this point by dragging them with the mouse (also see command *Rotate*).

Show / Hide Label

Click on an object to show or hide its label.

Show/Hide Object

Select the object you want to show or hide after activating this tool. Then, switch to another tool in order to apply the visibility changes to this object.
Note: When you activate this tool, all objects that should be hidden are displayed in the *Graphics View* highlighted. In this way, you can easily show hidden objects again by deselecting them before switching to another tool.

Zoom In

Click on any place on the drawing pad to zoom in (also see section Customizing the *Graphics View*).
Note: The position of your click determines the center of zoom.

Zoom Out

Click on any place on the drawing pad to zoom out (also see section Customizing the *Graphics View*).
Note: The position of your click determines the center of zoom.

2.2.2. Point Tools

Intersect Two Objects

Intersection points of two objects can be created in two ways (also see command *Intersect*).

- Selecting two objects creates *all intersection points* (if possible).
- Directly clicking on an intersection of the two objects creates only this *single intersection point* .

Note: For segments, rays, or arcs you may specify whether you want to *Allow outlying intersections* on tab *Basic* of the *Properties Dialog*. This can be used to get intersection points that lie on the extension of an object. For example, the extension of a segment or a ray is a straight line.

Midpoint or Center

You may click on either two points or one segment to get its midpoint. You can also click on a conic section (circle or ellipse) in order to create its center point (also see commands *Center* and *Midpoint*).

New Point

Click on the *drawing pad* in the *Graphics View* in order to create a new point. The coordinates of the point are fixed when the mouse button is released.
Note:

- By clicking on a segment, straight line, polygon, conic section, function, or curve you can create a *point on this object* (also see command *Point*).
- Clicking on the intersection of two objects creates this intersection point (also see tool *Intersect Two Objects* and command *Intersect*).

2.2.3. Vector Tools

Vector between Two Points

Select the starting point and then the end point of the vector (also see command *Vector*).

Vector from Point

Select a point *A* and a vector *v* to create the new point *B = A + v* as well as the vector from *A* to *B* (also see command *Vector*).

2.2.4. Segment Tools

Segment between Two Points

Select two points *A* and *B* in order to create a segment between *A* and *B* (also see command *Segment*).
Note: In the *Algebra View*, the segment's length is displayed.

Segment with Given Length from Point

Click on a point *A* that should be the starting point of the segment. Specify the desired length *a* of the segment in the appearing window (also see command *Segment*).
Note: This tool creates a segment with length *a* and endpoint *B* which may be rotated around the starting point *A* by using tool *Move*.

2.2.5. Ray Tool

Ray through Two Points

Selecting two points *A* and *B* creates a ray starting at *A* through *B* (also see command *Ray*).
Note: In the *Algebra View* the equation of the corresponding line is displayed.

2.2.6. Polygon Tools

Polygon

Successively select at least three points which will be the vertices of the polygon. Then, click the first point again in order to close the polygon (also see command *Polygon*).
Note: In the *Algebra View*, the polygon's area is displayed.

Regular Polygon

Select two points *A* and *B* and specify the number *n* of vertices in the text field of the appearing dialog window. This gives you a regular polygon with *n* vertices including points *A* and *B* (also see command *Polygon*).

2.2.7. Line Tools

Angle Bisector

Angle bisectors can be defined in two ways (also see command *AngleBisector*):

- Selecting three points *A*, *B*, and *C* produces the angle bisector of the enclosed angle, where point *B* is the apex.
- Selecting two lines produces their two angle bisectors.

Note: The direction vectors of all angle bisectors have length 1.

Best Fit Line

Create the best fit line for a set of points in the following ways (also see command *FitLine*):

- Create a *selection rectangle* that contains all points.
- Select a *list of points* to create their corresponding best fit line.

Line through Two Points

Selecting two points *A* and *B* creates a straight line through *A* and *B* (also see command *Line*).
Note: The line's direction vector is (*B* - *A*).

Parallel Line

Selecting a line *g* and a point *A* defines a straight line through *A* parallel to *g* (also see command *Line*).
Note: The line's direction is the direction of line *g*.

Perpendicular Bisector

Click on either a segment *s* or two points *A* and *B* in order to create a perpendicular bisector (also see command *PerpendicularBisector*).
Note: The bisector's direction is equivalent to the perpendicular vector of segment *s* or *AB* (also see command *PerpendicularVector*).

Perpendicular Line

Selecting a line *g* and a point *A* creates a straight line through *A* perpendicular to line *g* (also see command *PerpendicularLine*).
Note: The line's direction is equivalent to the perpendicular vector of *g* (also see command *PerpendicularVector*).

Polar or Diameter Line

This tool creates the polar or diameter line of a conic section (also see command *Polar*).

- Select a point and a conic section to get the polar line.
- Select a line or a vector and a conic section to get the diameter line.

Tangents

Tangents to a conic section can be produced in several ways (also see command *Tangent*):

- Selecting a point *A* and a conic *c* produces all tangents through *A* to *c*.
- Selecting a line *g* and a conic *c* produces all tangents to *c* that are parallel to line *g*.
- Selecting a point *A* and a function *f* produces the tangent line to *f* in *x = x(A)*.
  Note: *x(A)* represents the *x*-coordinate of point *A*. If point *A* lies on the function graph, the tangent runs through point *A*.

2.2.8. Conic Section Tools

Circle with Center and Radius

Select the center point *M* and enter the radius in the text field of the appearing dialog window (also see command *Circle*).

Circle with Center through Point

Selecting a point *M* and a point *P* defines a circle with center *M* through *P* (also see command *Circle*).
Note: The circle's radius is the distance *MP*.

Circle through Three Points

Selecting three points *A*, *B*, and *C* defines a circle through these points (also see command *Circle*).
Note: If the three points lie on one straight line, the circle degenerates to this line.

Compass

UK English: Compasses

Select a segment or two points to specify the radius. Then, click on a point that should be the center of the new circle.

Conic through Five Points

Selecting five points produces a conic section through these points (also see command *Conic*).
Note: If four of these five points lie on a line, the conic section is not defined.

Ellipse

Select the two foci of the ellipse. Then, specify a third point that lies on the ellipse (also see command *Ellipse*).

Hyperbola

Select the two foci of the hyperbola. Then, specify a third point that lies on the hyperbola (also see command *Hyperbola*).

Parabola

Select a point and the directrix of the parabola (also see command *Parabola*).

2.2.9. Arc and Sector Tools

Note: In GeoGebra, the algebraic value of an arc is its length. The value of a sector is its area.

Circular Arc with Center between Two Points

First, select the center point *M* of the circular arc. Then, select the starting point *A* of the arc, before you select a point *B* that specifies the length of the arc (also see command CircularArc).
Note: While point *A* always lies on the circular arc, point *B* does not have to lie on it.

Circular Sector with Center between Two Points

First, select the center point *M* of the circular sector. Then, select the starting point *A* of the sector's arc, before you select a point *B* that specifies the length of the sector's arc (also see command *CircularSector*).
Note: While point *A* always lies on the sector's arc, point *B* does not have to lie on it.

Circumcircular Arc through Three Points

Selecting three points *A*, *B*, and *C* creates a circular arc through these points. Thereby, point *A* is the starting point of the arc, point *B* lies on the arc, and point *C* is the endpoint of the arc (also see command *CircumcircularArc*).

Circumcircular Sector through Three Points

Selecting three points *A*, *B*, and *C* creates a circular sector through these points. Thereby, point *A* is the starting point of the sector's arc, point *B* lies on the arc, and point *C* is the endpoint of the sector's arc (also see command *CircumcircularSector*).

Semicircle

Select two points *A* and *B* to create a semicircle above the segment *AB* (also see command *Semicircle*).

2.2.10. Number and Angle Tools

Angle

With this tool you can create angles in different ways (also see command *Angle*):

- Click on three points to create an angle between these points. The second point selected is the vertex of the angle.
- Click on two segments to create the angle between them.
- Click on two lines to create the angle between them.
- Click on two vectors to create the angle between them.

- Click on a polygon to create all angles of this polygon.
  Note: If the polygon was created by selecting its vertices in counter clockwise orientation, the *Angle* tool gives you the interior angles of the polygon.

Note: Angles are created in *counter clockwise* orientation. Therefore, the order of selecting these objects is relevant for the *Angle* tool. If you want to limit the maximum size of an angle to 180°, un-check *Allow Reflex Angle* on tab *Basic* of the *Properties Dialog*.

Angle with Given Size

Select two points *A* and *B* and type the angle's size into the text field of the appearing window (also see command *Angle*).
Note: This tool creates a point *C* and an angle *α*, where *α* is the angle *ABC*.

Area

This tool gives you the area of a polygon, circle, or ellipse as a number and shows a dynamic text in the *Graphics View* (also see command *Area*).

Distance or Length

This tool gives you the distance between two points, two lines, or a point and a line as a number and shows a dynamic text in the *Graphics View*. It can also give you the length of a segment, the circumference of a circle, or the perimeter of a polygon (also see commands *Distance* and *Length*).

Slider

Click on any free place in the *Graphics View* to create a slider for a number or an angle. The appearing dialog window allows you to specify the *Name*, *Interval* [*min, max*], and *Increment* of the number or angle, as well as the *Alignment* and *Width* of the slider (in pixel).

Note: In the *Slider* dialog window you can enter a degree symbol ° or *pi (π)* for the interval and increment by using the following keyboard shortcuts:

- *Alt-O* (Mac OS: *Ctrl-O*) for the degree symbol °
- *Alt-P* (Mac OS: *Ctrl*-P) for the pi symbol *π*

The position of a slider may be absolute in the *Graphics View* (this means that the slider is not affected by zooming, but always remains in the visible part of the *Graphics View*) or relative to the coordinate system (see *Properties Dialog* of the corresponding number or angle).

Note: In GeoGebra, a slider is the graphical representation of a *free number* or *free angle*. You can easily create a slider for any existing *free number or angle* by showing this object in the *Graphics View* (see *Context Menu*; see tool *Show/Hide Object*).

Slope

This tool gives you the slope of a line and shows a slope triangle in the *Graphics View* (also see command *Slope*).

2.2.11. Boolean Variable Tool

Check Box to Show/Hide Objects

Clicking in the *Graphics View* creates a check box (see section Boolean Variables and Operations) that allows you to show and hide one or more objects. In the appearing dialog window you can specify which objects should be affected by the check box.
Note: You may select these objects from the list provided in the dialog window or select them with the mouse in any view.

2.2.12. Locus Tool

Locus

Select a point *B* that depends on another point *A* and whose locus should be drawn. Then, click on point *A* to create the locus of point *B* (also see command *Locus*).
Note: Point *A* has to be a point on an object (e. g. line, segment, circle).

Example:

- Type `f(x) = x^2 - 2 x - 1` into the *Input Bar* and press the *Enter*-key.
- Place a new point A on the x-axis (see tool *New Point*; see command *Point*).
- Create point `B = (x(A), f'(x(A)))` that depends on point *A*.
- Select tool *Locus* and successively click on point *B* and point *A*.
- Drag point *A* along the *x*-axis to see point *B* moving along its locus line.

2.2.13. Geometric Transformation Tools

The following geometric transformations work for points, lines, conic sections, polygons, and images.

Dilate Object from Point by Factor

UK English: *Enlarge Object from Point by Factor*

Select the object to be dilated. Then, click on a point to specify the dilation center and enter the dilation factor into the text field of the appearing dialog window (also see commands *Dilate* (US) and *Enlarge* (UK)).

Reflect Object about Line

UK English: *Reflect Object in Line*

Select the object you want to reflect. Then, click on a line to specify the mirror/line of reflection (also see command *Reflect*).

Reflect Object about Point

UK English: *Reflect Object in Point*

Select the object you want to reflect. Then, click on a point to specify the mirror/point of reflection (also see command *Reflect*).

Reflect Point about Circle

UK English: *Reflect Point in Circle*

This tool allows you to invert a point in a circle. Select the point you want to invert. Then, click on a circle to specify the mirror/circle of inversion (also see command *Reflect*).

Rotate Object around Point by Angle

Select the object you want to rotate. Then, click on a point to specify the center of rotation and enter the rotation angle into the text field of the appearing dialog window (also see command *Rotate*).

Translate Object by Vector

Select the object you want to translate. Then, click on the translation vector (also see command *Translate*).

2.2.14. Text Tool

Insert Text

With this tool you can create static and dynamic text or LaTeX formulas in the *Graphics View* (also see section Text Commands).

At first, you need to specify the location of the text in one of the following ways:

- Click in the *Graphics View* to create a new text at this location.
- Click on a point to create a new text that is attached to this point.

Then, a dialog appears where you may enter your text.

Note: You may specify the position of a text as absolute on screen or relative to the coordinate system on tab *Basic* of the *Properties Dialog*.

Static text does not depend on any mathematical objects and is usually not affected by changes of the construction.

Dynamic text contains values of objects that automatically adapt to changes made to these objects.

Mixed text is a combination of static and dynamic text. In order to create a mixed text you may enter the static part of the text using the keyboard (e. g., `Point A =`). Then, click on the object whose value you want to display in the dynamic part of the text.
Note: GeoGebra automatically adds the syntax (`"Point A = " + A`) necessary to create your mixed text: quotation marks around the static part of the text and a plus (+) symbol to connect the different parts of the text.

Input	Description
`This is static text`	Static text
`A`	Dynamic text (if point *A* exists)
`"Point A = " + A`	Two-part mixed text using the value of point *A*
`"a = " + a + "cm"`	Three-part mixed text using the value of number *a*

Note: If an object with the name *xx* already exists and you want to create a static text using the object's name, you need to enter it with quotation marks (`"xx"`). Otherwise, GeoGebra will automatically create a dynamic text that gives you the value of object *xx* instead of its name. However, you can type any text that doesn't match any existing object's name without the quotation marks.

Note: Within a mixed text, the static part needs to be in between a pair of quotation marks. Different parts of a text (e. g., static and dynamic parts) need to be connected using plus (+) symbols.

LaTeX Formulas

In GeoGebra you can write formulas as well. To do so, check the box *LaTeX formula* in the dialog window of the *Insert Text* tool and enter your formula in LaTeX syntax.
Note: In order to create text that contains a LaTeX formula as well as static text you may enter the static part of the text and then add the LaTeX formula in between a set of dollar symbols ($).
Example: `The length of the diagonal is $\sqrt{ 2 }$.`

You can select the syntax for common formula symbols from the drop-down menu next to the LaTeX checkbox. This inserts the corresponding LaTeX code into the text field and places the cursor in between a set of curly brackets. If you would like to create dynamic text within the formula, you need to click on an object causing GeoGebra to insert its name as well as the syntax for mixed text.

Some important LaTeX commands are explained in following table. Please have a look at any LaTeX documentation for further information.

LaTeX input	Result
`a \cdot b`	$a \cdot b$
`\frac{a}{b}`	$\frac{a}{b}$
`\sqrt{x}`	$\sqrt{x}$
`\sqrt[n]{x}`	$\sqrt[n]{x}$
`\vec{v}`	$\vec{v}$
`\overline{AB}`	$\overline{AB}$
`x^{2}`	x^2
`a_{1}`	a_1
`\sin\alpha + \cos\beta`	$\sin\alpha + \cos\beta$
`\int_{a}^{b} x dx`	$\int_a^b x dx$
`\sum_{i=1}^{n} i^2`	$\sum_{i=1}^{n} i^2$

2.2.15. Image Tool

Insert Image

This tool allows you to insert an image into the *Graphics View*.
First, specify the location of the image in one of the following two ways:

- Click in the *Graphics View* to specify the position of the image's lower left corner.
- Click on a point to specify this point as the lower left corner of the image.

Then, a file-open dialog appears that allows you to select the image file from the files saved on your computer.

Note: After selecting the tool *Insert Image*, you can use the keyboard shortcut *Alt*-click in order to paste an image directly from your computer's clipboard into the *Graphics View*.

Properties of Images

The **position** of an image may be absolute on screen or relative to the coordinate system. You can specify this on tab *Basic* of the *Properties Dialog* of the image.

You may specify up to three **corner points** of the image on tab *Position* of the *Properties Dialog*. This gives you the flexibility to scale, rotate, and even distort images (also see command *Corner*).

- *Corner 1*: position of the lower left corner of the image

- *Corner 2*: position of the lower right corner of the image
  Note: This corner may only be set if *Corner 1* was set before. It controls the width of the image.
- *Corner 4*: position of the upper left corner of the image
  Note: This corner may only be set if *Corner 1* was set before. It controls the height of the image.

Example: Create three points *A*, *B*, and *C* to explore the effects of the corner points.

- Set point *A* as the first and point *B* as the second corner of your image. By dragging points *A* and *B* in *Move* mode you can explore their influence.
- Now, remove point B as the second corner of the image. Set point *A* as the first and point *C* as the fourth corner and explore how dragging the points now influences the image.
- Finally, you may set all three corner points and see how dragging the points distorts your image.

Example: You already saw how to influence the position and size of your image. If you want to attach your image to a point *A* and set its width to 3 and its height to 4 units, you could do the following:

- Set *Corner 1* to `A`
- Set *Corner 2* to `A + (3, 0)`
- Set *Corner 4* to `A + (0, 4)`

Note: If you now drag point *A* in *Move* mode, the size of your image does not change.

You may specify an image as a ***Background Image*** on tab *Basic* of the *Properties Dialog*. A background image lies behind the coordinate axes and cannot be selected with the mouse any more.
Note: In order to change the background setting of an image, you may open the *Properties Dialog* by selecting *Properties…* from the *Edit* menu.

The ***Transparency*** of an image can be changed in order to see objects or axes that lie behind the image. You can set the transparency of an image by specifying a *Filling* value between 0 % and 100 % on tab *Style* of the *Properties Dialog*.

3. Algebraic Input

3.1. General Notes

The algebraic representations of mathematical objects (e. g., values, coordinates, equations) are shown in the *Algebra View*. You can create and modify objects by using the *Input Bar* at the bottom of the GeoGebra window (see sections Direct Inputand and Commands).
Note: Always press the *Enter*-key after typing algebraic input into the *Input Bar*.

Note: Pressing the *Enter*-key at any time toggles the focus between the *Input Bar* and the *Graphics View*. This allows you to enter expressions and commands into the *Input Bar* without having to click on it with the mouse first.

