Tanya Gupta

MAT/19/73

PRESENTATION

SEC-2 : Computer Algebra Systems & Related Softwares

TOPIC : FUNCTIONS AND THEIR GRAPHS

Functions in Mathematica

Built-in Functions

- Mathematica contains many built-in functions. In Mathematica, every built-in function name begins with capital letter. Arguments of functions follow in square brackets.
- Example:-

```
In[1]:= Sin[Pi / 2]
Out[1]= 1

In[2]:= Log[1]
Out[2]= 0
```

Defining your own functions

- You can define your own function in Mathematica. The name of the function should be lowercase letter or a word that begins with lowercase letter.
- Example:-

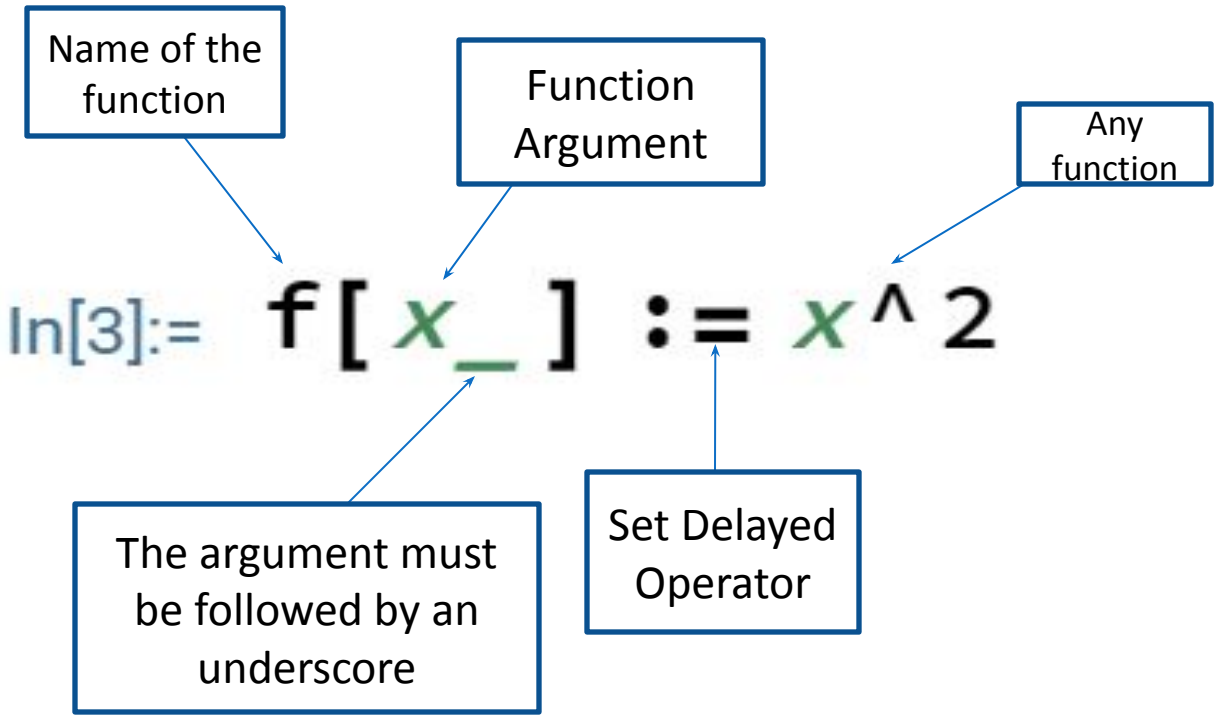
```
In[3]:= f[x_] := x^2
```

This function will take an input x, and output x^2

```
In[5]:= f[15]
Out[5]= 225

In[6]:= f[100]
Out[6]= 10 000
```

How to define a Function?

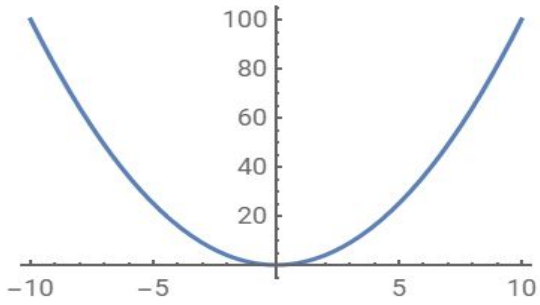


Plotting a Function

Mathematica offers many graphing options. You can create graph of functions using Mathematica's Plot command.

```
In[11]:= f[x_] := x^2
In[12]:= Plot[f[x], {x, -10, 10}]
```

Out[12]=



The Plot command takes 2 arguments, separated by comma

1. **The function to be graphed.**
2. ***Iterator*** **:- Specifies the domain over which the plot will be constructed i.e. in this case {x,-10,10}**

