159 Lecture 9 - Basic Syntax and Basic Commands

Note: **To type a command** in a *Mathematica* notebook, use the mouse to move the cursor until it is horizontal, click the left mouse button, and type the command. **To enter a command**, use the mouse to move the cursor until it is vertical and over the command line, then click the left mouse button, and finally press and hold the `Shift` key followed by `Enter` key or use the `Enter` key on the smaller number keypad (lower right-hand corner of keyboard). **To expand cells**, double click on the cell bracket (blue vertical bar to the right) with the downward pointing arrow.

Basic Syntax

1. Constants and Commands

In *Mathematica*, capital letters and built-in functions begin with a capital letter. For example, here is the syntax for **sin x** and ***π*** in *Mathematica.* Enter these commands!

```
In[1]:= Sin[x]
Out[1]= Sin[x]

In[2]:= Pi
Out[2]= π
```

If you try to use a symbol reserved by *Mathematica*, you will get an error message! Try entering the following command.

```
In[3]:= E = 4

        Set::wrsym : Symbol e is Protected. ≫

Out[3]= 4
```

To find out what the capital letter **E** stands for, use this command:

```
In[4]:= ? E
```

E is the exponential constant *e* (base of natural logarithms), with numerical value ≃ 2.71828. ≫

For more detail, use two question marks!

```
In[5]:= ?? E
```

E is the exponential constant *e* (base of natural logarithms), with numerical value ≃ 2.71828. ≫

```
Attributes[e] = {Constant, Protected, ReadProtected}
```

If you are not sure of a how a command is spelled, add a * to the end of a **?** command. Enter the next command to find all the *Mathematica* commands that begin with **E**:

In[6]:= **? E***

▼ **System`**

E	Enter	EventHandlerTag
EdgeForm	EnterExpressionPacket	ExactNumberQ
EdgeLabeling	EnterTextPacket	ExactRootIsolation
EdgeRenderingFunction	Entropy	ExampleData
Editable	EntropyFilter	Except
EditButtonSettings	Environment	ExcludedForms
EditCellTagsSettings	Epilog	Exclusions
EditDistance	Equal	ExclusionsStyle
EffectiveInterestRate	EqualColumns	Exists
Eigensystem	EqualRows	Exit
Eigenvalues	EqualTilde	ExitDialog
Eigenvectors	EquatedTo	Exp
Element	Equilibrium	Expand
ElementData	Equivalent	ExpandAll
Eliminate	Erf	ExpandDenominator
EliminationOrder	Erfc	ExpandFileName
EllipticE	Erfi	ExpandNumerator
EllipticExp	Erosion	ExpectedValue
EllipticExpPrime	ErrorBox	ExpIntegralE
EllipticF	ErrorBoxOptions	ExpIntegralEi
EllipticK	ErrorNorm	Exponent
EllipticLog	ErrorPacket	ExponentFunction
EllipticNomeQ	ErrorsDialogSettings	ExponentialDistribution
EllipticPi	EuclideanDistance	ExponentialFamily
EllipticReducedHalfPeriods	EulerE	ExponentialGeneratingFunction
EllipticTheta	EulerGamma	ExponentialMovingAverage
EllipticThetaPrime	EulerPhi	ExponentPosition
EmitSound	Evaluatable	ExponentStep
EmphasizeSyntaxErrors	Evaluate	Export

Empty	Evaluated	ExportAutoReplacements
EnableConsolePrintPacket	EvaluatePacket	ExportPacket
Enabled	EvaluationCell	ExportString
Encode	EvaluationCompletionAction	Expression
End	EvaluationMonitor	ExpressionCell
EndAdd	EvaluationNotebook	ExpressionPacket
EndDialogPacket	EvaluationObject	ExpToTrig
EndFrontEndInteractionPack⋱ et	EvaluationOrder	ExtendedGCD
EndOfFile	Evaluator	Extension
EndOfLine	EvaluatorNames	ExternalCall
EndOfString	EvenQ	ExternalDataCharacterEnco⋱ ding
EndPackage	EventEvaluator	Extract
EngineeringForm	EventHandler	ExtremeValueDistribution

Use the **?** command to find out what the symbols **N** and **C** stand for in *Mathematica*? How would you denote **cos x** or **arccos x** in *Mathematica*?