Naming Objects

You can assign a certain name to an object when you create it using the *Input Bar*:

- *Points*: In GeoGebra, points are always named using upper case letters. Just type in the name (e. g., *A*, *P*) and an equal sign in front of the coordinates or commands.
  Examples: `C = (2, 4)`, `P = (1; 180°)`, `Complex = 2 + i`
- *Vectors*: In order to distinguish between points and vectors, vectors need to have a lower case name in GeoGebra. Again, type in the name (e. g., *v*, *u*) and an equal sign in front of the coordinates or commands.
  Examples: `v = (1, 3)`, `u = (3; 90°)`, `complex = 1 - 2i`
- *Lines, circles, and conic sections*: These objects can be named by typing in the name and a colon in front of their equations or commands.
  Examples: `g: y = x + 3`, `c: (x-1)^2 + (y - 2)^2 = 4`,
  `hyp: x^2 - y^2 = 2`
- *Functions*: You can name functions by typing, for example, `f(x) =` or `g(x)=` in front of the function's equation or commands.
  Examples: `h(x) = 2 x + 4`, `q(x) = x^2`, `trig(x) = sin(x)`

Note:

- If you don't manually assign a name to an object, GeoGebra assigns the names of new objects in alphabetical order.
- You can create indices within the names of objects by using an underscore. For example A_1 is entered as `A_1` and s_{AB} is entered as `s_{AB}`.

Change Values

There are two ways of manipulating a *free object's* value:

- *Change* the value of the object by entering its name and the new value in the *Input Bar* (see section Direct Input).

Example: If you want to change the value of an existing number *a = 3*, type `a = 5` into the *Input Bar* and press the *Enter*-key.

- *Edit* the algebraic representation: Activate tool *Move* and double click on the object in the *Algebra View*. This opens a text box where you can edit the object's value. Press the *Enter*-key to apply your changes.

Note: While free objects' values can be changed directly, the values of dependent objects can only be influenced by changing their *parent* objects or by *redefining* the dependent object.

Display *Input Bar* History

After placing the cursor in the *Input Bar* you can use the ↑ *up* and ↓ *down* arrow keys of your keyboard in order to navigate through prior input step by step.

Note: Click on the little question mark to the left of the *Input Bar* in order to display the *help feature for the Input Bar*.

Insert Name, Value, or Definition of an Object into the *Input Bar*

Insert the name of an object: Activate tool *Move* and select the object whose name you want to insert into the *Input Bar*. Then, press the *F5* key on your keyboard.
Note: The name of the object is appended to any expression you typed into the *Input Bar* before pressing the *F5* key.

Insert the value of an object: There are two ways of inserting an object's value (e. g., *(1, 3)*, *3x – 5y = 12*) into the *Input Bar*.

- Right click (Mac OS: *Ctrl*-click) on the object and select item *Copy to Input Bar* from the appearing *Context Menu*.
- Activate tool *Move* and select the object whose value you want to insert into the *Input Bar*. Then, press the *F4* key on your keyboard.
  Note: The value of the object is appended to any expression you typed into the *Input Bar* before pressing the *F4* key.

Insert the definition of an object: There are two ways of inserting an object's definition (e. g., A = (4, 2), c = Circle[A, B]) into the *Input Bar*.

- *Alt* click on the object to insert the object's definition and delete whatever input might have been in the *Input Bar* before.
- Activate tool *Move* and select the object whose definition you want to insert into the *Input Bar*. Then, press the *F3* key on your keyboard.
  Note: The definition of the object replaces any expression you typed into the *Input Bar* before pressing the *F3* key.

3.2. Direct Input

GeoGebra can work with numbers, angles, points, vectors, segments, lines, conic sections, functions, and parametric curves. You can enter these objects into the *Input Bar* by using their coordinates or equations and pressing the *Enter*-key.

3.2.1. Numbers and Angles

Numbers

You can create numbers by using the *Input Bar*. If you only type in a number (e. g., `3`), GeoGebra assigns a lower case letter as the name of the number. If you want to give your number a specific name, you can type in the name followed by an equal sign and the number (e. g., create a decimal *r* by typing in `r = 5.32`).

Note: In GeoGebra, numbers and angles use a period (.) as a decimal point.

You can also use the constant π and the Euler constant e for expressions and calculations by selecting them from the drop down menu next to the *Input Bar* or by using keyboard shortcuts.
Note: If the variable *e* is not used as a name of an existing object yet, GeoGebra will recognize it as the Euler constant if you use it in new expressions.

Angles

Angles are entered in *degree* (°) or *radians* (rad). The constant π is useful for radian values and can also be entered as `pi`.
Note: You can enter a degree symbol (°) or the pi symbol (π) by using the following keyboard shortcuts:

- *Alt-O* (Mac OS: *Ctrl-O*) for the degree symbol °
- *Alt-P* (Mac OS: *Ctrl-P*) for the pi symbol π

Example: You can enter an angle α in degree (e. g., `α = 60°`) or in radians (e. g., `α = pi/3`).

Note: GeoGebra does all internal calculations in radians. The degree symbol (°) is nothing but the constant $\pi/180$ used to convert degree into radians.

Examples:

- If $a = 30$ is a number, then `α = a°` converts number *a* to an angle $\alpha = 30°$, without changing its value.
- If you type in `b = α / °`, the angle α is converted back to the number $b = 30$, without changing its value.

Sliders and Arrow Keys

Free numbers and angles can be displayed as sliders in the *Graphics View* (see tool *Slider*). Using the arrow keys, you may change the value of numbers and angles in the *Algebra View* too (see section Manual Animation).

Limit Value to Interval

Free numbers and angles may be limited to an interval [min, max] by using tab Slider of the *Properties Dialog* (see also tool *Slider*).

Note: For dependent angles you can specify whether they may become reflex or not on tab *Basic* of the *Properties Dialog*.

3.2.2. Points and Vectors

Points and vectors may be entered in *Cartesian* or *polar coordinate*s (see section Numbers and Angles).
Note: Upper case labels denote points whereas lower case labels refer to vectors.

Examples:

- To enter a point *P* or a vector *v* in Cartesian coordinates you may use `P = (1, 0)` or `v = (0, 5)`.
- In order to use polar coordinates type in `P = (1; 0°)` or `v = (5; 90°)`.
  Note: You need to use a semicolon to separate the two coordinates. If you don't type in the degree symbol, GeoGebra will treat the angle as if entered in radians.

In GeoGebra, you can also do calculations with points and vectors.
Examples:

- You can create the midpoint *M* of two points *A* and *B* by entering `M = (A + B) / 2` into the *Input Bar*.
- You may calculate the length of a vector *v* using `length = sqrt(v * v)`

3.2.3. Lines and Axes

Lines

You can enter a line as a linear equation in *x* and *y* or in parametric form into the *Input Bar*. In both cases previously defined variables (e. g. numbers, points, vectors) can be used within the equation.
Note: You can enter a line's name at the beginning of the input followed by a colon.

Examples:

- Type in `g: 3x + 4y = 2` to enter line *g* as a linear equation.
- Define a parameter *t* (e. g., `t = 3`) before entering line *g* in parametric form using `g: X = (-5, 5) + t (4, -3)`.

- Define the parameters `m = 2` and `b = -1`. Then, you can enter the equation `h: y = m*x + b` to get a line *h* in *y*-intercept-form.

Axes

The two coordinate axes are available in commands using the names *xAxis* and *yAxis*.
Example: The command `Perpendicular[A, xAxis]` constructs the perpendicular line to the *x*-axis through a given point *A*.

3.2.4. Conic Sections

You may enter a conic section as a quadratic equation in *x* and *y*. Prior defined variables (e. g., numbers, points, vectors) can be used within the conic's equation.
Note: The conic section's name can be entered at the beginning of the input followed by a colon.

Examples:

- Ellipse *ell*: `ell: 9 x^2 + 16 y^2 = 144`
- Hyperbola *hyp*: `hyp: 9 x^2 - 16 y^2 = 144`
- Parabola *par*: `par: y^2 = 4 x`
- Circle *c1*: `c1: x^2 + y^2 = 25`
- Circle *c2*: `c2: (x - 5)^2 + (y + 2)^2 = 25`

Note: If you define two parameters `a = 4` and `b = 3` in advance, you may enter for example an ellipse as `ell: b^2 x^2 + a^2 y^2 = a^2 b^2`.

3.2.5. Functions of *x*

To enter a function you can use previously defined variables (e. g. numbers, points, vectors) as well as other functions.

Examples:

- Function *f*: `f(x) = 3 x^3 - x^2`
- Function *g*: `g(x) = tan(f(x))`
- Nameless function: `sin(3 x) + tan(x)`

Note: All available pre-defined functions (e. g. *sin*, *cos*, *tan*) are described in section Pre-defined Functions and Operations.

In GeoGebra you can also use *commands* to get for example, the *Integral* and *Derivative* of a function.
Note: You can also use the commands `f'(x)` or `f''(x)`,… in order to get the derivatives of a previously defined function *f(x)*.
Example: Define function *f* as `f(x) = 3 x^3 - x^2`. Then, you can type in `g(x) = cos(f' (x + 2))` in order to get function *g*.

Furthermore, functions can be translated by a vector (see command *Translate*) and a *free function* can be moved with the mouse by using tool *Move*.

Limit Function to Interval

In order to limit a function to an interval [*a, b*], you can use the command *Function*.

3.2.6. Pre-defined Functions and Operations

To create numbers, coordinates, or equations (see section Direct Input) you may also use the following pre-defined functions and operations.
Note: The pre-defined functions need to be entered using parentheses. You must not put a space between the function name and the parentheses.

Operation / Function	Input
Addition	`+`
Subtraction	`-`
Multiplication	`*` or *Space* key
Scalar product	`*` or *Space* key
Division	`/`
Exponentiation	`^` or `2`
Factorial	`!`
Gamma function	`gamma( )`
Parentheses	`( )`
x-coordinate	`x( )`
y-coordinate	`y( )`
Absolute value	`abs( )`
Sign	`sgn( )`
Square root	`sqrt( )`
Cubic root	`cbrt( )`
Random number between 0 and 1	`random( )`
Exponential function	`exp( )` or e^x
Logarithm (natural, to base *e*)	`ln( )` or `log( )`
Logarithm to base 2	`ld( )`
Logarithm to base 10	`lg( )`
Cosine	`cos( )`
Sine	`sin( )`
Tangent	`tan( )`
Arc cosine	`acos( )`
Arc sine	`asin( )`
Arc tangent	`atan( )`
Hyperbolic cosine	`cosh( )`
Hyperbolic sine	`sinh( )`
Hyperbolic tangent	`tanh( )`
Antihyperbolic cosine	`acosh( )`
Antihyperbolic sine	`asinh( )`

Operation / Function	Input
Antihyperbolic tangent	`atanh( )`
Greatest integer less than or equal	`floor( )`
Least integer greater than or equal	`ceil( )`
Round	`round( )`

3.2.7. Boolean Variables and Operations

You can use the Boolean variables *true* and *false* in GeoGebra. Just type, for example, `a = true` or `b = false` into the *Input Bar* and press the *Enter*-key.

Check Box and Arrow Keys

Free Boolean variables can be displayed as check boxes in the *Graphics View* (see tool *Check Box to Show/Hide objects*). By using the arrow keys of your keyboard you may also change Boolean variables in the *Algebra View* (see section Manual Animation).

Note: You may also use Boolean variables like numbers (value 0 or 1). This allows you to use a checkbox as the dynamic speed of an *animated slider* allowing you to start and stop the animation. In this case, the *animation button* is only shown in the *Graphics View* if there is also an animated slider with static (i. e. non-dynamic) speed.

Operations

You can use the following operations for Boolean variables and conditions in GeoGebra by either selecting them from the list next to the *Input Bar* or by entering them using the keyboard:

	List	Keyboard	Example	Object types
Equal	$\stackrel{?}{=}$	==	`a` $\stackrel{?}{=}$ `b` or `a == b`	numbers, points, lines, conics *a*, *b*
Unequal	≠	!=	`a` ≠ `b` or `a != b`	numbers, points, lines, conics *a*, *b*
Less than	<	<	`a < b`	numbers *a*, *b*
Greater than	>	>	`a > b`	numbers *a*, *b*
Less or equal than	≤	<=	`a` ≤ `b` or `a <= b`	numbers *a*, *b*
Greater or equal than	≥	>=	`a` ≥ `b` or `a >= b`	numbers *a*, *b*
And	∧	&&	`a` ∧ `b` or `a && b`	Booleans *a*, *b*
Or	∨	||	`a` ∨ `b` or `a || b`	Booleans *a*, *b*
Not	¬	!	¬`a` or `!a`	Boolean *a*
Parallel	∥		`a` ∥ `b`	lines *a*, *b*
Perpendicular	⊥		`a` ⊥ `b`	lines *a*, *b*

3.2.8. List Objects and Operations

Using curly braces you can create a list of several objects (e. g. points, segments, circles).
Examples:

- `L = {A, B, C}` gives you a list consisting of three prior defined points *A*, *B*, and *C*.
- `L = {(0, 0), (1, 1), (2, 2)}` produces a list that consists of the entered points, as well as these nameless points.
  Note: By default, the elements of this list are not shown in the *Graphics View*.

Compare Lists of Objects

You can compare two lists of objects by using the following syntax:

- `List1 == List2`: Checks if the two lists are equal and gives you *true* or *false* as a result.
- `List1 != List2`: Checks if the two lists are not equal and gives you *true* or *false* as a result.

Apply Pre-defined Operations and Functions to Lists

Note: If you apply operations and pre-defined functions to lists, you will always get a new list as a result.

Addition and Subtraction examples:

- `List1 + List2`: Adds corresponding elements of two lists.
  Note: The two lists need to be of the same length.
- `List + Number`: Adds the number to every element of the list.
- `List1 - List2`: Subtracts the elements of the second list from corresponding elements of the first list.
  Note: The lists need to be of the same length.
- `List - Number`: Subtracts the number from every element of the list.

Multiplication and Division examples:

- `List1 * List2`: Multiplies corresponding elements of two lists.
  Note: The lists need to be of the same length. If the two lists are compatible matrices, matrix multiplication is used.
- `List * Number`: Multiplies every list element with the number.
- `List1 / List2`: Divides elements of the first list by corresponding elements of the second list.
  Note: The two lists need to be of the same length.
- `List / Number`: Divides every list element by the number.
- `Number / List`: Divides the number by every element of the list.

Examples using functions:

- `List^2`: Squares every element of the list.
- `sin(List)`: Applies the sine function to every element of the list.

3.2.9. Matrix Objects and Operations

GeoGebra also supports matrices, which are represented as a list of lists that contain the rows of the matrix.

Example: In GeoGebra, *{{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}* represents the matrix $\begin{Bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{Bmatrix}$.

Matrix Operations

Addition and subtraction examples:

- `Matrix1 + Matrix2`: Adds the corresponding elements of two compatible matrices.
- `Matrix1 – Matrix2`: Subtracts the corresponding elements of two compatible matrices.

Multiplication examples:

- `Matrix * Number`: Multiplies every element of the matrix by the given number.
- `Matrix1 * Matrix2`: Uses matrix multiplication to calculate the resulting matrix.
  Note: The rows of the first and columns of the second matrix need to have the same number of elements.
  Example: `{{1, 2}, {3, 4}, {5, 6}} * {{1, 2, 3}, {4, 5, 6}}` gives you the matrix *{{9, 12, 15}, {19, 26, 33}, {29, 40, 51}}*.
- `2x2 Matrix * Point` (or `Vector`): Multiplies the matrix with the given point/vector and gives you a point as a result.
  Example: `{{1, 2}, {3, 4}} * (3, 4)` gives you the point *A = (11, 25)*.
- `3x3 Matrix * Point` (or `Vector`): Multiplies the matrix with the given point/vector and gives you a point as a result.
  Example: `{{1, 2, 3}, {4, 5, 6}, {0, 0, 1}} * (1, 2)` gives you the point *A = (8, 20)*.
  Note: This is a special case for affine transformations where homogenous coordinates are used: (x, y, 1) for a point and (x, y, 0) for a vector. This example is therefore equivalent to:
  `{{1, 2, 3}, {4, 5, 6}, {0, 0, 1}} * {1, 2, 1}`.

Other examples (see also section Matrix Commands):

- `Determinant[Matrix]`: Calculates the determinant for the given matrix.
- `Invert[Matrix]`: Inverts the given matrix
- `Transpose[Matrix]`: Transposes the given matrix

3.2.10. Complex Numbers and Operations

GeoGebra does not support complex numbers directly, but you may use points to simulate operations with complex numbers.

Example: If you enter the complex number `3 + 4i` into the *Input Bar*, you get the point *(3, 4)* in the *Graphics View*. This point's coordinates are shown as *3 + 4i* in the *Algebra View*.

Note: You can display any point as a complex number in the *Algebra View*. Open the *Properties Dialog* for the point and select *Complex Number* from the list of *Coordinates* formats on tab *Algebra*.

If the variable *i* has not already been defined, it is recognized as the ordered pair *i = (0, 1)* or the complex number *0 + 1i*. This also means, that you can use this variable *i* in order to type complex numbers into the *Input Bar* (e. g., `q = 3 + 4i`).

Addition and subtraction examples:

- `(2 + 1i) + (1 – 2i)` gives you the complex number *3 – 1i*.
- `(2 + 1i) - (1 – 2i)` gives you the complex number *1 + 3i*.

Multiplication and division examples:

- `(2 + 1i) * (1 – 2i)` gives you the complex number *4 – 3i*.
- `(2 + 1i) / (1 – 2i)` gives you the complex number *0 + 1i*.

Note: The usual multiplication `(2, 1)*(1, -2)` gives you the scalar product of the two vectors.

Other examples:
GeoGebra also recognizes expressions involving real and complex numbers.

- `3 + (4 + 5i)` gives you the complex number *7 + 5i*.
- `3 – (4 + 5i)` gives you the complex number *-1 - 5i*.
- `3 / (0 + 1i)` gives you the complex number *0 - 3i*.
- `3 * (1 + 2i)` gives you the complex number *3 + 6i*.

3.3. Commands

Using *commands* you can produce new and modify existing objects.

Note: A command's result may be named by entering a label followed by an equal sign (=). In the example below, the new point is named *S*.
Example: To get the intersection point of two lines *g* and *h* you can enter `S = Intersect[g, h]` (see command *Intersect*).

Note: You can also use indices within the names of objects: A_1 is entered as `A_1` while s_{AB} is created using `s_{AB}`.

Automatic Completion of Commands

When you type a command into GeoGebra's *Input Bar*, the software tries to automatically complete the command for you. This means that after you typed in the first two letters of

the command into the *Input Bar*, GeoGebra displays the first command of an alphabetically sorted list that starts with these letters.

- In order to *accept this suggestion* and place the cursor in between the brackets, hit the *Enter*-key.
- If the suggested command is not the one you wanted to type in, just *keep typing*. GeoGebra will adapt its suggestions to the letters you enter.

3.3.1. General Commands

ConstructionStep

`ConstructionStep[]`: Returns the current *Construction Protocol* step as a number.
`ConstructionStep[Object]`: Returns the *Construction Protocol* step for the given object as a number.

Delete

`Delete[Object]`: Deletes the object and all its dependents objects.