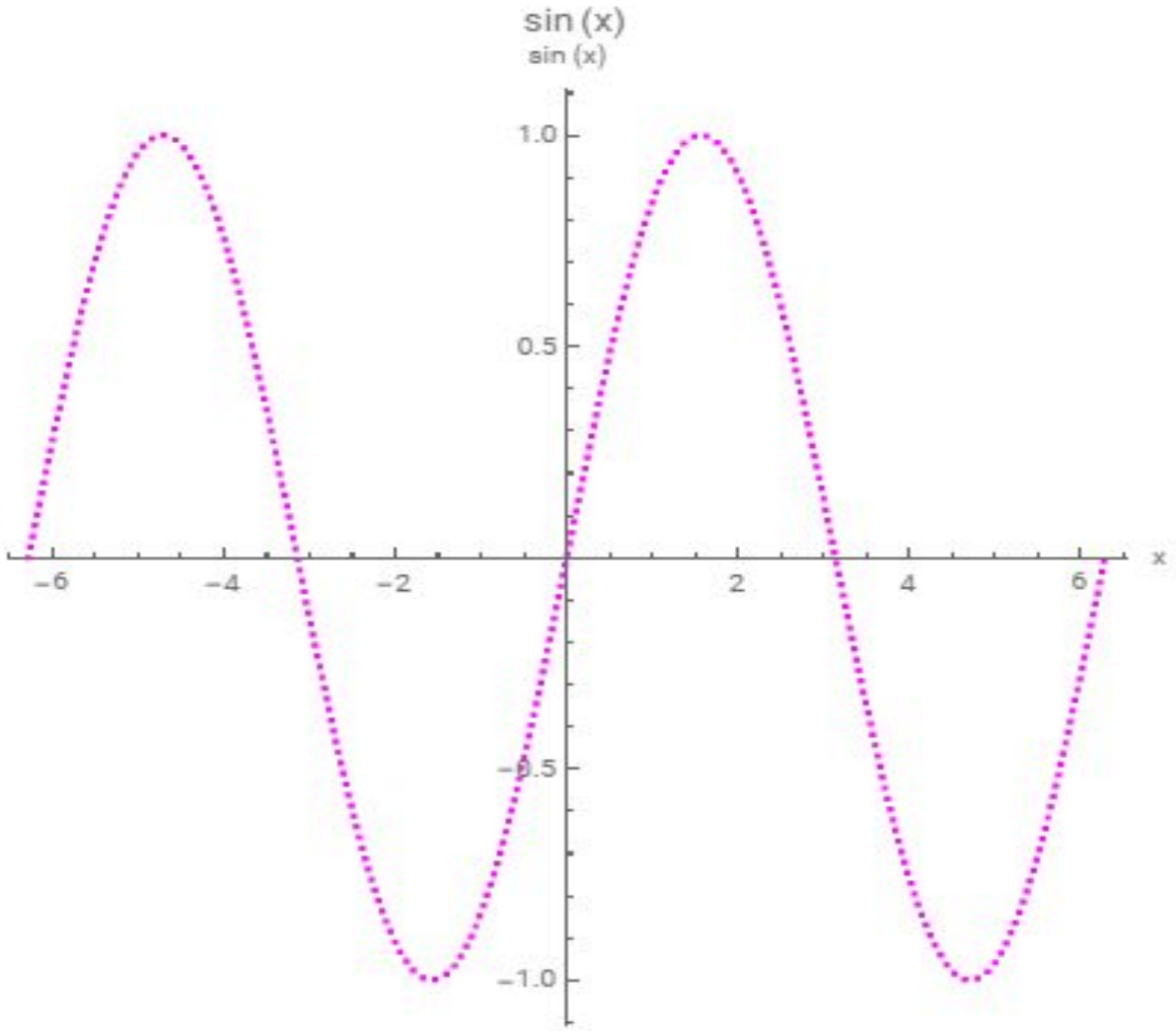
Mathematica's Plot Options

Mathematica gives you the power to customize every aspect of the styling of your plots. There are over 50 options for plot command

• **AspectRatio**	specifies the ratio of height to width for a plot
• **Axes**	specifies whether axes should be drawn.
• **Frame**	Replace axes with a frame around entire graph
• **PlotLabel**	specifies an overall label for a plot.
• **PlotLabels**	specifies what labels to use for each curve for visualization of functions
• **PlotLegends**	specifies what legends to use for each curve
• **PlotRange**	specifies what range of coordinates to include in a plot
• **PlotStyle**	graphics directives to specify the style for each curve. The PlotStyle option may be set to any single graphic directive . Multiple graphic directive should be wrapped in **Directive** command
• **PlotPoints**	specifies how many initial sample points to use.
• **AxesLabel**	Labels can be added to the axes via this option