In[7]:= **? N**

N[*expr*] gives the numerical value of *expr*.
N[*expr*, *n*] attempts to give a result with *n*-digit precision. ≫

In[8]:= **? C**

C[*i*] is the default form for the i^{th} parameter or
constant generated in representing the results of various symbolic computations. ≫

2. Grouping Symbols

In *Mathematica*, paranthesis **( )** are used for mathematical grouping, square brackets **[ ]** are used for functions and built-in commands, and curly brackets **{ }** are used for lists. If you use the wrong kind of parenthesis or brackets, *Mathematica* will let you know with an error message or the output may not look right! Try entering these commands. Which are the correct syntax?

a. Mathematical Grouping

In[9]:= **(4 + 2) * 7**

Out[9]= 42

In[10]:= **[4 + 2] * 7**

In[10]:= **{4 + 2} * 7**

Out[10]= {42}

b. Function Syntax

In[11]:=
```
Sin[Pi]
```
Out[11]= 0

In[12]:=
```
Sin (Pi)
```
Out[12]= π Sin

In[13]:=
```
Sin {Pi}
```
Out[13]= {π Sin}

c. List Syntax

In[14]:=
```
{1, 2, 3, 4.5, 6, 7, 8, 9, -10}
```
Out[14]= {1, 2, 3, 4.5, 6, 7, 8, 9, -10}

In[15]:=
```
[1, 2, 3, 4.5, 6, 7, 8, 9, -10]
```

In[15]:=
```
(1, 2, 3, 4.5, 6, 7, 8, 9, -10)
```

3. Types of Equals Signs

In *Mathematica*, there are three types of equals signs: ==, =, and :=. The symbol ==, which stands for equality in an equation such as x = 2, is used for solving equations or verifying logical statements. We'll see more examples how == is used below. The symbol = is used to store a value in memory, for example **x = 4** assigns the value 4 to x. The symbol := is a replacement symbol that is used for defining functions. See **Defining Functions** below.

In[15]:=
```
3 == 9 - 6
```
Out[15]= True

In[16]:=
```
3 == 4
```
Out[16]= False

In[17]:=
```
x = 2
```
Out[17]= 2

In[18]:=
```
? x
```

Global`x

x = 2

In[19]:=
```
x * 4
```
Out[19]= 8

To clear a definition of a variable or function, such as x = 2 above, use the **Clear** command.

In[20]:=
```
Clear[x]
```

In[21]:= **? x**

```
Global`x
```

Basic Commands

1. Interactive Environment

After a command is entered, *Mathematica* immediately evaluates the command and is ready to accept more instructions. To denote the number of the command that was entered and the result that *Mathematica* returns, the symbols In[#] and Out[#] are used, where # is the number of the command that was entered.

By using the **%** symbol alone for the output of the last command entered or the **%** symbol in conjunction with a previously entered command's number (i.e. **%#**), the output of a previously entered command can be used in a new command.

2. Calculator Commands

Mathematica can be used just like a calculator. Try the following commands. If we have time, make up your own commands!

In[22]:= **Sin[Pi / 4]**

Out[22]= $\frac{1}{\sqrt{2}}$

In[23]:= **4^14 + 2**

Out[23]= 268 435 458

In[24]:= **% / 31**

Out[24]= $\frac{268435458}{31}$

In[25]:= **%% * $\frac{987}{5}$**

Out[25]= $\frac{264945797046}{5}$

In the next calculation, the output (Out[18]) from 18th command entered (In[18]) is being referenced via the **%18** symbol. Replace the **%18** with **%##** where **Out[##]** is the output from the first command entered in this section (**Sin[Pi/4]**).