Note: Also see tool *Delete Object*

Relation

`Relation[Object a, Object b]`: Shows a message box that gives you information about the relation of object *a* and object *b*.
Note: This command allows you to find out whether two objects are equal, if a point lies on a line or conic, or if a line is tangent or a passing line to a conic.

Note: Also see tool *Relation*

3.3.2. Boolean Commands

If

`If[Condition, Object]`: Yields a copy of the object if the condition evaluates to *true*, and an undefined object if it evaluates to *false*.
`If[Condition, Object a, Object b]`: Yields a copy of object *a* if the condition evaluates to *true*, and a copy of object *b* if it evaluates to *false*.

IsDefined

`IsDefined[Object]`: Returns *true* or *false* depending on whether the object is defined or not.

IsInteger

`IsInteger[Number]`: Returns *true* or *false* depending whether the number is an integer or not.

3.3.3. Number Commands

AffineRatio

`AffineRatio[Point A, Point B, Point C]`: Returns the affine ratio λ of three collinear points *A*, *B*, and *C*, where *C = A + λ * AB*.

Area

`Area[Point A, Point B, Point C, ...]`: Calculates the area of the polygon defined by the given points *A*, *B*, *C,…*
`Area[Conic c]`: Calculates the area of a conic section *c* (circle or ellipse).

Note:

- In order to calculate the area between two function graphs, you need to use the command *Integral*.
- Also see tool *Area*

AxisStep

`AxisStepX[]`: Returns the current step width for the *x*-axis.
`AxisStepY[]`: Returns the current step width for the *y*-axis.

Note: Together with the *Corner* and *Sequence* commands, the *AxisStep* commands allow you to create custom axes (also see section Customizing Coordinate Axes and Grid).

BinomialCoefficient

`BinomialCoefficient[Number n, Number r]`: Calculates the binomial coefficient *n choose r*.

Circumference

`Circumference[Conic]`: Returns the circumference of a circle or ellipse.

CrossRatio

`CrossRatio[Point A, Point B, Point C, Point D]`: Calculates the cross ratio *λ* of four collinear points *A*, *B*, *C*, and *D*, where
λ = AffineRatio[B, C, D] / AffineRatio[A, C, D].

Curvature

`Curvature[Point, Function]`: Calculates the curvature of the function in the given point.
`Curvature[Point, Curve]`: Calculates the curvature of the curve in the given point.

Distance

`Distance[Point A, Point B]`: Yields the distance of two points *A* and *B.*
`Distance[Point, Line]`: Yields the distance of the point and the line.
`Distance[Line g, Line h]`: Yields the distance of the parallel lines *g* and *h*.
Note: The distance of intersecting lines is 0. Thus, this command is only interesting for parallel lines.

Note: Also see tool *Distance or Length*

GCD

UK English: HCF

`GCD[Number a, Number b]`: Calculates the greatest common divisor of numbers *a* and *b* (UK-English: HCF = highest common factor).
`GCD[List of Numbers]`: Calculates the greatest common divisor of the list of numbers (UK-English: HCF = highest common factor).

Integer Division

`Div[Number a, Number b]`: Calculates the integer quotient for division of number *a* by number *b*.

Integral

`Integral[Function, Number a, Number b]`: Returns the definite integral of the function in the interval *[a , b]*.
Note: This command also draws the area between the function graph of *f* and the *x*-axis.
`Integral[Function f, Function g, Number a, Number b]`: Yields the definite integral of the difference *f(x) - g(x)* in the interval [*a, b*].
Note: This command also draws the area between the function graphs of *f* and *g*.

Note: Also see command for *Indefinite Integral*

Iteration

`Iteration[Function, Number x0, Number n]`: Iterates the function *n* times using the given start value *x0.*
Example: After defining `f(x) = x^2` the command `Iteration[f, 3, 2]` gives you the result $(3^2)^2 = 81$.

LCM

`LCM[Number a, Number b]`: Calculates the least common multiple of two numbers *a* and *b* (UK English: LCM = lowest common multiple).

`LCM[List of numbers]`: Calculates the least common multiple of the elements of the list (UK English: LCM = lowest common multiple).

Length

`Length[Vector]`: Yields the length of the vector.

`Length[Point A]`: Yields the length of the position vector of the given point .

`Length[Function, Number x1, Number x2]`: Yields the length of the function graph in the interval [*x1, x2*].

`Length[Function, Point A, Point B]`: Yields the length of the function graph between the two points *A* and *B*.

Note: If the given points do not lie on the function graph, their *x*-coordinates are used to determine the interval.

`Length[Curve, Number t1, Number t2]`: Yields the length of the curve between the parameter values *t1* and *t2*.

`Length[Curve c, Point A, Point B]`: Yields the length of curve *c* between two points *A* and *B* that lie on the curve.

`Length[List]`: Yields the length of the list which is the number of elements in the list.

Note: Also see tool *Distance or Length*

LinearEccentricity

`LinearEccentricity[Conic]`: Calculates the linear eccentricity of the conic section.

Note: The linear eccentricity is the distance between a conic's center and its focus, or one of its two foci.

LowerSum

`LowerSum[Function, Number a, Number b, Number n]`: Yields the lower sum of the given function on the interval [*a, b*] with *n* rectangles.

Note: This command draws the rectangles for the lower sum as well.

Minimum and Maximum

`Min[Number a, Number b]`: Yields the minimum of the given numbers *a* and *b*.

`Max[Number a, Number b]`: Yields the maximum of the given numbers *a* and *b*.

Modulo Function

`Mod[Integer a, Integer b]`: Yields the remainder when integer *a* is divided by integer *b*.

Parameter

`Parameter[Parabola]`: Returns the parameter of the parabola, which is the distance of directrix and focus.

Perimeter

`Perimeter[Polygon]`: Returns the perimeter of the polygon.

Radius

`Radius[Circle]`: Returns the radius of the circle.

Random commands

`RandomBetween[Min Integer, Max Integer]`: Generates a random integer between *min* and *max* (inclusive).
`RandomBinomial[Number n of Trials, Probability p]`: Generates a random number from a binomial distribution with *n* trials and probability *p*.
`RandomNormal[Mean, Standard Deviation]`: Generates a random number from a normal distribution with given mean and standard deviation.
`RandomPoisson[Mean]`: Generates a random number from a Poisson distribution with given mean.

SemiMajorAxisLength

`SemiMajorAxisLength[Conic]`: Returns the length of the semimajor axis (half of the major axis) of the conic section.

SemiMinorAxisLength

`SemiMinorAxisLength[Conic]`: Returns the length of the semiminor axis (half of the minor axis) of the conic section.

Slope

`Slope[Line]`: Returns the slope of the given line.
Note: This command also draws the slope triangle whose size may be changed on tab *Style* of the *Properties Dialog*.

Note: Also see tool *Slope*

TrapezoidalSum

UK English: TrapeziumSum

`TrapezoidalSum[Function, Number a, Number b, Number n]`: Calculates the trapezoidal sum of the function in the interval [*a, b*] using *n* trapezoids.
Note: This command draws the trapezoids of the trapezoidal sum as well.

UpperSum

`UpperSum[Function, Number a, Number b, Number n]`: Calculates the upper sum of the function on the interval [*a*, *b*] using *n* rectangles.
Note: This command draws the rectangles of the upper sum as well.

3.3.4. Angle Command

Angle

`Angle[Vector v1, Vector v2]`: Returns the angle between two vectors *v1* and *v2* (between 0 and 360°) .
`Angle[Line g, Line h]`: Returns the angle between the direction vectors of two lines *g* and *h* (between 0 and 360°) .
`Angle[Point A, Point B, Point C]`: Returns the angle enclosed by *BA* and *BC* (between 0 and 360°), where point *B* is the apex.
`Angle[Point A, Point B, Angle α]`: Returns the angle of size *α* drawn from point *A* with apex *B*.
Note: The point *Rotate[A, α, B]* is created as well.
`Angle[Conic]`: Returns the angle of twist of a conic section's major axis (see command *Axes*) .
`Angle[Vector]`: Returns the angle between the *x*-axis and given vector.
`Angle[Point]`: Returns the angle between the *x*-axis and the position vector of the given point.
`Angle[Number]`: Converts the number into an angle (result between 0 and 2pi).
`Angle[Polygon]`: Creates all angles of a polygon in mathematically positive orientation (i. e., counter clockwise).
Note: If the polygon was created in counter clockwise orientation, you get the interior angles. If the polygon was created in clockwise orientation, you get the exterior angles.

Note: Also see tools *Angle* and *Angle with Given Size*

3.3.5. Point Commands

Center

UK English: Centre

`Center[Conic]`: Returns the center of a circle, ellipse, or hyperbola.

Note: Also see tool *Midpoint or Center*

Centroid

`Centroid[Polygon]`: Returns the centroid of the polygon.

Corner

`Corner[Number n of Corner]`: Creates a point at the corner of the *Graphics View* (*n = 1, 2, 3, 4*) which is never visible on screen.
`Corner[Image, Number n of Corner]`: Creates a point at the corner of the image (*n = 1, 2, 3, 4*).
`Corner[Text, Number n of Corner]`: Creates a point at the corner of the text (*n = 1, 2, 3, 4*).
Note: The numbering of the corners is counter-clockwise and starts at the lower left corner.

Extremum

UK English: TurningPoint

`Extremum[Polynomial]`: Yields all local extrema of the polynomial function as points on the function graph.

Focus

`Focus[Conic]`: Yields (all) foci of the conic section.

InflectionPoint

`InflectionPoint[Polynomial]`: Yields all inflection points of the polynomial as points on the function graph.

Intersect

`Intersect[Line g, Line h]`: Yields the intersection point of lines *g* and *h*.
`Intersect[Line, Conic]`: Yields all intersection points of the line and conic section (max. 2).
`Intersect[Line, Conic, Number n]`: Yields the n^{th} intersection point of the line and the conic section.
`Intersect[Conic c1, Conic c2]`: Yields all intersection points of conic sections *c1* and *c2* (max. 4).
`Intersect[Conic c1, Conic c2, Number n]`: Yields the n^{th} intersection point of conic sections *c1* and *c2*.
`Intersect[Polynomial f1, Polynomial f2]`: Yields all intersection points of polynomials *f1* and *f2*.
`Intersect[Polynomial f1, Polynomial f2, Number n]`: Yields the n^{th} intersection point of polynomials *f1* and *f2*.
`Intersect[Polynomial, Line]`: Yields all intersection points of the polynomial and the line.
`Intersect[Polynomial, Line, Number n]`: Yields the n^{th} intersection point of the polynomial and the line.
`Intersect[Function f, Function g, Point A]`: Calculates the intersection point of functions *f* and *g* by using Newton's method with initial point *A*.
`Intersect[Function, Line, Point A]`: Calculates the intersection point of the function and the line by using Newton's method with initial point *A*.

Note: Also see tool *Intersect two Objects*

Midpoint

`Midpoint[Point A, Point B]`: Returns the midpoint of points *A* and *B*.
`Midpoint[Segment]`: Returns the midpoint of the segment.

Note: Also see tool *Midpoint or Center*

Point

`Point[Line]`: Returns a point on the line.
`Point[Conic]`: Returns a point on the conic section.
`Point[Function]`: Returns a point on the function.
`Point[Polygon]`: Returns a point on the polygon.
`Point[Vector ]`: Returns a point on the vector.
`Point[Point, Vector]`: Creates a new point by adding the vector to the given point.

Note: Also see tool *New Point*

Root

`Root[Polynomial]`: Yields all roots of the polynomial as intersection points of the function graph and the *x*-axis.
`Root[Function, Number a]`: Yields one root of the function using the initial value *a* for Newton's method.
`Root[Function, Number a, Number b]`: Yields one root of the function in the interval [*a*, *b*] (regula falsi).

Vertex

`Vertex[Conic]`: Returns (all) vertices of the conic section.

3.3.6. Vector Commands

CurvatureVector

`CurvatureVector[Point, Function]`: Yields the curvature vector of the function in the given point.
`CurvatureVector[Point, Curve]`: Yields the curvature vector of the curve in the given point.

Direction

`Direction[Line]`: Yields the direction vector of the line.
Note: A line with equation $ax + by = c$ has the direction vector $(b, -a)$.

PerpendicularVector

`PerpendicularVector[Line]`: Returns the perpendicular vector of the line.
Note: A line with equation *ax + by = c* has the perpendicular vector *(a, b)*.
`PerpendicularVector[Vector v]`: Returns the perpendicular vector of the given vector.
Note: A vector with coordinates *(a, b)* has the perpendicular vector *(-b, a)*.

UnitPerpendicularVector

`UnitPerpendicularVector[Line]`: Returns the perpendicular vector with length 1 of the given line.
`UnitPerpendicularVector[Vector]`: Returns the perpendicular vector with length 1 of the given vector.

UnitVector

`UnitVector[Line]`: Yields the direction vector with length 1 of the given line.
`UnitVector[Vector]`: Yields a vector with length 1, which has the same direction and orientation as the given vector.

Vector

`Vector[Point A, Point B]`: Creates a vector from point *A* to point *B*.
`Vector[Point]`: Returns the position vector of the given point.

Note: Also see tool *Vector between Two Points*

3.3.7. Segment Command

Segment

`Segment[Point A, Point B]`: Creates a segment between two points *A* and *B*.
`Segment[Point A, Number a]`: Creates a segment with length *a* and starting point *A*.
Note: The endpoint of the segment is created as well.

Note: Also see tools *Segment between Two Points* and *Segment with Given Length from Point*

3.3.8. Ray Command

Ray

`Ray[Point A, Point B]`: Creates a ray starting at point *A* through point *B*.
`Ray[Point, Vector v]`: Creates a ray starting at the given point which has the direction vector *v*.

Note: Also see tool *Ray through Two Points*

3.3.9. Polygon Command

Polygon

`Polygon[Point A, Point B, Point C,...]`: Returns a polygon defined by the given points *A*, *B*, *C*,…

`Polygon[Point A, Point B, Number n]`: Creates a regular polygon with *n* vertices (including points *A* and *B*).

Note: Also see tools *Polygon* and *Regular Polygon*

3.3.10. Line Commands

AngleBisector

`AngleBisector[Point A, Point B, Point C]`: Returns the angle bisector of the angle defined by points *A*, *B*, and *C*.
Note: Point *B* is apex of this angle.

`AngleBisector[Line g, Line h]`: Returns both angle bisectors of the lines.

Note: Also see tool *Angle Bisector*

Asymptote

`Asymptote[Hyperbola]`: Yields both asymptotes of the hyperbola.

Axes

`Axes[Conic]`: Returns the major and minor axis of a conic section.

ConjugateDiameter

`ConjugateDiameter[Line, Conic]`: Returns the conjugate diameter of the diameter that is parallel to the line (relative to the conic section).

`ConjugateDiameter[Vector, Conic]`: Returns the conjugate diameter of the diameter that is parallel to the vector (relative to the conic section).

Directrix

`Directrix[Parabola]`: Yields the directrix of the parabola.

Line

`Line[Point A, Point B]`: Creates a line through two points *A* and *B*.

`Line[Point, Parallel Line]`: Creates a line through the given point parallel to the given line.

`Line[Point, Direction Vector v]`: Creates a line through the given point with direction vector *v*.

Note: Also see tool *Line through Two Points*

MajorAxis

`MajorAxis[Conic]`: Returns the major axis of the conic section.

MinorAxis

`MinorAxis[Conic]`: Returns the minor axis of the conic section.

PerpendicularLine

`PerpendicularLine[Point, Line]`: Creates a line through the point perpendicular to the given line.
`PerpendicularLine[Point, Vector]`: Creates a line through the point perpendicular to the given vector.

Note: Also see tool *Perpendicular Line*

PerpendicularBisector

`PerpendicularBisector[Point A, Point B]`: Yields the perpendicular bisector of the line segment *AB*.
`PerpendicularBisector[Segment]`: Yields the perpendicular bisector of the segment.

Note: Also see tool *Perpendicular Bisector*

Polar

`Polar[Point, Conic]`: Creates the polar line of the given point relative to the conic section.

Note: Also see tool *Polar or Diameter Line*

Tangent

`Tangent[Point, Conic]`: Creates (all) tangents through the point to the conic section.
`Tangent[Line, Conic]`: Creates (all) tangents to the conic section that are parallel to the given line.
`Tangent[Number a, Function]`: Creates the tangent to the function at $x = a$.
`Tangent[Point A, Function]`: Creates the tangent to the function at $x = x(A)$.
Note: *x(A)* is the x-coordinate of point *A*.
`Tangent[Point, Curve]`: Creates the tangent to the curve in the given point.

Note: Also see tool *Tangents*

3.3.11. Conic Section Commands

Circle

`Circle[Point M, Number r]`: Yields a circle with midpoint *M* and radius *r*.
`Circle[Point M, Segment]`: Yields a circle with midpoint *M* whose radius is equal to the length of the given segment.
`Circle[Point M, Point A]`: Yields a circle with midpoint *M* through point *A*.
`Circle[Point A, Point B, Point C]`: Yields a circle through the given points *A*, *B* and *C*.

Note: Also see tools *Compass*, *Circle with Center through Point*, *Circle with Center and Radius*, and *Circle through Three Points*

Conic

`Conic[Point A, Point B, Point C, Point D, Point E]`: Returns a conic section through the five given points *A*, *B*, *C*, *D*, and *E*.
Note: If four of the points lie on one line the conic section is not defined.

Note: Also see tool *Conic through Five Points*

Ellipse

`Ellipse[Point F, Point G, Number a]`: Creates an ellipse with focal points *F* and *G* and semimajor axis length *a*.
Note: Condition: $2a > Distance[F, G]$
`Ellipse[Point F, Point G, Segment]`: Creates an ellipse with focal points *F* and *G* where the length of the semimajor axis equals the length of the given segment.
`Ellipse[Point F, Point G, Point A]`: Creates an ellipse with foci *F* and *G* passing through point *A*.

Note: Also see tool *Ellipse*

Hyperbola

`Hyperbola[Point F, Point G, Number a]`: Creates a hyperbola with focal points *F* and *G* and semimajor axis length *a*.
Note: Condition: $0 < 2a < Distance[F, G]$
`Hyperbola[Point F, Point G, Segment]`: Creates a hyperbola with focal points *F*. and *G* where the length of the semimajor axis equals the length of segment *s*.
`Hyperbola[Point F, Point G, Point A]`: Creates a hyperbola with foci *F* and *G* passing through point *A*.

Note: Also see tool *Hyperbola*

OsculatingCircle

`OsculatingCircle[Point, Function]`: Yields the osculating circle of the function in the given point.
`OsculatingCircle[Point, Curve]`: Yields the osculating circle of the curve in the given point.

Parabola

`Parabola[Point F, Line g]`: Returns a parabola with focal point *F* and directrix *g*.

