

Mathematica Tipsheet

- The most common sources of error are capitalization, using the wrong type of brackets, missing or misplaced commas, and misspelled words.
- Mathematica remembers everything you tell it. But it doesn't listen to everything you type. It will pay attention to commands you tell it to evaluate, when you tell it.
- You can enter commands by hitting `shift + enter`, or by hitting the `enter` key next to the numerical keypad. Just hitting `enter` will just start a new line.
- Mathematica is case-sensitive; `x` and `X` do not mean the same thing. Built-in Mathematica commands are always written in CamelCase, which means that the first letter of each distinct word is capitalized. For instance, write `E` and `Pi` for the mathematical constants. `Sin` and `Cos` will work; `sin` and `cos` will not. At various points we write `PlotRange` and `GridLines`.
- Mathematica uses square brackets to denote inputs to functions. `Cos[x]` means $\cos(x)$; `Cos(x)` will return an error. It uses curly braces to construct lists; we will mainly use this to feed in lists of data or parameters for graphs. Mathematica uses parentheses to control order of operations, but not for else.
- You can use any variable names you want. `thisisavariablename` works perfectly fine as a variable, as does `x213`.
- If you write two variables or function names with no space between them, Mathematica will assume you actually have one long variable. `x*y` and `x y` both mean $x \cdot y$; `xy` is one variable whose name is "xy."
- You can set the value of a variable with `=`. If you write `x=3`, every `x` is the same as a `3`.
- Anything in dark blue has no value assigned. Anything in light blue is bound locally. If Mathematica knows a value for something, it will be black.
- If you want to test for whether two things are equal, use a double equals sign. `3==3` returns `True` and `3==5` returns `False`.
- Define new functions like `f[x_]:=x^2+1`. Don't forget the underscore and colon.
- Mathematica will return exact outputs given exact inputs. If you want decimals, either use decimals in your inputs, or plug your answer into the function `N`, e.g. `N[Pi+1]`.
- If `f` is a function, then you can plot it with a command like `Plot[f[x],{x,0,5}]`. This tells us to plot the function `f` with domain going from 0 to 5. Similarly, `Plot[Sin[x] + x^2,{x,-Pi, Pi}]` will work fine.
- There are many options you can add to `Plot`, by placing a comma after the domain. One helpful one is `PlotRange`. `Plot[f[x],{x,0,3},PlotRange->{0,5}]` will force the function to display the y -axis from 0 to 5, overriding the automatic scaling.