In[26]:= **% * %18 + $\frac{47}{309876543}$**

Out[26]= $\left\{\frac{47}{309876543} + \frac{264945797046\ \text{Null}}{5}\right\}$

The above examples show that *Mathematica* leaves answers in fractional form. To get a decimal answer like a calculator gives, use the next commands. What is the difference?

```
In[27]:= N[π]

Out[27]= 3.14159

In[28]:= N[π, 50]

Out[28]= 3.1415926535897932384626433832795028841971693993751
```

3. Symbol Manipulation

Mathematica can also be used to manipulate symbols algebraically to simplify expressions and solve equations. Try the following sample commands.

*Note: The **Clear[x]** command is included to clear the value of x from memory if it has been defined!*

```
In[29]:= Clear[x]

In[30]:= Expand[(x + 3) (x - 3) (4 x^2 - 3 x + 10)]
```

Out[30]= $-90 + 27 x - 26 x^2 - 3 x^3 + 4 x^4$

```
In[31]:= Factor[%]
```

Out[31]= $(-3 + x) (3 + x) (10 - 3 x + 4 x^2)$

```
In[32]:= Solve[% == 0, x]
```

Out[32]= $\left\{\{x \to -3\}, \{x \to 3\}, \left\{x \to \frac{1}{8}\left(3 - i\sqrt{151}\right)\right\}, \left\{x \to \frac{1}{8}\left(3 + i\sqrt{151}\right)\right\}\right\}$

```
In[33]:= N[%]

Out[33]= {{x → -3.}, {x → 3.}, {x → 0.375 - 1.53603 i}, {x → 0.375 + 1.53603 i}}
```

In[34]:= Apart$\left[\frac{x^2 + x}{(x^2 - 9) (x + 4)}\right]$

Out[34]= $\frac{2}{7 (-3 + x)} - \frac{1}{3 + x} + \frac{12}{7 (4 + x)}$

```
In[35]:= Together[%]
```

Out[35]= $\frac{x + x^2}{(-3 + x) (3 + x) (4 + x)}$

4. Graphing Commands

Mathematica can be used to graph functions and lists of data. The main commands used are **Plot** and **ListPlot**.

- **a. Graphing a function with the command Plot**

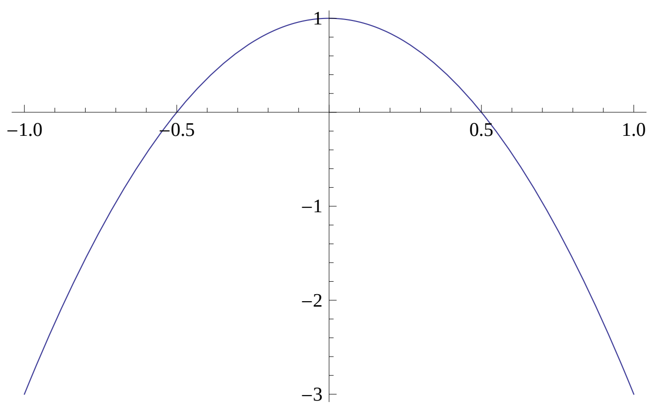
To graph a function of the form **y = f(x) on the x-interval (a,b)**, use the following command, where **f(x), a, and b are replaced with the appropriate function in *Mathematica*'s syntax and numbers are chosen for a and b. The syntax for this command is Plot[*function goes here*,{x,a,b}]**. Here are some examples.

*Note: The **Clear[x]** command is included to clear the value of x from memory if it has been defined!*

```
In[36]:= Clear[x]
```

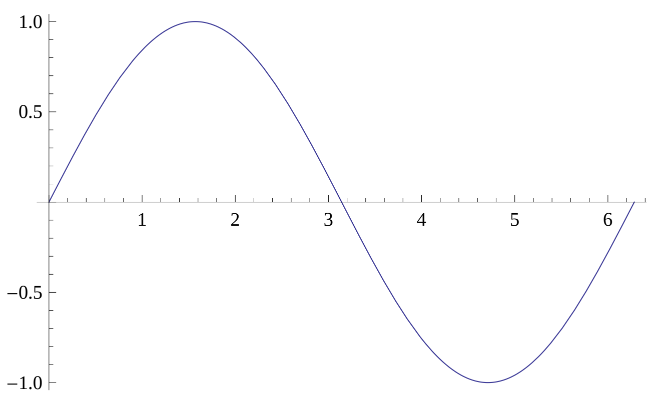
```
In[37]:= Plot[-4 x^2 + 1, {x, -1, 1}]
```