Note: Also see tool *Parabola*

3.3.12. Function Commands

Conditional Functions

You can use the Boolean command *If* in order to create a conditional function.
Note: You can use derivatives and integrals of such functions and intersect conditional functions like "normal" functions.

Examples:

- `f(x) = If[x < 3, sin(x), x^2]` gives you a function that equals *sin(x)* for x < 3 and x^2 for $x \geq 3$.
- `a ≟ 3 ∧ b ≥ 0` tests whether "*a* equals 3 *and b* is greater than or equal to 0".

Note: Symbols for conditional statements (e. g., ≟, ∧, ≥) can be found in the list next to the right of the *Input Bar*.

Derivative

`Derivative[Function]`: Returns the derivative of the function.
`Derivative[Function, Number n]`: Returns the n^{th} derivative of the function.

Note: You can use `f'(x)` instead of `Derivative[f]` as well as `f''(x)` instead of `Derivative[f, 2]` and so on.

Expand

`Expand[Function]`: Multiplies out the brackets of the expression.
Example: `Expand[(x + 3)(x − 4)]` gives you $f(x) = x^2 - x - 12$

Factor

UK English: Factorise

`Factor[Polynomial]`: Factors the polynomial.
Example: `Factor[x^2 + x − 6]` gives you *f(x) = (x-2)(x+3)*

Function

`Function[Function, Number a, Number b]`: Yields a function graph, that is equal to *f* on the interval [*a, b*] and not defined outside of [*a, b*].
Note: This command should be used only in order to display functions in a certain interval.

Example: `f(x) = Function[x^2, -1, 1]` gives you the graph of function x^2 in the interval [*-1, 1*]. If you then type in `g(x) = 2 f(x)` you will get the function $g(x) = 2x^2$, but this function is not restricted to the interval [*-1, 1*].

Integral

`Integral[Function]`: Yields the indefinite integral for the given function.

Note: Also see command for Definite integral

Polynomial

`Polynomial[Function]`: Yields the expanded polynomial function.
Example: `Polynomial[(x - 3)^2]` yields $x^2 - 6x + 9$.
`Polynomial[List of n points]`: Creates the interpolation polynomial of degree *n-1* through the given *n* points.

Simplify

`Simplify[Function]`: Simplifies the terms of the given function if possible.
Examples:

- `Simplify[x + x + x]` gives you a function *f(x) = 3x*.
- `Simplify[sin(x) / cos(x)]` gives you a fun*ction f(x) = tan(x)*.
- `Simplify[-2 sin(x) cos(x)]` gives you *a function f(x) = sin(-2 x)*.

TaylorPolynomial

`TaylorPolynomial[Function, Number a, Number n]`: Creates the power series expansion for the given function about the point *x = a* to order *n*.

3.3.13. Parametric Curve Command

Curve

`Curve[Expression e1, Expression e2, Parameter t, Number a, Number b]`: Yields the Cartesian parametric curve for the given *x*-expression *e1* and *y*-expression *e2* (using parameter *t*) within the given interval [*a, b*].
Example: Input of `c = Curve[2 cos(t), 2 sin(t), t, 0, 2 pi]` creates a circle with radius 2 around the origin of the coordinate system.

Note: Parametric curves can be used with pre-defined functions and arithmetic operations.
Example: Input `c(3)` returns the point at parameter position 3 on curve *c*.

Note: Using the mouse you can also place a point on a curve using tool *New Point* or command *Point*. Since the parameters *a* and *b* are dynamic you could use slider variables as well (see tool *Slider*).

Commands for Parametric Curves

`Curvature[Point, Curve]`: Calculates the curvature of the curve in the given point.
`CurvatureVector[Point, Curve]`: Yields the curvature vector of the curve in the given point.
`Derivative[Curve]`: Returns the derivative of the parametric curve.
`Derivative[Curve, Number n]`: Returns the n^{th} derivative of the parametric curve.
`Length[Curve, Number t1, Number t2]`: Yields the length of the curve between the parameter values *t1* and *t2*.
`Length[Curve c, Point A, Point B]`: Yields the length of curve *c* between two points *A* and *B* that lie on the curve.
`OsculatingCircle[Point, Curve]`: Yields the osculating circle of the curve in the given point.
`Tangent[Point, Curve]`: Creates the tangent to the curve in the given point.

3.3.14. Arc and Sector Commands

Note: The algebraic value of an arc is its length and the value of a sector is its area.

Arc

`Arc[Conic, Point A, Point B]`: Returns a conic section arc between two points *A* and *B* on the conic section *c*.
Note: This only works for a circle or ellipse.
`Arc[Conic, Number t1, Number t2]`: Returns a conic section arc between two parameter values *t1* and *t2* on the conic section.
Note: Internally the following parameter forms are used:

- *Circle*: *(r cos(t), r sin(t))* where *r* is the circle's radius.
- *Ellipse*: *(a cos(t), b sin(t))* where *a* and *b* are the lengths of the semimajor and semiminor axis.

CircularArc

`CircularArc[Point M, Point A, Point B]`: Creates a circular arc with midpoint *M* between points *A* and *B*.
Note: Point *B* does not have to lie on the arc.

Note: Also see tool *Circular Arc with Center between Two Points*

CircularSector

`CircularSector[Point M, Point A, Point B]`: Creates a circular sector with midpoint *M* between two points *A* and *B*.
Note: Point *B* does not have to lie on the arc of the sector.

Note: Also see tool *Circular Sector with Center between Two Points*

CircumcircularArc

`CircumcircularArc[Point A, Point B, Point C]`: Creates a circular arc through three points *A*, *B*, and *C*, where *A* is the starting point and *C* is the endpoint of the circumcircular arc.

Note: Also see tool *Circumcircular Arc through Three Points*

CircumcircularSector

`CircumcircularSector[Point A, Point B, Point C]`: Creates a circular sector whose arc runs through the three points *A*, *B*, and *C*. Point *A* is the starting point and point *C* is the endpoint of the arc.

Note: Also see tool *Circumcircular Sector through Three Points*

Sector

`Sector[Conic, Point A, Point B]`: Yields a conic section sector between two points *A* and *B* on the conic section.
Note: This works only for a circle or ellipse.

`Sector[Conic, Number t1, Number t2]`: Yields a conic section sector between two parameter values *t1* and *t2* on the conic section.
Note: Internally the following parameter forms are used:

- *Circle*: *(r cos(t), r sin(t))* where *r* is the circle's radius.
- *Ellipse*: *(a cos(t), b sin(t))* where *a* and *b* are the lengths of the semimajor and semiminor axis.

Semicircle

`Semicircle[Point A, Point B]`: Creates a semicircle above the segment *AB*.

Note: Also see tool *Semicircle*

3.3.15. Text Commands

FormulaText

`FormulaText[Object]`: Returns the formula for the object as a LaTeX text.
Example: If $a = 2$ and $f(x) = a\,x^2$, then `FormulaText[f]` returns $2\,x^2$ (as a LaTeX text).

`FormulaText[Object, Boolean]`: Returns the formula for the object as LaTeX text. The Boolean variable determines if values are substituted for variables (*true*) or if variable names are shown in the text (*false*).
Examples: If $a = 2$ and $f(x) = a\,x^2$, then

`FormulaText[f, true]` returns $2\,x^2$ (as a LaTeX text).
`FormulaText[f, false]` returns $a\,x^2$ (as a LaTeX text).

FractionText

`FractionText[Number]`: Converts the number to a fraction, which is displayed as a (LaTeX) text object in the *Graphics View*.
Example: If *a: y = 1.5 x + 2* is a line, then `FractionText[Slope[a]]` gives you the fraction *3/2* as a text.

LetterToUnicode

`LetterToUnicode["Letter"]`: Turns a single letter into its Unicode number
Note: The letter needs to be in between a set of quotation marks.
Example: `LetterToUnicode["a"]` returns the number 97

Name

`Name[Object]`: Returns the name of an object as a text in the *Graphics View*.
Note: Use this command in dynamic text for objects that might be renamed. The *Name* command is the opposite of the *Object* command.

Object

`Object[Name of Object as Text]`: Returns the object for a given name. The result is always a dependent object.
Note: The *Object* command is the opposite of the *Name* command.
Example: If points *A1*, *A2*, ... , *A20* exist and slider *n = 2*, then `Object["A" + n]` gives you a copy of point *A2*.

TableText

`TableText[List 1, List 2, List 3,...]`: Creates a text that contains a table of the list objects.
Note: By default, every list is displayed in a new column of the table.
Examples:

- `TableText[{x^2, 4}, {x^3, 8}, {x^4, 16}]`
  creates a table as a text object with three rows and two columns. All items of the table are left aligned.
- `TableText[Sequence[i^2, i, 1, 10]]`
  creates a table as a text object with one row. All items of the table are left aligned .

`TableText[List 1, List 2, List 3,..., "Alignment of text"]`: Creates a text that contains a table of the list objects. The optional text “Alignment of text” controls the orientation and alignment of the table text.
Note: Possible values are "vl", "vc", "vr", "v", "h", "hl", "hc", "hr". Default is "hl".

- "v" = vertical, i. e. lists are columns
- "h" = horizontal, i. e. lists are rows
- "l" = left aligned

- "r" = right aligned
- "c" = centered

Examples:

- `TableText[{1,2,3,4},{1,4,9,16},"v"]`
  creates a text with two columns and four rows whose elements are left aligned.
- `TableText[{1,2,3,4},{1,4,9,16},"h"]`
  creates a text with two rows and four columns whose elements are left aligned.
- `TableText[{11.2,123.1,32423.9,"234.0"},"vr"]`
  creates a text with one column whose elements are right aligned.

Text

`Text[Object]`: Returns the formula for the object as a text object.

Note: By default, values are substituted for variables.

Example: If *a = 2* and $c = a^2$, then `Text[c]` returns the text "4".

`Text[Object, Boolean]`: Returns the formula for the object as a text object. The Boolean variable determines if values are substituted for variables (*true*) or if variable names are shown in the text (*false*).

Example: If *a = 2* and $c = a^2$, then

`Text[c, true]` returns the text *"4"*.

`Text[c, false]` returns the text *"a^2"*.

`Text[Object, Point]`: Returns the formula for the object as a text object at the position of the given point.

Example: `Text["hello", (2, 3)]` draws the text at the position *(2, 3)*.

`Text[Object, Point, Boolean]`: Returns the formula for the object as a text object at the position of the given point. The Boolean variable determines if values are substituted for variables (*true*) or if variable names are shown in the text (*false*).

TextToUnicode

`TextToUnicode["Text"]`: Turns the text into a list of Unicode numbers, one for each character.

Examples:

- `TextToUnicode["Some text"]` gives you the list of Unicode numbers *{83, 111, 109, 101, 32, 116, 101, 120, 116}*.
- If *text1* is *"hello"*, then `TextToUnicode[text1]` gives you the list of Unicode numbers *{104, 101, 108, 108, 111}*.

UnicodeToLetter

`UnicodeToLetter[Integer]`: Converts the integer Unicode number back into a letter which is displayed as a text object in the *Graphics View*.

Example: `UnicodeToLetter[97]` gives you the text *"a"*.

UnicodeToText

`UnicodeToText[List of Integers]`: Converts the integer Unicode numbers back into text.
Example: `UnicodeToText[{104, 101, 108, 108, 111}]` gives you the text *"hello"*.

3.3.16. Locus Command

Locus

`Locus[Point Q, Point P]`: Returns the locus line of point *Q* which depends on point *P*.
Note: Point *P* has to be a point on an object (e. g. line, segment, circle).

Note: Also see tool *Locus*

3.3.17. List and Sequence Commands

Append

`Append[List, Object]`: Appends the object to the list.
Example: `Append[{1, 2, 3}, 4]` gives you *{1, 2, 3, 4}*.
`Append[Object, List]`: Appends the list to the object.
Example: `Append[4, {1, 2, 3}]` gives you *{4, 1, 2, 3}*.

CountIf

`CountIf[Condition, List]`: Counts the number of elements in the list satisfying the condition.
Examples:

- `CountIf[x < 3, {1, 2, 3, 4, 5}]` gives you the number 2.
- `CountIf[x<3, A1:A10]` where *A1:A10* is a range of cells in the spreadsheet, counts all cells whose values are less than 3.

Element

`Element[List, Number n]`: Yields the n^{th} element of the list.
Note: The list can contain only elements of one object type (e. g., only numbers or only points).

First

`First[List]`: Returns the first element of the list.
`First[List, Number n of elements]`: Returns a new list that contains just the first *n* elements of the list.

Insert

`Insert[Object, List, Position]`: Inserts the object in the list at the given position.

Example: `Insert[x^2, {1, 2, 3, 4, 5}, 3]` places x^2 at the third position and gives you the list *{1, 2, x^2, 3, 4, 5}*.

Note: If the position is a negative number, then the position is counted from the right.

Example: `Insert[x^2, {1, 2, 3, 4, 5}, -1]` places x^2 at the end of the list and gives you the list *{1, 2, 3, 4, 5, x^2}*.

`Insert[List 1, List 2, Position]`: Inserts all elements of *list1* in *list2* at the given position.

Example: `Insert[{11, 12}, {1, 2, 3, 4, 5}, 3]` places the elements of *list1* at the third (and following) position(s) of *list2* and gives you the list *{1, 2, 11, 12, 3, 4, 5}*.

Note: If the position is a negative number, then the position is counted from the right.

Example: `Insert[{11, 12}, {1, 2, 3, 4, 5}, -2]` places the elements of *list1* at the end of *list2* before its last element and gives you *{1, 2, 3, 4, 11, 12, 5}*.

Intersection

`Intersection[List 1, List 2]`: Gives you a new list containing all elements that are part of both lists.

IterationList

`IterationList[Function, Number x0, Number n]`:

Gives you a list of length *n*+1 whose elements are iterations of the function starting with the value *x0.*

Example: After defining function `f(x) = x^2` the command `L = IterationList[f, 3, 2]` gives you the list *L = {3, 9, 81}*.

Join

`Join[List 1, List 2, ...]`: Joins the two (or more) lists.

Note: The new list contains all elements of the initial lists even if they are the same. The elements of the new list are not re-ordered.

Example: `Join[{5, 4, 3}, {1, 2, 3}]` creates the list {*5, 4, 3, 1, 2, 3}*.

`Join[List of lists]`: Joins the sub-lists into one longer list.

Note: The new list contains all elements of the initial lists even if they are the same. The elements of the new list are not re-ordered.

Examples:

- `Join[{{1, 2}}]` creates the list *{1, 2}*.
- `Join[{{1, 2, 3}, {3, 4}, {8, 7}}]` creates the list *{1, 2, 3, 3, 4, 8, 7}*.

KeepIf

`KeepIf[Condition, List]`: Creates a new list that only contains those elements of the initial list that fulfill the condition.
Example: `KeepIf[x<3, {1, 2, 3, 4, 1, 5, 6}]` returns the new list *{1, 2, 1}*.

Last

`Last[List]`: Returns the last element of the list.
`Last[List, Number n of Elements]`: Returns a list containing just the last *n* elements of the list.

Length

`Length[List]`: Yields the length of the list, which is the number of list elements.

Min

`Min[List]`: Returns the minimal element of the list.

Max

`Max[List]`: Returns the maximal element of the list.

Product

`Product[List of Numbers]`: Calculates the product of all numbers in the list.

RemoveUndefined

`RemoveUndefined[List]`: Removes undefined objects from a list.
Example: `RemoveUndefined[Sequence[(-1)^i, i, -3, -1, 0.5]]` removes the second and fourth element of the sequence which have a non-integer exponent and therefore, are undefined.

Reverse

`Reverse[List]`: Reverses the order of a list.

Sequence

`Sequence[Expression, Variable i, Number a, Number b]`: Yields a list of objects created using the given expression and the index *i* that ranges from number *a* to number *b*.
Example: `L = Sequence[(2, i), i, 1, 5]` creates a list of points whose *y*-coordinates range from 1 to 5: *L = {(2, 1), (2, 2), (2, 3), (2, 4), (2, 5)}*.
`Sequence[Expression, Variable i, Number a, Number b, Increment]`: Yields a list of objects created using the given expression and the index *i* that ranges from number *a* to number *b* with given increment.
Example: `L = Sequence[(2, i), i, 1, 3, 0.5]` creates a list of points whose

y-coordinates range from 1 to 3 with an increment of 0.5:
L = {(2, 1), (2, 1.5), (2, 2), (2, 2.5), (2, 3)}.

Note: Since the parameters *a* and *b* are dynamic you could use *slider* variables as well.

Sort

`Sort[List]`: Sorts a list of numbers, text objects, or points.
Note: Lists of points are sorted by *x*-coordinates.
Examples:

- `Sort[{3, 2, 1}]` gives you the list *{1, 2, 3}*.
- `Sort[{"pears", "apples", "figs"}]` gives you the list elements in alphabetical order.
- `Sort[{(3, 2), (2, 5), (4, 1)}]` gives you *{(2, 5), (3, 2), (4, 1)}*.

Sum

`Sum[List]`: Calculates the sum of all list elements.
Note: This command works for numbers, points, vectors, text, and functions.
Examples:

- `Sum[{1, 2, 3}]` gives you a number *a = 6*.
- `Sum[{x^2, x^3}]` gives you $f(x) = x^2 + x^3$.
- `Sum[Sequence[i,i,1,100]]` gives you a number *a = 5050*.
- `Sum[{(1, 2), (2, 3)}]` gives you a point *A = (3, 5)*.
- `Sum[{(1, 2), 3}]` gives you point *B = (4, 2)*.
- `Sum[{"a","b","c"}]` gives you the text *"abc"*.

`Sum[List, Number n of Elements]`: Calculates the sum of the first *n* list elements.
Note: This command works for numbers, points, vectors, text, and functions.
Example: `Sum[{1, 2, 3, 4, 5, 6}, 4]` gives you the number *a = 10*.

Take

`Take[List, Start Position m, End Position n]`: Returns a list containing the elements from positions *m* to *n* of the initial list.

Union

`Union[List 1, List 2]`: Joins the two lists and removes elements that appear multiple times.

3.3.18. Geometric Transformation Commands

Dilate

UK English: Enlarge

`Dilate[Point A, Number, Point S]`: Dilates point *A* from point *S* using the given factor.

`Dilate[Line, Number, Point S]`: Dilates the line from point *S* using the given factor.
`Dilate[Conic, Number, Point S]`: Dilates the conic section from point *S* using the given factor.
`Dilate[Polygon, Number, Point S]`: Dilates the polygon from point *S* using the given factor.
Note: New vertices and segments are created too.
`Dilate[Image, Number, Point S]`: Dilates the image from point *S* using the given factor.

