The items marked with ▶ should be mastered early in the semester (first two weeks).

▶Arithmetic

Syntax	Read As	Example
$-, +, *, /$	subtraction, addition, multiplication, division	`2*x-4/x`
space	multiplication	`k x` is the same as `k*x`
$\wedge$	power	`2^3`

Common Error: forgetting the space in multiplication: `kx` does not equal `k` times `x`.

▶Brackets

Syntax	Read As	Use	Example
`[ ]`	square brackets	enclosing arguments of functions	`Sin[2.5]`
`( )`	round brackets	algebraic groupings	`(x-x^3)/24`
`{ }`	curly brackets	lists, ordered pairs	`ListPlot[{{1,2},{2,3},{3,4}}]`

Common Error: missing brackets in algebra: `x/2-x` is not the same as `x/(2-x)`.

▶Built-in Functions

Built-in functions are functions already defined in *Mathematica*.

Function	Syntax	Function	Syntax	Function	Syntax
$\sin x$	`Sin[x]`	$\cos x$	`Cos[x]`	$\tan x$	`Tan[x]`
$\arcsin x$	`ArcSin[x]`	$\arccos x$	`ArcCos[x]`	$\arctan x$	`ArcTan[x]`
$\log_a x$	`Log[a,x]`	$\ln x$	`Log[x]`	e^x	`Exp[x]`
$\sqrt{x}$	`Sqrt[x]`	$n!$	`n!`	$\Gamma(x)$	`Gamma[x]`

Common Error: *Mathematica* is picky about capitalization. In particular, all built-in functions begin with a capital, so `cos[x]` is not the same as `Cos[x]`.

▶Built-in Constants

Built-in constants are constants already defined in *Mathematica*.

Constant	π	e	i	∞
Syntax	`Pi`	`E`	`I`	`Infinity`

Common Error: using `e` instead of `E`.
Other constants (speed of light, Avogadro's constant, etc) are available if you load the package `PhysicalConstants` using the command: ``<<PhysicalConstants` ``

▶Equal Signs

Syntax	Read As	Use	Example
`=`	set	defining variables and functions	`a=3.2`
`:=`	set delayed	defining variables and functions	`a:=Pi`
`==`	equal	equations	`equation1 = x^2-y^2==4`

Common Error: not using double equal sign == for equations.

Symbolic and Numeric Output

Mathematica works all its computations symbolically unless you tell it not to. You can tell it not to by using a decimal in a number you use, for example `Pi/3.0`, or you can use the command `N` as in `N[Pi/3]`. To get more decimals, use `N[Pi/3,320]` or `SetPrecision[Pi/3,320]`

▶Defining Your Own Functions

You tell *Mathematica* which variables are the independent variables by using an underscore:

```
f[x_,t_] = Sin[t]*(Cos[k*x]-4)
g[x_] = Piecewise[{{x^2, x < -1}, {x + 2, x >= -1}}]
```

Common Error: forgetting the underscore.

Working With Functions

Mathematical Operation		Syntax
value of function	$f(3)$	`f[3]`
decimal value of function	$f(3)$	`f[3.0]` or `N[f[3]]`
derivative	$\frac{d}{dx}[f(x)]$	`D[f[x],x]` or `f´[x]`
indefinite integral	$\int f(x)\,dx$	`Integrate[f[x],x]`
definite integral	$\int_a^b f(x)\,dx$	`Integrate[f[x],{x,a,b}]`
numerical integration	$\int_0^2 f(x)\,dx$	`NIntegrate[f[x],{x,0,2}]`
composition	$(f \circ g)(x) = f(g(x))$	`f[g[x]]`

Common Error: for indefinite integrals, *Mathematica* does not include a constant of integration in its answer.

Solving Equations

Syntax	Use	Example
▶`Solve`	symbolic solution of equations	`Solve[{x==y-2,x^2+y^4==4},{x,y}]`
▶`NSolve`	decimal solution of equations	`NSolve[{x==y-2,x^2+y^4==4},{x,y}]`
`Eliminate`	eliminate a variable from a set of equations	`Eliminate[{x==t^2+1,y==5/t},t]`
`Reduce`	symbolic solution of equations, returns conditions	`Reduce[{x+Cos[x*y]==0},{x, y}]`

`Reduce` is very useful for trig equations. `Eliminate` is used to determine an implicit function from a parametric representation.

Plotting

I have included some useful options (`PlotStyle`, `Joined`, `AspectRatio`, `PlotRange`) in the examples below. These options can be left out to create a simple plot.

Plot Type	Example
▶plot $f(x)$	`Plot[f[x], {x, -1, 5}]`
▶plot $f(x)$ and $g(x)$	`Plot[{f[x], g[x]}, {x, -1, 5}, PlotStyle -> Thick]`
plot list of data points	`ListPlot[{{1,2},{2,3},{3,6}}, Joined -> True]`
implicit plot of $f(x,y)=0$ in $\mathbb{R}^2$	`ContourPlot[f[x,y]==0, {x,-5,5}, {y,-5,5}, AspectRatio->1]`
plot of parametric function $x=f(t), y=g(t)$ in $\mathbb{R}^2$	`ParametricPlot[{f[t],g[t]},{t,0,8},PlotRange->{{-1,1},{-2,2}}]`
contour plot of $z=f(x,y)$	`ContourPlot[f[x,y],{x,-2,2},{y,-2,4}, Contours->100]`
plot of $z=f(x,y)$ in $\mathbb{R}^3$	`Plot3D[f[x,y],{x,-2,2},{y,-2,4}]`
plot of space curve $x=f(t), y=g(t), z=h(t)$ in $\mathbb{R}^3$	`ParametricPlot3D[{f[t],g[t],h[t]},{t,-2,8}]`
plot of surface $x=f(s,t), y=g(s,t), z=h(s,t)$ in $\mathbb{R}^3$	`ParametricPlot3D[{f[s,t],g[s,t],h[s,t]},{t,-2,8},{s,-3,9}]`
plot $f(x)$ with area between the curve and x-axis shaded	`Plot[f[x], {x, -1, 3}, Filling -> Axis]`
plot $f(x)$ and $g(x)$ with area between the curves shaded	`Plot[{f[x], g[x]}, {x, -1, 3}, Filling -> {1}]`
animation of $\cos(ax+b)+c$ as a,b,c vary	`Manipulate[ Plot[Cos[a*x+b]+c,{x,0,2*Pi},PlotRange->{{0,2*Pi},{-5,5}}], {a, -2, 2}, {b, 0, 2*Pi}, {c, -2, 2}]`