Out[37]=



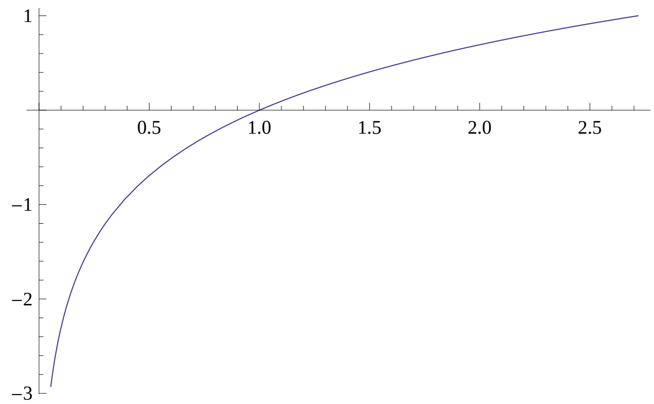

```
In[38]:= Plot[Sin[x], {x, 0, 2 π}]
```

Out[38]=



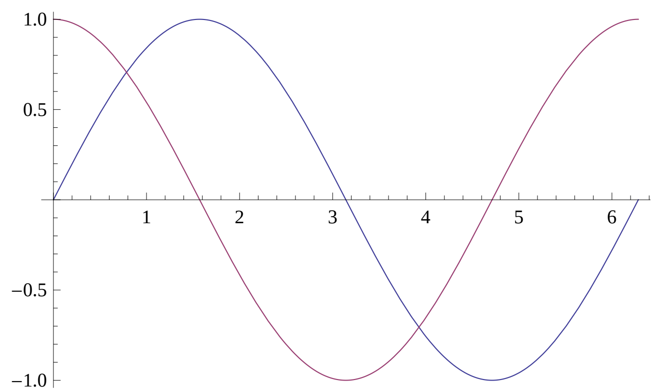
In[39]:=
```
Plot[Log[x], {x, 0, ⅇ}]
```

Out[39]=



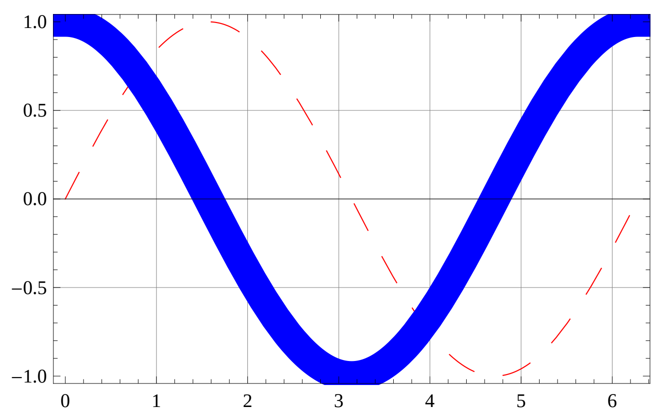
In[40]:=
```
Plot[{Sin[x], Cos[x]}, {x, 0, 2 π}]
```

Out[40]=



In[41]:=
```
Plot[{Sin[x], Cos[x]}, {x, 0, 2 π}, GridLines → Automatic, PlotStyle →
  {{RGBColor[1, 0, 0], Dashing[{0.05}]}, {RGBColor[0, 0, 1], Thickness[0.05]}}, Frame → True]
```

Out[41]=



Try plotting the polynomial $x^3 + x^2 - 6x + 9$ on an appropriate interval. Then make the curve red with dashed lines!

■ b. Plotting sets of data with ListPlot

ListPlot[{y1, y2, ... }] plots a list of data-values {y1, y2, ... }.The x coordinates for each point are taken to be 1, 2, ...
ListPlot[{{x1, y1}, {x2, y2}, ... }] plots a list of points with specified x and y coordinates.
ListPlot[{list1, list2, ... }] plots several lists of points.

Here are some examples.