Note: Also see tool *Dilate Object from Point*

Reflect

`Reflect[Point A, Point B]`: Reflects point *A* about point *B*.
`Reflect[Line, Point]`: Reflects the line about the given point.
`Reflect[Conic, Point]`: Reflects the conic section about the given point.
`Reflect[Polygon, Point]`: Reflects the polygon about the given point.
Note: New vertices and segments are created as well.
`Reflect[Image, Point]`: Reflects the image about the given point.
`Reflect[Point, Line]`: Reflects the point about the given line.
`Reflect[Line g, Line h]`: Reflects line *g* about line *h*.
`Reflect[Conic, Line]`: Reflects the conic section about the line.
`Reflect[Polygon, Line]`: Reflects the polygon about the line.
Note: New vertices and segments are created as well.
`Reflect[Image, Line]`: Reflects the image about the line.
`Reflect[Point, Circle]`: Inverts the point in the circle.

Note: Also see tools *Reflect Object about Point*; *Reflect Object about Line*; *Reflect Point about Circle*

Rotate

`Rotate[Point, Angle]`: Rotates the point by the angle around the axis origin.
`Rotate[Vector, Angle]`: Rotates the vector by the angle around the starting point of the vector.
`Rotate[Line, Angle]`: Rotates the line by the angle around the axis origin.
`Rotate[Conic, Angle]`: Rotates the conic section by the angle around the axis origin.
`Rotate[Polygon, Angle]`: Rotates the polygon by the angle around the axis origin.
Note: New vertices and segments are created as well.
`Rotate[Image, Angle]`: Rotates the image by the angle around the axis origin.
`Rotate[Point A, Angle, Point B]`: Rotates point *A* by the angle around point *B*.
`Rotate[Line, Angle, Point]`: Rotates the line by the angle around the point.
`Rotate[Vector, Angle, Point]`: Rotates the vector by the angle around the point.
`Rotate[Conic, Angle, Point]`: Rotates the conic section by the angle around the point.
`Rotate[Polygon, Angle, Point]`: Rotates the polygon by the angle around point *B*.
Note: New vertices and segments are created as well.
`Rotate[Image, Angle, Point]`: Rotates the image by the angle around the point.

Note: Also see tool *Rotate Object around Point by Angle*

Translate

`Translate[Point, Vector ]`: Translates the point by the vector.
`Translate[Line, Vector]`: Translates the line by the vector.
`Translate[Conic, Vector]`: Translates the conic by the vector.
`Translate[Function, Vector]`: Translates the function by the vector.
`Translate[Polygon, Vector]`: Translates the polygon by the vector.
Note: New vertices and segments are created as well.
`Translate[Image, Vector]`: Translates the image by the vector.
`Translate[Vector, Point]`: Translates the vector *v* to point.

Note: Also see tool *Translate Object by Vector*

3.3.19. Statistics Commands

BarChart

`BarChart[Start Value, End Value, List of Heights]`: Creates a bar chart over the given interval where the number of bars is determined by the length of the list whose elements are the heights of the bars.
Example: `BarChart[10, 20, {1,2,3,4,5} ]` gives you a bar chart with five bars of specified height in the interval [*10, 20*].

`BarChart[Start Value a, End Value b, Expression, Variable k, From Number c, To Number d]`: Creates a bar chart over the given interval [*a, b*], that calculates the bars' heights using the expression whose variable *k* runs from number *c* to number *d*.
Example: If *p = 0.1*, *q = 0.9*, and *n = 10* are numbers, then
`BarChart[ -0.5, n + 0.5, BinomialCoefficient[n,k]*p^k*q^(n-k), k, 0, n ]` gives you a bar chart in the interval [*-0.5, n+0.5*]. The heights of the bars depend on the probabilities calculated using the given expression.

`BarChart[Start Value a, End Value b, Expression, Variable k, From Number c, To Number d, Step Width s]`: Creates a bar chart over the given interval [*a, b*], that calculates the bars' heights using the expression whose variable *k* runs from number *c* to number *d* using step width *s*.

`BarChart[List of Raw Data, Width of Bars]`: Creates a bar chart using the given raw data whose bars have the given width.
Example: `BarChart[ {1,1,1,2,2,2,2,2,3,3,3,5,5,5,5}, 1]`

`BarChart[List of Data, List of Frequencies]`: Creates a bar chart using the list of data with corresponding frequencies.
Note: The List of data must be a list where the numbers go up by a constant amount.
Examples:

- `BarChart[{10,11,12,13,14}, {5,8,12,0,1}]`
- `BarChart[{5, 6, 7, 8, 9}, {1, 0, 12, 43, 3}]`

- `BarChart[{0.3, 0.4, 0.5, 0.6}, {12, 33, 13, 4}]`

`BarChart[List of Data , List of Frequencies, Width of Bars w]`: Creates a bar chart using the list of data and corresponding frequencies whose bars are of width *w*.
Note: The List of data must be a list where the numbers go up by a constant amount
Examples:

- `BarChart[{10,11,12,13,14}, {5,8,12,0,1}, 0.5]` leaves gaps between bars.
- `BarChart[{10,11,12,13,14}, {5,8,12,0,1}, 0]` produces a line graph.

BoxPlot

`BoxPlot[yOffset, yScale, List of Raw Data]`: Creates a box plot using the given raw data and whose vertical position in the coordinate system is controlled by variable *yOffset* and whose height is influenced by factor *yScale*.
Example: `BoxPlot[0, 1, {2,2,3,4,5,5,6,7,7,8,8,8,9}]`

`BoxPlot[yOffset, yScale, Start Value a, Q1, Median, Q3, End Value b]`: Creates a box plot for the given statistical data in interval [a, b].

CorrelationCoefficient

`CorrelationCoefficient[List of x-Coordinates, List of y-Coordinates]`: Calculates the product moment correlation coefficient using the given *x*- and *y*-coordinates.

`CorrelationCoefficient[List of Points]`: Calculates the product moment correlation coefficient using the coordinates of the given points.

Covariance

`Covariance[List 1 of Numbers, List 2 of Numbers]`: Calculates the covariance using the elements of both lists.

`Covariance[List of Points]`: Calculates the covariance using the x- and y-coordinates of the points.

FitLine

`FitLine[List of Points]`: Calculates the *y* on *x* regression line of the points.
`FitLineX[List of Points]`: Calculates the *x* on *y* regression line of the points.

Note: Also see tool *Best Fit Line*

Other Fit Commands

`FitExp[List of Points]`: Calculates the exponential regression curve.
`FitLog[List of Points]`: Calculates the logarithmic regression curve.
`FitLogistic[List of Points]`: Calculates the regression curve in the form *a/(1+b x^(-kx))*.

Note: The first and last data point should be fairly close to the curve. The list should have at least 3 points, preferably more.

`FitPoly[List of Points, Degree n of Polynomial]`: Calculates the regression polynomial of degree *n*.

`FitPow[List of Points]`: Calculates the regression curve in the form $a\,x^b$.

Note: All points used need to be in the first quadrant of the coordinate system.

`FitSin[List of Points]`: Calculates the regression curve in the form *a + b sin(cx + d)*.

Note: The list should have at least 4 points, preferably more. The list should cover at least two extremal points. The first two local extremal points should not be too different from the absolute extremal points of the curve.

Histogram

`Histogram[List of Class Boundaries, List of Heights]`: Creates a histogram with bars of the given heights. The class boundaries determine the width and position of each bar of the histogram.

Example: `Histogram[{0, 1, 2, 3, 4, 5}, {2, 6, 8, 3, 1}]` creates a histogram with 5 bars of the given heights. The first bar is positioned at the interval [*0, 1*], the second bar is positioned at the interval [*1, 2*], and so on.

`Histogram[List of Class Boundaries, List of Raw Data]`: Creates a histogram using the raw data. The class boundaries determine the width and position of each bar of the histogram and are used to determine how many data elements lie in each class.

Example: `Histogram[{1, 2, 3, 4},{1.0, 1.1, 1.1, 1.2, 1.7, 2.2, 2.5, 4.0}]` creates a histogram with 3 bars, with the heights 5 (first bar), 2 (second bar), and 1 (third bar).

InverseNormal

`InverseNormal[Mean μ, Standard Deviation σ, Probability P]`: Calculates the function $\Phi^{-1}(P) * \sigma + \mu$ where Φ^{-1} is the inverse of the probability density function Φ for N(0,1).

Note: Returns the *x*-coordinate with the given probability to the left under the normal distribution curve.

Mean commands

`Mean[List of Numbers]`: Calculates the mean of the list elements.

`MeanX[List of Points]`: Calculates the mean of the *x*-coordinates of the points in the list.

`MeanY[List of Points]`: Calculates the mean of the *y*-coordinates of the points in the list.

Median

`Median[List of Numbers]`: Determines the median of the list elements.

Mode

`Mode[List of Numbers]`: Determines the mode(s) of the list elements.

Examples:

- `Mode[{1,2,3,4}]` returns an empty list *{}*.
- `Mode[{1,1,1,2,3,4}]` returns the list *{1}* .
- `Mode[{1,1,2,2,3,3,4}]` returns the list *{1, 2, 3}*.

Normal

`Normal[Mean μ, Standard Deviation σ, Variable Value x]`: Calculates the function $\Phi((x - \mu) / \sigma)$ where Φ is the probability density function for N(0,1) .
Note: Returns the probability for a given *x*-coordinate value (or area under the normal distribution curve to the left of the given *x*-coordinate).

Quartile commands

`Q1[List of Numbers]`: Determines the lower quartile of the list elements.
`Q3[List of Numbers]`: Determines the upper quartile of the list elements.

SD

`SD[List of Numbers]`: Calculates the standard deviation of the numbers in the list.

Sigma commands

`SigmaXX[List of Numbers]`: Calculates the sum of squares of the given numbers.
Example: In order to work out the variance of a list you may use `SigmaXX[list]/Length[list] - Mean[list]^2`.
`SigmaXX[List of Points]`: Calculates the sum of squares of the *x*-coordinates of the given points.
`SigmaXY[List of x-Coordinates, List of y-Coordinates]`: Calculates the sum of the products of the *x*- and *y*-coordinates.
`SigmaXY[List of Points]`: Calculates the sum of the products of the x- and y-coordinates.
Example: You can work out the covariance of a list of points using `SigmaXY[list]/Length[list] - MeanX[list] * MeanY[list]`.
`SigmaYY[List of Points]`: Calculates the sum of squares of *y*-coordinates of the given points.

Commands for statistic quantities

`Sxx[List of Numbers]`: Calculates the statistic $\Sigma(x^2) - \Sigma(x) * \Sigma(x)/n$.
`Sxx[List of Points]`: Calculates the statistic $\Sigma(x^2) - \Sigma(x) * \Sigma(x)/n$ using the *x*-coordinates of the given points.
`Sxy[List of Numbers, List of Numbers]`: Calculates the statistic $\Sigma(xy) - \Sigma(x) * \Sigma(y)/n$.
`Sxy[List of Points]`: Calculates the statistic $\Sigma(xy) - \Sigma(x) * \Sigma(y)/n$.

`Syy[List of Points]`: Calculates the statistic $\Sigma(y^2) - \Sigma(y) * \Sigma(y)/n$ using the *y*-coordinates of the given points.

Note: These quantities are simply unnormalized forms of the variances and covariance of *X* and *Y* given by *Sxx = N var(X)*, *Syy = N var(Y)*, and *Sxy = N cov(X, Y)*.
Example: You can work out the correlation coefficient for a list of points using `Sxy[list] / sqrt(Sxx[list] Syy[list])`.

Variance

`Variance[List of Numbers]`: Calculates the variance of list elements.

3.3.20. Spreadsheet Commands

CellRange

`CellRange[Start Cell, End Cell]`: Creates a list containing the cell values in this cell range.
Example: Enter the following values into the corresponding spreadsheet cells: `A1 = 1`, `A2 = 4`, `A3 = 9`. The command `CellRange[A1, A3]` then gives you the list *{1, 4, 9}*.

Column

`Column[Spreadsheet Cell]`: Returns the column of the cell as a number (starting at 1).
Example: `Column[B3]` gives you number *a = 2* since column *B* is the second column of the spreadsheet.

ColumnName

`ColumnName[Spreadsheet Cell]`: Returns the column name of the cell as a text.
Example: `ColumnName[A1]` gives you a text "*A*" in the *Graphics View*.

Row

`Row[Spreadsheet Cell]`: Returns the row number of a spreadsheet cell (starting at 1).
Example: `Row[B3]` gives you number *a = 3*.

3.3.21. Matrix Commands

Determinant

`Determinant[Matrix]`: Returns the determinant of the matrix.
Example: `Determinant[{{1, 2}, {3, 4}}]` gives you the number *a = -2*.

Invert

`Invert[Matrix]`: Inverts the given matrix.

Example: `Invert[{{1, 2}, {3, 4}}]` gives you the inverse matrix *{{-2, 1}, {1.5, -0.5}}*.

Transpose

Transpose[Matrix]: Transposes the matrix.

Example: `Transpose[{{1, 2}, {3, 4}}]` gives you the matrix *{{1, 3}, {2, 4}}*.

4. Menu Items

4.1. File Menu

New Window

Keyboard shortcut: *Ctrl-N* (Mac OS: *Cmd-N*)

This menu item opens a new GeoGebra window that uses the default settings of the GeoGebra user interface.
Note: If you change and *save* some of these settings, the new GeoGebra window will open using your *customized settings*.

New

This menu item opens a new and empty user interface in the same GeoGebra window. You are asked if you would like to save the existing construction before opening the new user interface.
Note: The new user interface adopts the settings used for the prior construction. For example, if the coordinate axes were hidden before selecting the menu item *New*, the axes will be hidden in the new user interface as well.

Open…

Keyboard shortcut: *Ctrl-O* (Mac OS: *Cmd-O*)

This menu item allows you to open a GeoGebra file (file name extension GGB) that is saved on your computer.
Note: In order to open a GeoGebra file you can also drag it with the mouse to the GeoGebra window and drop it there.

Save

Keyboard shortcut: *Ctrl-S* (Mac OS: *Cmd-S*)

This menu item allows you to save your current construction as a GeoGebra file (file name extension GGB) on your computer.
Note: If the file was saved before, this menu item overwrites the old file by using the same file name.

Save As…

This menu item allows you to save your current construction as a GeoGebra file (file name extension GGB). You will be asked to enter a new name for your GeoGebra file before it is saved on your computer.

Print Preview

Keyboard shortcut: *Ctrl-P* (Mac OS: *Cmd-P*)

This menu item opens the *Print Preview* window for the *Graphics View*. You may specify Title, Author, Date and the Scale of your printout (in cm).

Note: Press the *Enter*-key after you made a change in order to update the preview of your printout.

Export - *Dynamic Worksheet* as Webpage (html)…

Keyboard shortcut: *Ctrl-Shift-W* (Mac OS: *Cmd-Shift-W*)

This menu item allows you to export your current construction as a webpage (HTML) in order to create a so called *Dynamic Worksheet*.
Please see section Creating Interactive Webpagesfor more information.

Export - *Graphics View* as Picture (png, eps)…

Keyboard shortcut: *Ctrl-Shift-P* (Mac OS: *Cmd-Shift-P*)

This menu item allows you to save GeoGebra's *Graphics View* as a picture file on your computer. In the appearing dialog window, you can select the picture file *Format*, and change the *Scale* (in cm) and *Resolution* (in dpi) of the picture.

When exporting the *Graphics View* as a picture you can choose out of the following *formats*:

- **PNG – Portable Network Graphics**: This is a pixel graphics format. The higher the resolution (dpi), the better the quality (300dpi will usually suffice). PNG graphics should not be scaled subsequently to avoid a loss of quality.
  PNG graphic files are well suited for the use on web pages (HTML) and in word processing documents.
  Note: Whenever you insert a PNG graphic file into a word processing document (menu *Insert*, *Image from file*) make sure that the size is set to 100 %. Otherwise the given scale (in cm) would be changed.
- **EPS – Encapsulated Postscript**: This is a vector graphics format. EPS pictures may be scaled without loss of quality. EPS graphic files are well suited for the use with vector graphics programs (e. g., Corel Draw) and professional text processing systems (e. g., LaTeX).
  The resolution of an EPS graphic is always 72dpi. This value is only used to calculate the true size of an image in centimeters and has no effect on the image's quality.
  Note: The transparency effect with filled polygons or conic sections is not possible with EPS.

- **PDF – Portable Document Format** (see EPS format above)
  Note: In SVG and PDF export you have the option to export text as editable text or shapes. This stores the text either as text (this lets you edit the text in e. g., InkScape) or as Bézier curves (this guarantees that the text looks the same even if the correct font is not installed).
- **SVG – Scaleable Vector Graphic** (see EPS format above)
- **EMF – Enhanced Meta Format** (see EPS format above)

Export - *Graphics View* to Clipboard

Keyboard shortcut: *Ctrl-Shift-C* (Mac OS: *Cmd-Shift-C*)

This menu item copies the *Graphics View* to your computer's clipboard. Afterwards, you can easily paste this picture into other documents (e. g., word processing document).

Export - *Graphics View* as PSTricks…

Keyboard shortcut: *Ctrl-Shift-T* (Mac OS: *Cmd-Shift-T*)

This menu item allows you to save the *Graphics View* as a PSTricks picture file, which is a LaTeX picture format.

Export - *Graphics View* as PGF/TikZ……

This menu item allows you to save the *Graphics View* as a PGF/TikZ picture file, which is a LaTeX picture format.

Close

Keyboard shortcut: *Alt-F4* (Mac OS: *Cmd-W*)

This menu item closes the GeoGebra window. If you didn't save your construction prior to selecting *Close*, you are asked if you would like to do so.

4.2. Edit Menu

Undo

Keyboard shortcut: *Ctrl-Z* (Mac OS: *Cmd-Z*)

This menu item allows you to undo your activities step by step.
Note: You can also use the *Undo* button to the right of the *Toolbar*.

Redo

Keyboard shortcut: *Ctrl-Y* (Mac OS: *Cmd-Shift-Z*)

This menu item allows you to redo your activities step by step.
Note: You can also use the *Redo* button to the right of the *Toolbar*.

Delete

Keyboard shortcut: *Delete* key

This menu items allows you to delete selected objects and their dependent objects.
Note: You need to select the objects you want to delete first (e. g., use a selection rectangle).

Select All

Keyboard shortcut: *Ctrl-A* (Mac OS: *Cmd-A*)

This menu item allows you to select all objects used in your construction.

Select Current Layer

Keyboard shortcut: *Ctrl-L* (Mac OS: *Cmd-L*)

This menu item allows you to select all objects that are on the same layer as a selected object.
Note: You need to select one object that lies on the desired layer prior to using this menu item.

Select Descendants

Keyboard shortcut: *Ctrl-Shift-Q* (Mac OS: *Cmd-Shift-Q*)

This menu item allows you to select all objects that depend on the selected object.
Note: You need to select the *parent* object prior to using this menu item.

Select Ancestors

Keyboard shortcut: *Ctrl-Q*

This menu item allows you to select all objects that are ancestors of the selected object, meaning all objects the selected one depends on.
Note: You need to select the dependent object prior to using this menu item.

Graphics View to Clipboard

Keyboard shortcut: *Ctrl-Shift-C* (Mac OS: *Cmd-Shift-C*)

This menu item copies the *Graphics View* to your computer's clipboard. Afterwards, you can easily paste this picture into other documents (e. g., word processing document).

Properties…

Keyboard shortcut: *Ctrl-E* (Mac OS: *Cmd-E*)

This menu item opens the *Properties Dialog* which allows you to modify the properties of all objects used in the GeoGebra file.

4.3. View Menu

Axes

This menu item allows you to show or hide the coordinate axes in the *Graphics View*.
Note: You can use the *Properties Dialog* for *Graphics View* in order to customize the coordinate axes.

Grid

This menu item allows you to show or hide the coordinate grid in the *Graphics View*.
Note: You can use the *Properties Dialog* for *Graphics View* in order to customize the coordinate grid.

Algebra View

Keyboard shortcut: *Ctrl-Shift-A* (Mac OS: *Cmd-Shift-A*)

This menu item allows you to show or hide the *Algebra View*.

Spreadsheet View

Keyboard shortcut: *Ctrl-Shift-S* (Mac OS: *Cmd-Shift-S*)

This menu item allows you to show or hide the *Spreadsheet View*.

Auxiliary Objects

This menu item allows you to show or hide auxiliary objects in the *Algebra View*.

Horizontal Splitting

This menu item allows you to split the GeoGebra window either vertically or horizontally into the different views.

Input Bar

This menu item allows you to show or hide the *Input Bar* at the bottom of the GeoGebra window.

Command List

This menu item allows you to show or hide the list of commands next to the *Input Bar* at the bottom of the GeoGebra window.

Construction Protocol…

This menu item allows you to open the *Construction Protocol* in a new window.

***Navigation Bar* for Construction Steps**

This menu item allows you to show or hide the *Navigation Bar* at the bottom of the *Graphics View*.

Refresh Views

Keyboard shortcut: *Ctrl-F* (Mac OS: *Cmd-F*)

This menu item allows you to repaint all views on screen.
Note: You can use this menu item to delete any traces of points or lines in the *Graphics View*.

Recompute All Objects

Keyboard shortcut: *F9*

This menu item recomputes all objects used in your GeoGebra file.
Note: You can use this menu item to create new random numbers if you used any in your GeoGebra file.

4.4. Options Menu

Global options may be changed in the menu *Options*.
Note: To change object settings, please use the *Context Menu* and *Properties Dialog*.

Point Capturing

This menu item determines if the *point capturing* is *On* or *Off* or if points are captured by the grid (item *On (Grid)*).
Note: Option *Automatic* turns the point capturing *On* when the grid or the coordinate system are shown and turns it *Off* if they are hidden.

Angle Unit

This menu item determines whether angles are displayed in *Degree* (°) or *Radians* (rad).
Note: Input is always possible in both ways (degree and radian).

Rounding

This menu item allows you to set the number of decimal places or significant figures displayed on screen.

Continuity

GeoGebra allows you to turn the continuity heuristic *On* or *Off* in the *Options* menu. The software uses a near-to-heuristic to keep moving intersection points (e. g., line-conic, conic-conic) close to their old positions and avoid jumping intersection points.

Note: By default this heuristic is turned off. For user defined tools continuity is always turned off as well.

Point Style

This menu item determines whether by default points are displayed as ● dots, ○ circles, or **x** crosses.

Checkbox Size

This menu item allows you to set the size of checkboxes to *Regular* or *Large*.
Note: If you are using GeoGebra as a presentation tool or are working with an interactive whiteboard, setting the checkbox size to *Large* might make it easier for you to use checkboxes.

Right Angle Style

Determines whether right angles are displayed as a rectangle □, a dot •, or like all other angles (*Off*).

Coordinates

This menu item determines whether coordinates of points are displayed as *A = (x, y)* or *A(x | y)*.

Labeling

You can specify whether the label of a newly created object should be shown or not. You can choose between the settings *All New Objects*, *No New Objects*, *New Points Only*, and *Automatic*.
Note: The setting *Automatic* shows the labels of newly created objects if the *Algebra View* is shown.

Font Size

This menu item determines the font size for labels and text in points (pt).
Note: If you are using GeoGebra as a presentation tool, increasing the font size makes it easier for your audience to read text, labels, and algebraic input you are using.

Language

GeoGebra is multilingual and allows you to change the current language setting. This affects all input including command names and all output.
Note: No matter which language was selected, the globe icon will lead you back to the language menu. All language names are always displayed in English.

Drawing Pad

This menu item opens a dialog where *properties of the Drawing Pad* (e. g. coordinate grid and axes, background color) can be set.
Note: You can also open this dialog window by right clicking (Mac OS: *Ctrl*-click) on the drawing pad.

Save Settings

GeoGebra remembers your favorite settings (e. g., settings in the *Options* menu, current *Toolbar* and *Graphics View* settings) if you select *Save settings* in the *Options* menu.

Restore Default Settings

You can restore the default settings of GeoGebra using this menu item.

4.5. Tools Menu

Create New Tool…

Based on an existing construction you can create your own tools in GeoGebra. After preparing the construction of your tool, choose *Create new tool* in the *Tools* menu. In the appearing dialog you can specify the output and input objects of your tool and choose names for the *Toolbar icon* and corresponding *command*.

Note: Your tool can be used both with the mouse and as a command in the *Input Bar*. All tools are automatically saved in your GGB construction file.

Manage Tools…

Using the *Manage tools* dialog you can delete a tool or modify its name and icon. You can also save selected tools to a *GeoGebra Tools File* (GGT). This file can be used later on (*File* menu*, Open*) to load the tools into another construction.
Note: Opening a GGT file doesn't change your current construction, but opening a GGB file does.

Customize *Toolbar*…

You can customize the tools in GeoGebra's *Toolbar* by selecting *Customize Toolbar* in the menu *Tools*. This is especially useful for *Dynamic Worksheets* where you want to restrict the available tools in the *Toolbar*.

Note: The current *Toolbar* setting is saved with your construction in a GGB file.

4.6. Window Menu

New Window

Keyboard shortcut: *Ctrl-N* (Mac OS: *Cmd-N*)

This menu item opens a new GeoGebra window that uses the default settings of the GeoGebra interface.
Note: If you change and save some of these settings the new GeoGebra window will open using your customized settings.

List of Open GeoGebra Windows

If you have more than one GeoGebra window open, this menu item allows you to switch between these different windows.
Note: This might be helpful when you are using GeoGebra as a presentation tool and want to have several GeoGebra files open at the same time as well as to toggle between them.

4.7. Help Menu

Help

This menu item gives you access to the HTML-version of the GeoGebra Help document. Depending on how you installed GeoGebra on your computer you may need Internet access for this feature:

- If you downloaded and installed GeoGebra using the **installer file** from the webpage, you don't need Internet access in order to use the help feature. The HTML-version of the GeoGebra Help document is locally saved on your computer when you install GeoGebra.
- If you used **GeoGebraWebstart** to install GeoGebra on your computer, you need to have Internet access in order to use this help feature. If you don't have Internet access, you will get an error message.

Note: The HTML-version of this GeoGebra Help document is available online at http://www.geogebra.org/help.

www.geogebra.org

If you have Internet access, this menu item opens the GeoGebra webpage in your default web browser (http://www.geogebra.org).

GeoGebra Forum

If you have Internet access, this menu item opens the GeoGebra online User Forum in your default web browser (http://www.geogebra.org/forum).

Note: You can post and answer GeoGebra-related questions and problems in the GeoGebra User Forum.

GeoGebraWiki

If you have Internet access, this menu item opens the GeoGebraWiki webpage in your default web browser (http://www.geogebra.org/wiki).

Note: The GeoGebra Wiki is a pool of free instructional materials created with GeoGebra by users from all around the world.

About / License

This menu item opens a dialog window that gives you information about the license of GeoGebra and gives credit to people who support the GeoGebra project by contributing in many different ways (e. g., programming, translations).

5. Special GeoGebra Features

5.1. Animation

5.1.1. Automatic Animation

GeoGebra allows you to animate one or several free numbers and/or angles at the same time if they are shown as *sliders* in the *Graphics View*.

If you want to animate a free number or angle in GeoGebra, you need to right click (Mac OS: *Ctrl*-click) on the number or angle and select *Animation On* from the appearing *Context Menu*. In order to stop the animation, you need to un-check *Animation On* in the same *Context Menu*.

Note: After animating a free number or a free angle, an animation button appears in the lower left corner of the *Graphics View*. It allows you to either pause or continue an animation.

In the *Properties Dialog* on tab *Slider* you can change the behavior of the animation:
On the one hand, you may control the *Speed* of the animation.
Note: A speed of 1 means that the animation takes about 10 seconds to run once through the interval of the slider.
On the other hand, you may change how the animation cycle is repeated:

- <=> *Oscillating*:
  The animation cycle alternates between *Decreasing* and *Increasing*.
- => *Increasing*:
  The slider value is always increasing. After reaching the maximum value of the slider, it jumps back to the minimum value and continues the animation.
- <= *Decreasing*:
  The slider value is always decreasing. After reaching the minimum value of the slider, it jumps back to the maximum value and continues the animation.

Note: While an automatic animation is activated, GeoGebra remains fully functional. This allows you to make changes to your construction while the animation is playing.

5.1.2. Manual Animation

To manually change a number or angle continuously, select the *Move* tool. Then, click on a *free number or angle* and press either the + or – key or the arrow keys on you keyboard. Keeping one of these keys pressed allows you to produce manual animations.

Example: If a point's coordinates depend on a number *k* like in *P = (2 k, k)*, the point will move along a straight line when *k* is changed continuously.

Note: You can adjust the increment of the slider on tab *Slider* of the *Properties Dialog* of this object.

Keyboard Shortcuts:

- *Shift* + *arrow* key gives you a step width of 0.1 units
- *Ctrl* + *arrow* key gives you a step width of 10 units
- *Alt* + *arrow* key gives you a step width of 100 units

Note: A point on a line can also be moved along its line using the + or – key.

5.2. Conditional Visibility

Apart from just showing or hiding certain objects you can also have their visibility status depend on a certain condition. For example, you would like an object to appear on screen if you check a *checkbox* positioned in the *Graphics View* or if a *slider* is changed to a certain value.

Conditionally Show or Hide Existing Objects

You can use the tool *Checkbox to Show / Hide Objects* in order to create a checkbox that controls the visibility of one or more existing objects on screen. Alternatively, you could also create a Boolean variable (e. g., `b = true`) using the *Input Bar* and make it visible as a checkbox in the *Graphics View* by changing its visibility status (e. g., use tool *Show / Hide Object* or use the *Context Menu*). In order to use this Boolean variable as a condition for the visibility of certain objects, you need to follow the steps described below.

Changing the Visibility of Newly Created Objects

In the *Properties Dialog*, you can enter a condition for the visibility of an object on tab *Advanced*.
Note: You can select the logic operators (e. g., ≠, ≥, ∧, ‖) from the drop down menu in order to create your conditional statements.

Examples:

- If *a* is a slider, then the conditional statement `a < 2` means that the corresponding object is only shown in the *Graphics View* if the slider's value is less than 2.
- If *b* is a Boolean variable, you can use `b` as a conditional statement. The corresponding object is shown whenever the value of *b* is *true* and is hidden when the value of *b* is *false*.
- If *g* and *h* are two lines and you would like a text to be shown whenever these lines are parallel, then you could use `g || h` as a conditional statement for the text.

5.3. User Defined Tools

GeoGebra allows you to create your own construction tools based on an existing construction. Once created, your custom tool can be used both with the mouse and as a command in the *Input Bar*. All tools are automatically saved in your GeoGebra file.

Creating a User Defined Tool

First, create the construction your tool should be able to create later on. In the *Tools* menu, click on *Create New Tool* in order to open the corresponding dialog box. Now you need to fill in the three tabs *Output Objects*, *Input Objects*, and *Name and Icon* in order to create your custom tool.

Example:
Create a Square-tool that creates a square whenever you click on two existing points or on two empty spots in the *Graphics View*.

- Construct a square starting with two points *A* and *B*. Construct the other vertices and connect them with the tool Polygon to get the square *poly1*.
- Select *Create New Tool* in the *Tools* menu.
- Specify the *Output Objects*: Click on the square or select it from the drop down menu. Also, specify the edges of the square as *Output Objects*.
- Specify the *Input Objects*: GeoGebra automatically specifies the *Input Objects* for you (here: points *A* and *B*). You can also modify the selection of input objects using the drop down menu or by clicking on them in your construction.
- Specify the *Tool Name* and *Command Name* for your new tool.
  Note: The *Tool Name* will appear in GeoGebra's *Toolbar*, while the *Command Name* can be used in GeoGebra's *Input Bar*.
- You may also enter text to be shown in the *Toolbar Help*.
- You can also choose an image from you computer for the *Toolbar* icon. GeoGebra resizes your image automatically to fit on a *Toolbar* button.

Saving a User Defined Tool

You can save your custom tools so you can reuse them in other GeoGebra constructions. In the *Tools* menu, select *Manage Tools*. Then, select the custom tool you want to save from the appearing list. Click on button *Save As…* in order to save your custom tool on your computer.
Note: User defined tools are saved as files with the file name extension *GGT* so you can distinguish custom tool files from usual GeoGebra files (*GGB*).

Accessing a User Defined Tool

If you open a new GeoGebra interface using item *New* from the *File* menu, after you created a custom tool, it will still be part of the GeoGebra *Toolbar*. However, if you open a new GeoGebra window (item *New Window* from the *File* menu), or open GeoGebra on another day, your custom tools won't be part of the *Toolbar* any more.

There are different ways of making sure that your user defined tools are displayed in the *Toolbar* of a new GeoGebra window:

- After creating a new user defined tool you can save your settings using item *Save Settings* from the *Options* menu. From now on, your customized tool will be part of the GeoGebra *Toolbar*.
  Note: You can remove the custom tool from the *Toolbar* after opening item *Customize Toolbar…* from the *Tools* menu. Then, select your custom tool from the list of tools on the left hand side of the appearing dialog window and click button *Remove >*. Don't forget to save your settings after removing the custom tool.
- After saving your custom tool on your computer (as a GGT file), you can import it into a new GeoGebra window at any time. Just select item *Open* from the *File* menu and open the file of your custom tool.
  Note: Opening a GeoGebra tool file (GGT) in GeoGebra doesn't affect your current construction. It only makes this tool part of the current GeoGebra *Toolbar*.

5.4. Dynamic Colors

In GeoGebra, you can change the color of objects using tab *Color* of the *Properties Dialog*. However, you can also have the color of an object change dynamically: Open the *Properties Dialog* for a certain object whose color you would like to change and click on tab *Advanced*. There you will find a section called *Dynamic Colors* with text boxes for the color components *Red*, *Green*, and *Blue*.
Note: In each of these text boxes, you can enter a function with range [*0, 1*].

Example:

- Create three sliders *a*, *b*, and *c* with an interval from 0 to 1.
- Create a polygon whose color should be influenced by the slider values.
- Open the *Properties Dialog* for the polygon *poly1* and enter the names of the three sliders into the text boxes for the color components.
- Close the *Properties Dialog* and change the values of the sliders in order to find out how each color component influences the resulting color of the polygon.
  Note: You could also animate the sliders with different speeds in order to see the color of the polygon change automatically.

5.5. JavaScript Interface

Note: GeoGebra's JavaScript interface is interesting for users who have some experience in HTML editing.

In order to enhance your *Dynamic Worksheets* and increase their interactivity, GeoGebra applets provide a *JavaScript interface*. For example, you could create a button to randomly generate new configurations of a dynamic construction.

Please, see the document GeoGebra Applets and JavaScript (http://www.geogebra.org in *Help*) for examples and information about using JavaScript with GeoGebra applets.

5.6. Keyboard Shortcuts

Key	**[plain]**	**Ctrl (Mac OS: Cmd)**	**Ctrl-Shift (Mac OS: Cmd-Shift)**	**Alt (Mac OS: Ctrl)**
A		Select All	Show / hide *Algebra View*	alpha α
B				beta β
C		Copy (Spreadsheet View)	Export Graphics View to Clipboard	
D				delta δ
E		*Properties Dialog*		Euler e
F		Refresh views		phi ϕ
G				gamma γ
H				
I				
J				
K				
L		Select current layer		lambda λ
M				mu μ
N		New window		
O		Open		degree symbol °
P		Print preview	Export Graphics View as Picture (png, eps)…	pi π
Q		Select descendants	Select ancestors	
R				
S		Save	Show / hide *Spreadsheet View*	sigma σ
T			Export as PSTricks	theta θ
U				
V		Paste (Spreadsheet View)		

Key	[plain]	Ctrl (Mac OS: Cmd)	Ctrl-Shift (Mac OS: Cmd-Shift)	Alt (Mac OS: Ctrl)
W		Close (Mac OS only)	Export Dynamic Worksheet as Webpage (html)	omega ω
X				
Y		Redo		
Z		Undo		
0				Exponent ^{0}
1				Exponent ^{1}
2				Exponent ^{2}
3				Exponent ^{3}
4				Exponent ^{4}
5				Exponent ^{5}
6				Exponent ^{6}
7				Exponent ^{7}
8				Exponent ^{8}
9				Exponent ^{9}
-	Decrease selected number / angle	Zoom Out		minus-or-plus
+	Increase selected number / angle	Zoom In		plus-or-minus ±
=	Increase selected number / angle	Zoom In		not-equal-to ≠
<				less-than-or-equal-to ≤
, (comma)				less-than-or-equal-to ≤
>				greater-than-or-equal-to ≥
. (period)				greater-than-or-equal-to ≥
*				
F1	Help			
F2	Start editing selected object (*Algebra View*)			
F3	Insert definition of selected object (Input Bar)			
F4	Insert value of selected object (Input Bar)			

Key	**[plain]**	**Ctrl (Mac OS: Cmd)**	**Ctrl-Shift (Mac OS: Cmd-Shift)**	**Alt (Mac OS: Ctrl)**
F5	Insert name of selected object (Input Bar)			
F9	Update random numbers			
Enter	Toggle focus between *Graphics View* and *Input Bar*			
Left-click				
Right-click (Mac OS: *Ctrl*-click) in *Graphcis View*	Click: Open *Context Menu* (on object) Properties Dialog of *Graphics View* (on background) Click and drag: Fast Drag Mode (on object) Zoom rectangle (on background)			
Scroll Wheel	Zoom in / out	Zoom in / out (Applet)		Accelerated Zoom in / Zoom out
Delete	Delete current selection			
Backspace	Delete current selection			
Up arrow ↑	Increase selected number / angle Move selected point up Go to prior entries in *Input Bar* history Go up in *Construction Protocol*	x10 speed multiplier	x0.1 speed multiplier (press Shift only)	x100 speed multiplier

Key	[plain]	Ctrl (Mac OS: Cmd)	Ctrl-Shift (Mac OS: Cmd-Shift)	Alt (Mac OS: Ctrl)
Right arrow →	Increase selected number/angle Move selected point to the right Go up in *Construction Protocol*	x10 speed multiplier	x0.1 speed multiplier (press Shift only)	x100 speed multiplier
Left arrow ←	Decrease selected number/angle Move selected point to the left Go down in *Construction Protocol*	x10 speed multiplier	x0.1 speed multiplier (press Shift only)	x100 speed multiplier
Down arrow ↓	Decrease selected number/angle Move selected point down Go to newer entry in *Input Bar* history Go down in *Construction Protocol*	x10 speed multiplier	x0.1 speed multiplier (press Shift only)	x100 speed multiplier
Home/PgUp	Go to first item in *Construction Protocol*			
End/PgDn	Go to last item in *Construction Protocol*			

Additional keyboard commands:

- *Alt-Shift* (Mac OS: *Ctrl-Shift*): Upper-case Greek letters
- *Spreadsheet View*: Ctrl-Alt-C copies values (not the formulae)

Note: The degree symbol ° (*Alt-O*, Mac OS: *Ctrl-O*) and the symbol π for pi (*Alt-P*, Mac OS: *Ctrl-P*) can also be used in the slider dialog window for interval (min, max) and the increment.