*Note: The **Clear[x]** command is included to clear the value of x from memory if it has been defined!*

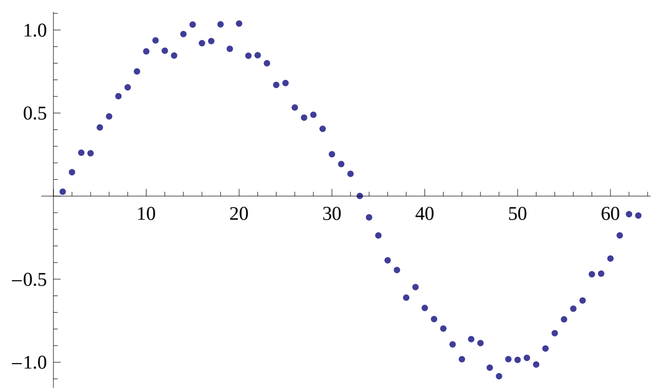
In[42]:=
```
Clear[x]
```

In[43]:=
```
data = Table[Sin[x] + RandomReal[{-0.1, 0.1}], {x, 0, 2 π, 0.1}]
```

Out[43]=
```
{0.0265898, 0.143839, 0.261307, 0.257967, 0.413373, 0.479765, 0.601512, 0.65488, 0.750546,
 0.871187, 0.93735, 0.875165, 0.846139, 0.975436, 1.03252, 0.920563, 0.933125,
 1.03389, 0.886532, 1.03869, 0.845002, 0.848016, 0.799721, 0.669365, 0.680602,
 0.533429, 0.472453, 0.489552, 0.40505, 0.252068, 0.192917, 0.134185, 0.00104081,
 -0.127852, -0.236851, -0.386648, -0.444896, -0.611055, -0.547682, -0.673067,
 -0.740344, -0.797225, -0.892619, -0.981952, -0.861047, -0.884696, -1.03261, -1.0841,
 -0.981481, -0.985798, -0.973695, -1.01367, -0.91738, -0.825183, -0.74178, -0.677647,
 -0.628433, -0.470411, -0.466664, -0.375679, -0.236585, -0.108811, -0.116941}
```

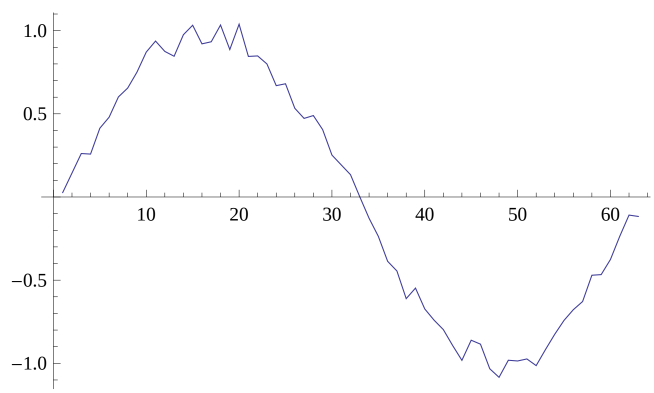
In[44]:=
```
ListPlot[data]
```

Out[44]=

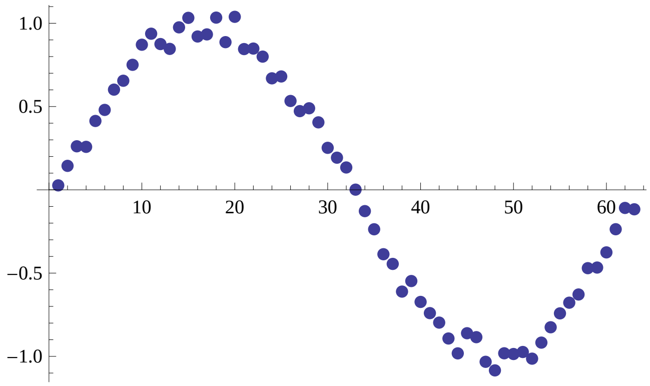


In[45]:=
```
ListPlot[data, Joined → True]
```

Out[45]=



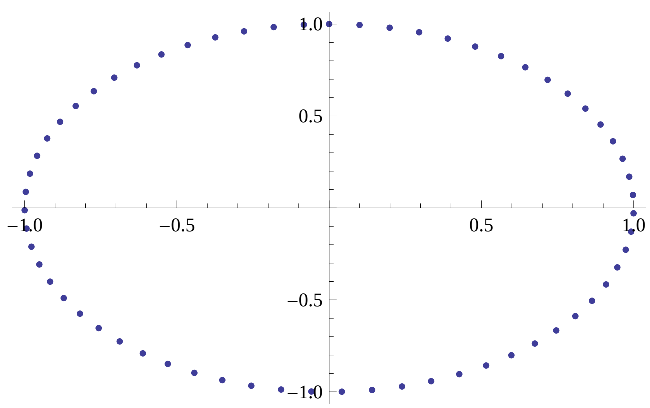
In[46]:= **ListPlot[data, PlotStyle → PointSize[0.02]]**

Out[46]= 

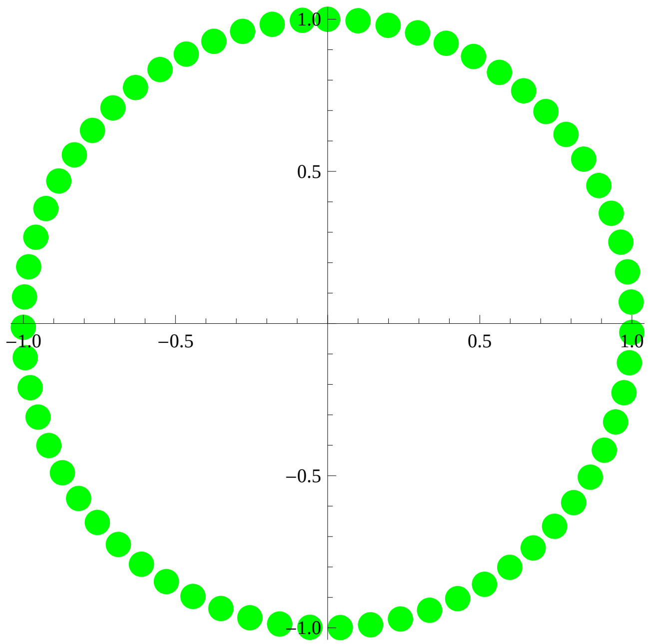
In[47]:= **xydata = Table[{Sin[x], Cos[x]}, {x, 0, 2 π, 0.1}]**

Out[47]= {{0., 1.}, {0.0998334, 0.995004}, {0.198669, 0.980067}, {0.29552, 0.955336}, {0.389418, 0.921061}, {0.479426, 0.877583}, {0.564642, 0.825336}, {0.644218, 0.764842}, {0.717356, 0.696707}, {0.783327, 0.62161}, {0.841471, 0.540302}, {0.891207, 0.453596}, {0.932039, 0.362358}, {0.963558, 0.267499}, {0.98545, 0.169967}, {0.997495, 0.0707372}, {0.999574, −0.0291995}, {0.991665, −0.128844}, {0.973848, −0.227202}, {0.9463, −0.32329}, {0.909297, −0.416147}, {0.863209, −0.504846}, {0.808496, −0.588501}, {0.745705, −0.666276}, {0.675463, −0.737394}, {0.598472, −0.801144}, {0.515501, −0.856889}, {0.42738, −0.904072}, {0.334988, −0.942222}, {0.239249, −0.970958}, {0.14112, −0.989992}, {0.0415807, −0.999135}, {−0.0583741, −0.998295}, {−0.157746, −0.98748}, {−0.255541, −0.966798}, {−0.350783, −0.936457}, {−0.44252, −0.896758}, {−0.529836, −0.8481}, {−0.611858, −0.790968}, {−0.687766, −0.725932}, {−0.756802, −0.653644}, {−0.818277, −0.574824}, {−0.871576, −0.490261}, {−0.916166, −0.400799}, {−0.951602, −0.307333}, {−0.97753, −0.210796}, {−0.993691, −0.112153}, {−0.999923, −0.0123887}, {−0.996165, 0.087499}, {−0.982453, 0.186512}, {−0.958924, 0.283662}, {−0.925815, 0.377978}, {−0.883455, 0.468517}, {−0.832267, 0.554374}, {−0.772764, 0.634693}, {−0.70554, 0.70867}, {−0.631267, 0.775566}, {−0.550686, 0.834713}, {−0.464602, 0.88552}, {−0.373877, 0.927478}, {−0.279415, 0.96017}, {−0.182163, 0.983268}, {−0.0830894, 0.996542}}