5.7. Labels and Captions

Show and Hide Labels

You can show or hide the labels of objects in the *Graphics View* in different ways:

- Select the tool *Show / Hide Label* and click on the object whose label you would like to show or hide.
- Open the *Context Menu* for the desired object and select *Show Label*.
- Open the *Properties Dialog* for the desired object and check or un-check the checkbox *Show Label* on tab *Basic*.

Name and Value

In GeoGebra, every object has a unique name that can be used to label the object in the *Graphics View*. In addition, an object can also be labeled using its value or its name and value. You can change this label setting in the *Properties Dialog* on tab *Basic* by selecting the corresponding option *Name*, *Value*, or *Name & Value* from the drop down menu next to the checkbox *Show Label*.
Note: The value of a point is his coordinates, while the value of a function is its equation.

Caption

However, sometimes you might want to give several objects the same label, for example, to label the four edges of a square *a*. In this case, GeoGebra offers captions for all objects in addition to the three labeling options mentioned above. You can set the caption of an object on tab *Basic* of the *Properties Dialog* by entering the desired caption into the text field called *Caption*. Afterwards, you can select the labeling option *Caption* from the drop down menu next to the checkbox *Show Label*.

5.8. Layers

Note: In GeoGebra, layers are used to determine which object to select or drag when the user clicks on multiple objects at the same time.

By default, all objects are drawn on layer 0, which is basically the *background* layer of the *Graphics View*. A total of 10 layers are available (numbers 0 to 9) and higher numbered layers are drawn on top of lower numbered layers.

Using the *Advanced* tab of the *Properties Dialog*, you can change the layer for a certain object (layers from 0 to 9 available). Once you change the layer number for at least one object to be different from layer 0 (e. g., to layer 3), all new objects will be drawn on the layer with the highest number that is used for any object.

Note: After selecting any object, you can select all objects in the same layer by selecting item *Select Current Layer* (keyboard shortcut: *Ctrl-L*) from the *Edit* menu. This menu item is only available if all selected objects lie on the same layer.

Further use of layers:

- For SVG export objects are grouped by layer.
- Layers can be controlled using the JavaScript Interface for GeoGebra applets.

5.9. Redefine

Redefining objects is a very versatile tool to change a construction. Please note that this may also change the order of the construction steps in the *Construction Protocol*.

In GeoGebra, an object may be *redefined* in different ways:

- Select the *Move* tool and double click on any object in the *Algebra View*.
  - For *free objects* an editing field is opened allowing you to directly change the algebraic representation of the object. Hit the *Enter*-key in order to apply these changes.
  - For *dependent objects* the *Redefine* dialog is opened allowing you to redefine the object.
- Select the *Move* tool and double click on any object in the *Graphics View*. This opens the *Redefine* dialog and allows you to redefine the object.
- Change any object by entering its name and the new definition into the *Input Bar*.
- Open the *Properties Dialog* and change the definition of an object on tab *Basic*.

Note: Fixed objects cannot be redefined. In order to redefine a fixed object, you need to free it first using tab *Basic* of the *Properties Dialog*.

Examples:

- In order to place a free point *A* on an existing line *h*, you first need to double click on the point *A* to open the *Redefine* dialog window. Then, enter the command `Point[h]` in the appearing text field and press the *Enter*-key. To remove point *A* from this line and make it free again, you need to *redefin*e it to some free coordinates like `(1, 2)`.
- Another example *is* the conversion of a line *h* through two points *A* and *B* into a segment. Open the *Redefine dialog* for line *h* and enter the command `Segment[A, B]` in the appearing text field.

5.10. Trace and Locus

Objects can leave a trace in the *Graphics View* when they are moved. Use the *Context Menu* to switch this *Trace On*. Then, modify the construction so that the object whose trace you turned on changes its position and leaves a trace.

Note: You can turn off the trace of an object by un-checking Trace On in the *Context Menu*. The menu item *Refresh Views* in the *View* menu clears all traces.

You can also let GeoGebra automatically create the locus of a point by either using tool *Locus* with the mouse, or enter the command *Locus* into the Input field.
Note: The point whose locus you would like to create must depend on another point's movement, which is restricted to move along an object (e. g., line, segment, circle).

Example:

- Create a segment *a* between the points *A = (-1, -1)* and *B = (1, -1)*.
- Place a point *C* on the segment, so it is restricted to move along segment *a*.
- Create a point *P* that depends on point *C* (e. g., `P = (x(C), x(C)^2)`).
- Use either tool or command *Locus* in order to create the locus of point *P* in dependence on point *C*:
    - Tool *Locus*: Click first on point *P* and then on point *C*.
    - Command *Locus*: Enter `Locus[P, C]` into the Input Bar and hit the Enter-key.
- Note: The locus created in this example is the graph of a parabola on the interval [*-1, 1*].

6. Index

A

About / License, Help Menu ... 78
Absolute value, direct input ... 35
Addition, direct input ... 35
Affine Ratio, command ... 41
Algebra View ... 7
  Modify objects ... 7
  Show/hide, View Menu ... 73
Algebraic input, Input Bar ... 30
Angle
  Allow reflex angle ... 24
  Angle Bisector, command ... 49
  Angle Bisector, tool ... 20
  Angle with Given Size, tool ... 24
  Angles in polygon ... 45
  Command ... 45
  Direct input ... 32
  Limit size ... 24
  Limit value ... 33
  Reflex ... 33
  Tool ... 23
Angle Unit
  Options Menu ... 74
  Settings ... 12
Animation ... 79
  Animation Button ... 79
  Animation On ... 79
  Animation On, Context Menu ... 10
  Arrow keys ... 79
  Automatic animation ... 79
  Cycle ... 79
  Decreasing ... 79
  Increasing ... 79
  Increment ... 80
  Manual ... 79
  Oscillating ... 79
  Pause ... 79
  Plus / Minus keys ... 79
  Repeat ... 79
  Speed ... 79
Antihyperbolic
  Antihyperbolic cosine, direct input ... 35
  Antihyperbolic sine, direct input ... 35
  Antihyperbolic tangent, direct input ... 36
Append, command ... 58
Arc
  Arc cosine, direct input ... 35
  Arc sine, direct input ... 35
  Arc tangent, direct input ... 35
  Circular Arc with Center between Two Points, tool ... 23
  Circular Arc, command ... 54
  Circumcircular Arc through Three Points, tool ... 23
  Circumcircular Arc, command ... 55
  Command ... 54
Area
  Area between two functions ... 42
  Command ... 41
  Definite integral ... 41, 42
  Tool ... 24
Arithmetic operations ... 35
Arrow keys ... 33
  Animation ... 79
Asymptote, command ... 49
Automatic animation ... 79
Auxiliary Objects ... 7, 8
  View Menu ... 73
Axes
  Command ... 49
  Customize ... 9
  Direct input ... 34
  Line style ... 9
  Ratio ... 9
  Scale ... 17
  Scaling ... 9
  Show / hide ... 9
  Show / hide individually ... 9
  Tickmarks ... 9
  Units ... 9
  View Menu ... 73
  xAxis and yAxis ... 34
AxisStep, command ... 41

B

Background image ... 29
Bar Chart, command ... 63
Best Fit Line, tool ... 20
Binomial Coefficient, command ... 41
Bisector
  Angle Bisector, command ... 49
  Angle Bisector, tool ... 20
  Perpendicular Bisector, command ... 50
  Perpendicular Bisector, tool ... 21
Boolean
  Boolean Variable, tool ... 25
  Operations ... 36
  Show/hide object ... 36
  Variables ... 36
Box Plot, command ... 64
Breakpoint ... 12

C

Caption ... 87
  Label ... 87
  Name ... 87
  Name & Value ... 87
Cartesian coordinates ... 33

Ceiling, direct input ... 36
Cell
  Cell Range, command ... 67
  Input ... 8
  Name ... 8
Center, command ... 45
Centre, command ... 45
Centroid, command ... 45
Checkbox
  Checkbox to Show/Hide Objects, tool ... 25
  Size, Options Menu ... 75
  Size, Settings ... 12
Circle
  Circle through Three Points, tool ... 22
  Circle with Center and Radius,Tool ... 21
  Circle with Center through Point, tool ... 22
  Command ... 51
  Osculating Circle, command ... 52
Circumference, command ... 41
Close, File Menu ... 71
Color
  Dynamic ... 82
  Properties Dialog ... 10
Column
  Column Name, command ... 67
  Command ... 67
Command
  AffineRatio ... 41
  Angle ... 45
  AngleBisector ... 49
  Append ... 58
  Arc ... 54
  Area ... 41
  Asymptote ... 49
  Axes ... 49
  AxisStep ... 41
  BarChart ... 63
  BinomialCoefficient ... 41
  BoxPlot ... 64
  CellRange ... 67
  Center ... 45
  Centre ... 45
  Centroid ... 45
  Circle ... 51
  CircularArc ... 54
  CircularSector ... 54
  CircumcircularArc ... 55
  CircumcircularSector ... 55
  Circumference ... 41
  Column ... 67
  ColumnName ... 67
  Conic ... 51
  ConjugateDiameter ... 49
  ConstructionStep ... 40
  Corner ... 46
  CorrelationCoefficient ... 64
  CountIf ... 58
  Covariance ... 64
  CrossRatio ... 41
  Curvature ... 42, 54
  CurvatureVector ... 47, 54
  Curve ... 53
  Delete ... 40
Derivative ... 52, 54
Determinant ... 67
Dilate ... 61
Direction ... 47
Directrix ... 49
Distance ... 42
Div (Division) ... 42
Element ... 58
Ellipse ... 51
Enlarge ... 61
Expand ... 52
Extremum ... 46
Factor ... 52
Factorise ... 52
First ... 58
FitExp ... 64
FitLine ... 64
FitLineX ... 64
FitLog ... 64
FitLogistic ... 64
FitPoly ... 64
FitPow ... 64
FitSin ... 64
Focus ... 46
FormulaText ... 55
FractionText ... 56
Function ... 53
GCD ... 42
Greatest Common Divisor ... 42
HCF ... 42
Highest Common Factor ... 42
Histogram ... 65
Hyperbola ... 51
If (Boolean command) ... 40
InflectionPoint ... 46
Insert ... 59
IntegerDivision ... 42
Integral ... 42, 53
Intersect ... 46
Intersection ... 59
InverseNormal ... 65
Invert ... 68
IsDefined ... 40
IsInteger ... 41
Iteration ... 42
IterationList ... 59
Join ... 59
KeepIf ... 60
Last ... 60
LCM ... 43
Least Common Multiple ... 43
Length ... 43, 54, 60
LetterToUnicode ... 56
Line ... 49
LinearEccentricity ... 43
LowerSum ... 43
Lowest Common Multiple ... 43
MajorAxis ... 50
Max ... 43, 60
Maximum ... 43, 60
Mean ... 65
MeanX ... 65
MeanY ... 65

Median ... 65
Midpoint ... 47
Min ... 43, 60
Minimum ... 43, 60
MinorAxis ... 50
Mod ... 43
Mode ... 66
Modulo Function ... 43
Name ... 56
Normal ... 66
Object ... 56
OsculatingCircle ... 52, 54
Parabola ... 52
Parameter ... 44
Perimeter ... 44
PerpendicularBisector ... 50
PerpendicularLine ... 50
PerpendicularVector ... 48
Point ... 47
Polar ... 50
Polygon ... 49
Polynomial ... 53
Product ... 60
Q1 (1. Quartile) ... 66
Q3 (3. Quartile) ... 66
Radius ... 44
Random ... 44
RandomBetween ... 44
RandomBinomial ... 44
RandomNormal ... 44
RandomPoisson ... 44
Ray ... 48
Reflect ... 62
Relation ... 40
Remainder ... 43
RemoveUndefined ... 60
Reverse ... 60
Root ... 47
Rotate ... 62
Row ... 67
SD (standard deviation) ... 66
Sector ... 55
Segment ... 48
Semicircle ... 55
SemiMajorAxisLength ... 44
SemiMinorAxisLength ... 44
Sequence ... 60
SigmaXX ... 66
SigmaXY ... 66
SigmaYY ... 66
Simplify ... 53
Slope ... 44
Sort ... 61
Sum ... 61
Sxx. ... 66
Sxy ... 66
Syy ... 66
TableText ... 56
Take ... 61
Tangent ... 50, 54
TaylorPolynomial ... 53
Text ... 57
TextToUnicode ... 57
Translate ... 63
Transpose ... 68
TrapeziumSum ... 44
TrapezoidalSum ... 44
TurningPoint ... 46
UnicodeToLetter ... 57
UnicodeToText ... 58
Union ... 61
UnitPerpendicularVector ... 48
UnitVector ... 48
UpperSum ... 45
Variance ... 67
Vector ... 48
Vertex ... 47
Commands ... 7, 39
Accept suggestion ... 40
Angle Command ... 45
Arc Commands ... 54
Automatic completion ... 39
Boolean Commands ... 40
Change suggestion ... 40
Command list, View Menu ... 74
Command syntax, Help ... 7
Conic Section Commands ... 51
Curve Command ... 53
Function Commands ... 52
General commands ... 40
Geometric Transformation Commands ... 61
Help for commands ... 7
Line Commands ... 49
List Commands ... 58
List of commands ... 7
Locus Command ... 58
Matrix Commands ... 67
Number Commands ... 41
Parametric Curve Command ... 53
Point Commands ... 45
Polygon Command ... 49
Ray Command ... 48
Sector Commands ... 54
Segment Command ... 48
Sequence Commands ... 58
Spreadsheet Commands ... 67
Statistics Commands ... 63
Text Commands ... 55
Transformation Commands ... 61
Vector Commands ... 47
Compass, tool ... 22
Compasses, tool ... 22
Complex number
Addition ... 39
Direct input ... 38
Multiplication ... 39
Operations ... 38
Subtraction ... 39
Conditional functions ... 52
Conditional visibility ... 80
Existing objects ... 80
New objects ... 80
Conic section
Conic through Five Points, tool ... 22
Conic, command ... 51
Direct input ... 34

Name 30, 34
Conjugate Diameter, command 49
Construction Protocol 11
Breakpoint 12
Change order 12
Columns 12, 13
Export as webpage 12
Export colorful Construction Protocol 12
Insert new step 12
Modify 11
Navigate 11
Print 13
View Menu 74
Construction step
Construction Protocol 11
Navigation Bar 11
Construction Step
Command 40
Construction Tools 6, 16
Activate 16
Context Menu 10
Animation On 10
Copy to Input Bar 10
Delete 10
Rename 10
Trace On 10, 88
Trace to Spreadsheet 10
Continuity, Options Menu 75
Convert
Degree to radians 32
Radians to degree 32
Coordinate axes
Customize 9
Show / hide 9
View Menu 73
Coordinate grid
Customize 9
Show / hide 9
View Menu 73
Coordinates 33
Cartesian coordinates 33
Coordinate Style, Settings 12
Coordinates style, Options Menu 75
Polar coordinates 33
x-coordinate 35
y-coordinate 35
Copy
Copy to Input Bar, Context Menu 10
Copy Visual Style, tool 17
Corner, command 46
Correlation Coefficient, command 64
Cosine, direct input 35
Count If, command 58
Covariance, command 64
Create New Tool, Tools Menu 76
Cross Ratio, command 41
Cubic root, direct input 35
Curvature
Command 42, 54
Curvature Vector, command 47, 54
Curve
Command 53
Customize
Graphics View 8
Settings 12
Toolbar 9, 76
User interface 8

D

Decimal places, Options Menu 75
Decimal point 32
Definite integral 42
Area 42
Area between two functions 42
Definition, insert in Input Bar 31
Degree
Convert to radians 32
Symbol 24, 32
Delete
Command 40
Context Menu 10
Delete Object, tool 17
Edit Menu 72
Trace 74
Dependent object 7
Derivative, command 52, 54
Determinant, command 67
Dilate
Command 61
Dilate Object from Point by Factor, tool 25
Direct input
Angles 32
Axes 34
Conic sections 34
Functions 34
Lines 33
Numbers 32
Points 33
Vectors 33
Direction, command 47
Directrix, command 49
Distance
Command 42
Distance or Length, tool 24
Div, command 42
Division, direct input 35
Drawing Pad, Options Menu 76
Dynamic colors 82
Dynamic text 27
Dynamic Worksheet
Edit applet 15
Edit text 15
Export 14
Export, File Menu 70
Java 15

E

Edit Menu 71
Delete 72
Export Graphics View to Clipboard 72
Properties Dialog 73
Redo 71
Select All 72
Select Ancestors 72

Select Current Layer ... 72
Select Descendants ... 72
Undo ... 71
Edit object ... 31
Element, command ... 58
Ellipse
Command ... 51
Tool ... 22
EMF format ... 71
Enlarge
Command ... 61
Enlarge Object from Point by Factor, tool ... 25
EPS format ... 70
Euler constant ... 32
Expand
Command ... 52
Polynomial ... 53
Exponential function, direct input ... 35
Exponentiation, direct input ... 35
Export
Construction Protocol as webpage ... 12
Dynamic Worksheet ... 14
Files created ... 15
GGB files ... 15
Height / width ... 14
HTML file ... 15
JAR files ... 15
Show reset icon ... 14
Show Toolbar ... 14
Tab Advanced ... 14
Tab General ... 14
Dynamic Worksheet as Webpage, File Menu ... 70
File Menu ... 70
Graphics View
as PGF/TikZ, FileMenu ... 71
as Picture ... 13
as Picture, File Menu ... 70
as PSTricks, File Menu ... 71
at scale ... 14
to clipboard ... 14
to Clipboard, Edit Menu ... 72
to Clipboard, File Menu ... 71
Interactive webpage ... 14
Interactive worksheet ... 14
Export rectangle ... 14
Extremum, command ... 46

F

Factor, command ... 52
Factorial, direct input ... 35
Factorise, command ... 52
File Menu ... 69
Close ... 71
Export
Dynamic Worksheet as Webpage (html) ... 70
Graphics View as PGF/TikZ ... 71
Graphics View as Picture (png, eps) ... 70
Graphics View as PSTricks ... 71
Graphics View to Clipboard ... 71
New ... 69
New Window ... 69
Open ... 69
Print Preview ... 70
Save ... 69
Save As ... 70
Filling, Properties Dialog ... 10
First axis
Command ... 50
First Axis Length, command ... 44
First, command ... 58
Fit commands
FitExp, command ... 64
FitLog, command ... 64
FitLogistic, command ... 64
FitPoly, command ... 64
FitPow, command ... 64
FitSin, command ... 64
Fit Line
Command ... 64
Fit Line X, command ... 64
Floor, direct input ... 36
Focus, command ... 46
Font Size
Increase ... 13
Options Menu ... 75
Format, Copy Visual Style ... 17
Formula ... 27
Formula Text, command ... 55
Fraction Text, command ... 56
Free object ... 7
Function, command ... 53
Functions
Direct input ... 34
Exponential function ... 35
Limit to interval ... 35
Name ... 30
Pre-defined functions ... 35

G

Gamma function, direct input ... 35
GCD, command ... 42
GeoGebra Forum, Help Menu ... 78
GeoGebraWiki, Help Menu ... 78
Geometric Transformations
Tools ... 25
Graphics View ... 6
Create pictures ... 13
Customize ... 8
Export as picture ... 13
Export at scale ... 14
Export to Clipboard ... 14
Export to Clipboard, Edit Menu ... 72
Geometric input ... 16
Move objects ... 7
Print ... 13
Update Print Preview ... 13
Zoom ... 8
Greatest Common Divisor, command ... 42
Greek letter, upper case ... 86
Grid
Color ... 9
Customize ... 9
Distance of lines ... 9
Isometric ... 9

Line style ... 9
Show / hide ... 9
View Menu ... 73

H

HCF, command ... 42
Help
Command syntax ... 7
Input Bar ... 7, 31
Toolbar ... 6
Help Menu ... 77
About / License ... 78
GeoGebra Forum ... 78
GeoGebraWiki ... 78
License ... 78
www.geogebra.org ... 77
Highest Common Factor, command ... 42
Histogram, command ... 65
Horizontal Splitting, View Menu ... 73
Hyperbola
Command ... 51
Tool ... 22
Hyperbolic
Hyperbolic cosine, direct input ... 35
Hyperbolic sine, direct input ... 35
Hyperbolic tangent, direct input ... 35

I

If
Command ... 40, 52
Conditional function ... 52
Image
Corner ... 46
Distort ... 28
Insert Image, tool ... 28
Properties Dialog ... 28
Rotate ... 28
Scale ... 28
Specify corners ... 28
Imaginary unit ... 39
Increment, manual animation ... 80
Indices ... 30, 39
Inflection Point, command ... 46
Input Bar ... 7, 30
Direct input ... 32
Help ... 7, 31
History ... 31
Show input ... 31
View Menu ... 73
Insert
Command ... 59
Definition in Input Bar ... 31
Insert Image, tool ... 28
Insert Text, tool ... 26
Name in Input Bar ... 31
Value in Input Bar ... 31
Insert Image, tool ... 28
Insert Text, tool ... 26
Integer Division, command ... 42
Integral
Command ... 42, 53
Definite integral ... 42
Indefinite integral ... 53
Interactive webpage, Export ... 14
Interactive worksheet, Export ... 14
Interface, customize ... 8
Intersect
All intersection points ... 18
Allow outlying intersection points ... 18
Command ... 46
Intersect Two Objects, tool ... 18
One intersection point ... 18
Intersection, command ... 59
Inverse Normal, command ... 65
Invert, command ... 68
Is Defined, command ... 40
Is Integer, command ... 41
Iteration
Command ... 42
Iteration List, command ... 59

J

JavaScript interface ... 82
Join, command ... 59

K

Keep If, command ... 60
Keyboard shortcuts ... 83

L

Labeling
Options Menu ... 75
Settings ... 12
Labels ... 87
Caption ... 87
Name and value ... 87
Show / hide ... 87
Language, Options Menu ... 76
Last, command ... 60
LaTeX Formula ... 27
Layers ... 87
LCM, command ... 43
Least Common Multiple, command ... 43
Length
Command ... 43, 54, 60
Distance or Length, tool ... 24
Letter To Unicode, command ... 56
License, Help Menu ... 78
Limit
Function to interval ... 35
Value of angle ... 33
Value of number ... 33
Line
Command ... 49
Direct input ... 33
Line style, Properties Dialog ... 10
Line thickness, Properties Dialog ... 10
Line through Two Points, tool ... 20
Name ... 30, 33
Perpendicular Line, command ... 50
Perpendicular Line, tool ... 21

Linear Eccentricity, command ... 43
List
  Direct Input ... 37
  Operations ... 37
Lists
  Addition ... 37
  Apply arithmetic operations ... 37
  Apply pre-defined functions ... 37
  Compare ... 37
  Division ... 37
  Multiplication ... 37
  Pre-defined functions ... 37
  Subtraction ... 37
Locus ... 88, 89
  Command ... 58
  Tool ... 25
Logarithm
  base 10, direct input ... 35
  base 2, direct input ... 35
  natural, direct input ... 35
Lower Sum, command ... 43
Lowest Common Multiple, command ... 43

M

Major Axis, command ... 50
Manage Tools, Tools Menu ... 76
Manual animation ... 79
Matrices
  Addition ... 38
  Apply arithmetic operations ... 38
  Commands ... 38
  Direct input ... 38
  Multiplication ... 38
  Operations ... 38
  Subtraction ... 38
Max, command ... 43, 60
Maximum, command ... 43, 60
Mean
  Command ... 65
  MeanX, command ... 65
  MeanY, command ... 65
Median, command ... 65
Menu
  Edit ... 71
  File ... 69
  Help ... 77
  Options ... 74
  Tools ... 76
  View ... 73
  Window ... 77
Menu items ... 69
Midpoint
  Command ... 47
  Midpoint or Center, tool ... 19
Min, command ... 43, 60
Minimum, command ... 43, 60
Minor Axis, command ... 50
Mixed text ... 27
Mod, command ... 43
Mode, command ... 66
Modulo Function, command ... 43
Move
  Move Drawing Pad, tool ... 17
  Tool ... 17
Multiplication, direct input ... 35

N

Name
  Command ... 56
  Conic section ... 34
  Insert in Input Bar ... 31
  Line ... 33
  of conic section ... 30
  of function ... 30
  of line ... 30
  of point ... 30
  of vector ... 30
  Point ... 33
  Vector ... 33
Naming objects ... 30
Navigation Bar ... 11
  View Menu ... 74
New Point, tool ... 19
New Window
  Menu ... 69
  Window Menu ... 77
New, Menu ... 69
Normal, command ... 66
Number
  Direct input ... 32
  Limit value ... 33
  Name ... 32

O

Object
  Algebraic notation ... 10
  Animation On ... 10
  Auxiliary object ... 7
  Change value ... 30
  Color ... 10
  Command ... 56
  Copy to Input Bar ... 10
  Delete ... 10
  Dependent object ... 7
  Edit ... 31
  Fast renaming ... 16
  Filling ... 10
  Free object ... 7
  Insert
    Definition in Input Bar ... 31
    Name in Input Bar ... 31
    Value in Input Bar ... 31
  Line style ... 10
  Line thickness ... 10
  Name ... 30
  Rename ... 10
  Select ... 16
  Select several objects ... 16
  Show / hide ... 8
  Size ... 10
  Trace On ... 10
  Visibility ... 10
Open, Menu ... 69

Options Menu 74
Angle Unit 74
Checkbox Size 75
Continuity 75
Coordinates Style 75
Decimal Places 75
Drawing Pad 76
Font Size 75
Labeling 75
Language 76
Point Capturing 74
Point Style 75
Restore Default Settings 76
Right Angle Style 75
Rounding 75
Save Settings 76
Significant Figures 75
Osculating Circle, command 54

P

Parabola
Command 52
Tool 22
Parallel Line, tool 21
Parameter, command 44
Parametric Curve, command 53
Parentheses, direct input 35
Pause animation 79
PDF format 71
Perimeter, command 44
Perpendicular
Perpendicular Bisector, command 50
Perpendicular Bisector, tool 21
Perpendicular Line, command 50
Perpendicular Line, tool 21
Perpendicular Vector, command 48
PGF/TikZ
Export, File Menu 71
Format 71
Pi
Constant 32
Symbol 24, 32
Picture
Background image 29
Distort 28
EMF format 71
EPS format 70
PDF format 71
PGF/TikZ 71
PNG format 70
PSTricks format 71
Rotate 28
Scale 28
Specify corners 28
SVG format 71
Transparency 29
Picture export, File Menu 70
Plus / Minus keys, Animation 79
PMCC, command 64
PNG format 70
Point
Command 47
Direct input 33
Name 30, 33
Point Capturing, Options Menu 74
Point Style
Options Menu 75
Settings 12
Polar
Command 50
Polar or Diameter Line, tool 21
Polar coordinates 33
Polygon
Angles 45
Command 49
Tool 20
Polynomial, command 53
Pre-defined functions 35
Principal axis
Command 50
Principal axis length, command 44
Print
Construction Protocol 13
Graphics View 13
Options 13
Print Preview
Menu 70
Update 13
Product moment correlation coefficient, command 64
Product, command 60
Properties Dialog 10
Edit Menu 73
for objects 10
Image 28
of Graphics View 9
Select all objects 10
Tabs 10
Protocol 11
Protocol, export as webpage 12
PSTricks
Export, File Menu 71
Format 71

Q

Quartile
Q1, command 66
Q3, command 66

R

Radians, convert to degree 32
Radius, command 44
Random
Command 44
Direct input 35
New random numbers 74
Random Between, command 44
Random Binomial, command 44
Random Normal, command 44
Random Poisson, command 44
Ray
Command 48
Ray through Two Points, tool 20
Tool 20

Recompute All Objects, View Menu 74
Record to Spreadsheet
Context Menu 10
Tool 17
Redefine 88
Fixed object 88
Redo, Edit Menu 71
Reflect
Command 62
Reflect Object about Line, tool 26
Reflect Object about Point, tool 26
Reflect Object in Line, tool 26
Reflect Object in Point, tool 26
Reflect Point about Circle, tool 26
Reflect Point in Circle, tool 26
Reflex angle 33
Refresh Views, View Menu 74
Regular Polygon, tool 20
Relation
Command 40
Tool 17
Remainder, command 43
Remove Undefined, command 60
Rename
Context Menu 10
Fast renaming option 16
Restore
Default settings 12
Default Settings, Options Menu 76
Default Toolbar 9
Reverse, command 60
Right Angle Style
Options Menu 75
Settings 12
Root, command 47
Rotate
Command 62
Rotate around Point, tool 18
Rotate Object around Point by Angle, tool 26
Rounding
Direct input 36
Options Menu 75
Row, command 67

S

Save
Menu 69
Save As, Menu 70
Save Settings, Options Menu 76
Settings 12
Scalar product, direct input 35
SD, command 66
Second axis
Command 50
Second axis length, command 44
Sector
Circular Sector with Center between Two Points Tool 23
Circular Sector, command 54
Circumcircular Sector through Three Points, tool 23
Circumcircular Sector, command 55
Command 55
Segment
Command 48
Segment between Two Points, tool 19
Segment with Given Length from Point, tool 19
Select
Object 16
Select All, Edit Menu 72
Select Ancestors, Edit Menu 72
Select Current Layer, Edit Menu 72
Select Descendants, Edit Menu 72
Several Objects 16
Select All, Properties Dialog 10
Selection rectangle 16
Semicircle
Command 55
Tool 23
Semimajor Axis Length, command 44
Semiminor Axis Length, command 44
Sequence
Command 60
Settings
Angle Unit 12
Checkbox Size 12
Coordinate Style 12
Customize 12
Increase Font Size 13
Labeling 12
Point Style 12
Restore Default Settings 12
Right Angle Style 12
Save 12
Show / Hide
Object 8
Show / Hide Label, tool 18
Show / Hide Object, tool 18
Sigma
SigmaXX, command 66
SigmaXY, command 66
SigmaYY, command 66
Sign, direct input 35
Significant Figures, Options Menu 75
Simplify
Command 53
Polynomial 53
Sine, direct input 35
Size, Properties Dialog 10
Slider 33
Tool 24
Slope
Command 44
Tool 25
Sort, command 61
Spreadsheet View 8
Cell input 8
Cell name 8
View Menu 73
Square root, direct input 35
Standard deviation, command 66
Static text 27
Subtraction, direct input 35
Sum, command 61
SVG format 71
Sxx, command 66

Sxy, command ... 66
Syy, command ... 66

T

Table Text, command ... 56
Take, command ... 61
Tangent
  Command ... 50, 54
  Tool ... 21
Tangent, direct input ... 35
Taylor Polynomial, command ... 53
Text
  Command ... 57
  Dynamic text ... 27
  Formula ... 27
  Insert Text, tool ... 26
  LaTeX Formula ... 27
  Mixed text ... 27
  Static text ... 27
Text To Unicode, command ... 57
Tool
  Angle Bisector ... 20
  Angle with Given Size ... 24
  Area ... 24
  Best Fit Line ... 20
  Checkbox to Show/Hide Objects ... 25
  Circle through Three Points ... 22
  Circle with Center and Radius ... 21
  Circle with Center through Point ... 22
  Circular Arc with Center between Two Points ... 23
  Circular Sector with Center between Two Points ... 23
  Circumcircular Arc through Three Points ... 23
  Circumcircular Sector through Three Points ... 23
  Compass ... 22
  Compasses ... 22
  Conic through Five Points ... 22
  Delete Object ... 17
  Dilate Object from Point by Factor ... 25
  Distance or Length ... 24
  Ellipse ... 22
  Enlarge Object from Point by Factor ... 25
  Hyperbola ... 22
  Insert Image ... 28
  Insert Text ... 26
  Intersect Two Objects ... 18
  Line through Two Points ... 20
  Locus ... 25
  Midpoint or Center ... 19
  Move ... 17
  Move Drawing Pad ... 17
  New Point ... 19
  Parabola ... 22
  Parallel Line ... 21
  Perpendicular Bisector ... 21
  Perpendicular Line ... 21
  Polar or Diameter Line ... 21
  Polygon ... 20
  Ray through Two Points ... 20
  Record to Spreadsheet ... 17
  Reflect Object about Line ... 26
  Reflect Object about Point ... 26
  Reflect Object in Line ... 26
  Reflect Object in Point ... 26
  Reflect Point about Circle ... 26
  Reflect Point in Circle ... 26
  Regular Polygon ... 20
  Relation ... 17
  Rotate around Point ... 18
  Rotate Object around Point by Angle ... 26
  Segment between Two Points ... 19
  Segment with Given Length from Point ... 19
  Semicircle ... 23
  Show / Hide Label ... 18
  Show/Hide Object ... 18
  Slider ... 24
  Slope ... 25
  Tangents ... 21
  Translate Object by Vector ... 26
  Vector between Two Points ... 19
  Vector from Point ... 19
  Zoom In ... 18
  Zoom Out ... 18
Toolbar ... 6
  Customize ... 9, 76
  Help ... 6
  Restore default ... 9
Toolbox ... 7
  Open ... 16
Tools ... 16
  Activate ... 16
  Angle Tool ... 23
  Angle Tools ... 23
  Arc Tools ... 22
  Boolean Variable Tool ... 25
  Conic Section Tools ... 21
  General tools ... 17
  Geometric Transformation Tools ... 25
  Insert Image ... 28
  Insert Text Tool ... 26
  Line Tools ... 20
  Locus Tool ... 25
  Manage tools ... 76
  Number Tools ... 23
  Point tools ... 18
  Polygon Tools ... 20
  Ray Tool ... 20
  Sector Tools ... 22
  Segment Tools ... 19
  Transformation Tools ... 25
  User defined tools ... 76, 81
  Vector Tools ... 19
Tools Menu ... 76
  Create New Tool ... 76
  Customize Toolbar ... 76
  Manage Tools ... 76
Trace ... 88
  Delete ... 74
  Trace On, Context Menu ... 10, 88
Trace to Spreadsheet, Context Menu ... 10
Transformation, tools ... 25
Translate
  Command ... 63
  Translate Object by Vector, tool ... 26
Transparency of picture ... 29
Transpose, command ... 68

Trapezium Sum, command 44
Trapezoidal Sum, command 44
Trigonometric functions 34
Antihyperbolic cosine 35
Antihyperbolic sine 35
Antihyperbolic tangent 36
Arc cosine 35
Arc sine 35
Arc tangent 35
Cosine 35
Hyperbolic cosine 35
Hyperbolic sine 35
Hyperbolic tangent 35
Sine 35
Tangent 35
Turning Point, command 46

U

Undo, Edit Menu 71
Unicode
Unicode To Letter, command 57
Unicode To Text, command 58
Union, command 61
Unit Perpendicular Vector, command 48
Unit Vector, command 48
UpperSum, command 45
User defined tools 76, 81
Create 81
Open 81
Save 81
User interface, customize 8

V

Value
Change 30
Insert in Input Bar 31
Variance, command 67
Vector
Command 48
Direct input 33
Name 30, 33
Perpendicular Vector, command 48
Vector between Two Points, tool 19
Vector from Point, tool 19
Vertex, command 47
View Menu 73
Algebra View 73
Auxiliary Objects 73
Axes 73
Command List 74
Construction Protocol 74
Coordinate Axes 73
Coordinate Grid 73
Grid 73
Horizontal Splitting 73
Input Bar 73
Navigation Bar 74
Recompute All Objects 74
Refresh Views 74
Spreadsheet View 73
Views
Algebra View 7
Graphics View 6
Spreadsheet View 8
Visibility
Conditional 80
Properties Dialog 10
Visual Style, Copy 17

W

Window Menu 77
New Window 77
www.geogebra.org, Help Menu 77

X

xAxis 34
x-coordinate, direct input 35

Y

yAxis 34
y-coordinate, direct input 35

Z

Zoom 8
Center of Zoom 8
Zoom In, tool 18
Zoom Out, tool 18
Zoom rectangle 9