In[48]:= **ListPlot[xydata]**

Out[48]= 

In[49]:=
```
ListPlot[xydata, AspectRatio → Automatic, PlotStyle → {RGBColor[0, 1, 0], PointSize[0.04]}]
```

Out[49]=


■ 5. Defining Functions

Mathematica can be used to define functions. **Remember to use square brackets, an underscore after the variable in the square brackets, and a semicolon before the equals sign!**

■ a. Defining a function

Here is how to define a function, check that the function is saved in memory, evaluate the function at various inputs, and clear the function from memory. Try each command.

*Note: The **Clear[x]** command is included to clear the value of x from memory if it has been defined!*

In[50]:=
```
Clear[x]
```

In[51]:=
```
f[x_] := x^2 + 3 x + 5
```

In[52]:=
```
? f
```

```
Global`f
```

$f[x\_] := x^2 + 3x + 5$

In[53]:=
```
f[2]
```

Out[53]= 15

In[54]:=
```
f[☺]
```

Out[54]= $5 + 3 ☺ + ☺^2$

In[55]:=
```
f[x]
```
Out[55]= $5 + 3x + x^2$

In[56]:=
```
f[Sin[x] + Cos[x]]
```
Out[56]= 5 + 3 (Cos[x] + Sin[x]) + (Cos[x] + Sin[x])^2

In[57]:=
```
Simplify[%]
```
Out[57]= 6 + 3 Cos[x] + 3 Sin[x] + Sin[2 x]

Just like variables, functions can be cleared from memory!

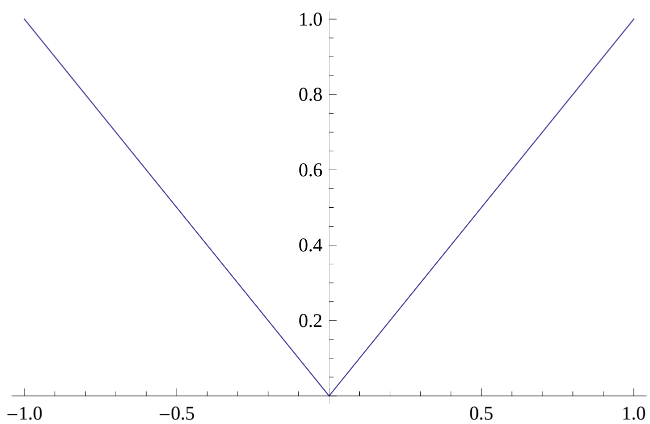
In[58]:=
```
Clear[f]
```
In[59]:=
```
?f
```
```
Global`f
```

Try defining the function g(t) = sin(t) and evaluating $g\left(\frac{\pi}{2}\right)$.

■ b. Defining a piecewise-defined function

To define a piecewise function, such as the absolute value of x, use the **condition symbol /;** along with following syntax. Try each command. **Clear** clears the variable a from memory so that it can be used later.

In[60]:=
```
a[x_] := x /; x ≥ 0
```
In[61]:=
```
a[x_] := -x /; x < 0
```
In[62]:=
```
Plot[a[x], {x, -1, 1}]
```
Out[62]=



In[63]:=
```
Clear[a]
```

Note: Portions of this lecture are based on the Introduction in "Calculus and *Mathematica*" by J.W. Emert and R.B. Nelson or come from the book "*Mathematica*" by S. Wolfram. The rest was created by M. A. Karls in Spring 2003, revised in Spring 2004, Fall 2007, and Fall 2008.