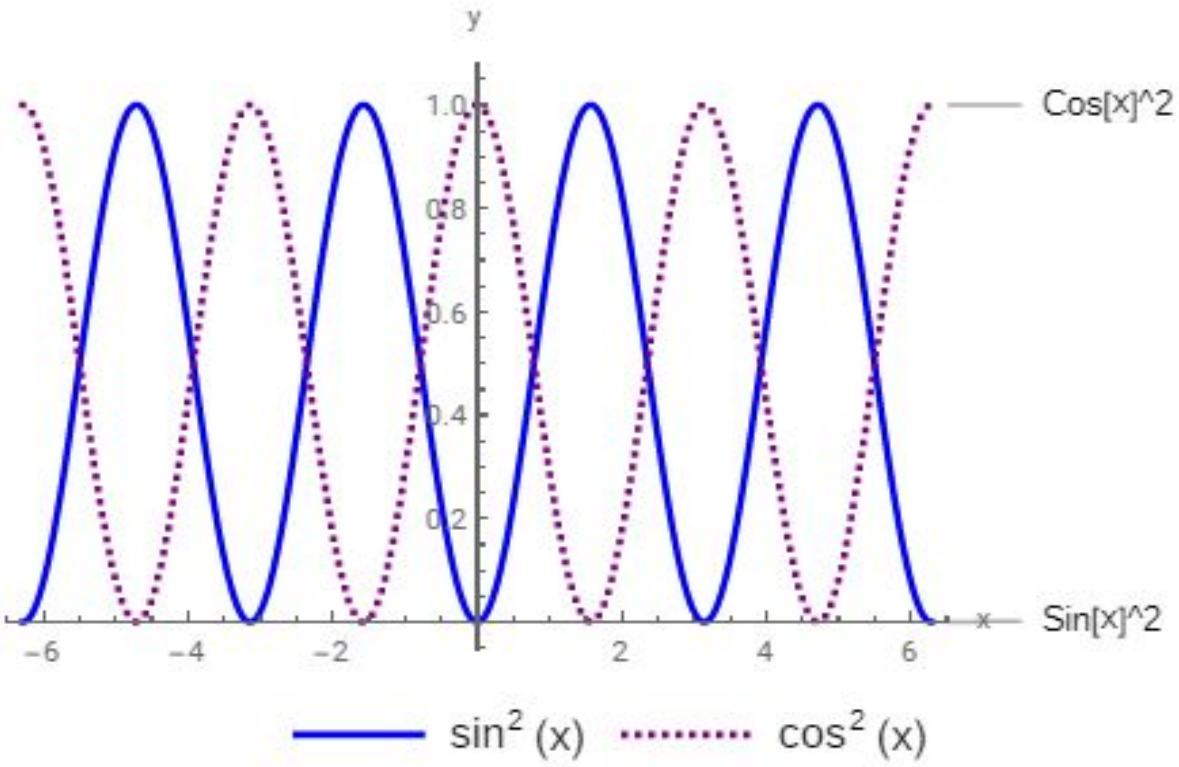
In[7]:=
```
Plot[Sin[x], {x, -2 Pi, 2 Pi}, AspectRatio → 1, Axes → True,
 AxesLabel → {x, Sin[x]}, PlotLabel → Sin[x],
 PlotStyle → Directive[Magenta, Dotted, Thick]]
```

Out[7]=



In[31]:=
```
Plot[{Sin[x]^2, Cos[x]^2}, {x, -2 Pi, 2 Pi},
  AspectRatio → 2 / 3, Axes → True,
  PlotStyle → {{Blue, Thick}, {Purple, Dotted, Thick}},
  PlotLabels → {"Sin[x]^2", "Cos[x]^2"},
  PlotLegends → "Expressions", AxesLabel → {x, y}]
```

Out[31]=

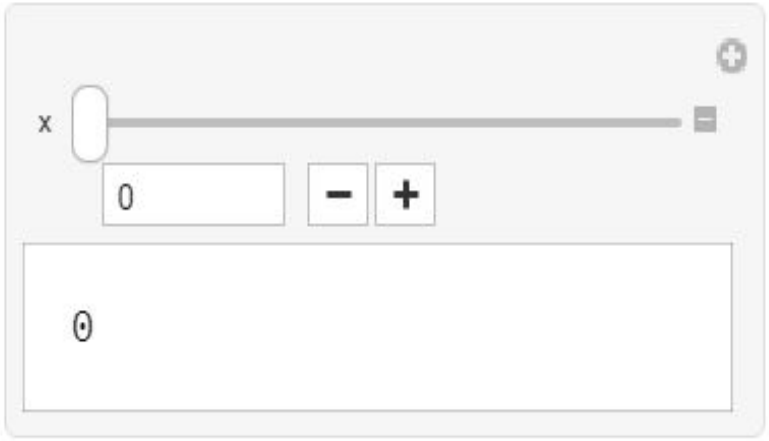


Manipulate Command

The Manipulate command is used to manipulate an expression in real time using mouse. The output you get from evaluating a Manipulate command is an interactive object containing one or more controls (sliders, etc.) that you can use to vary the value of one or more parameters.

In[25]:= `Manipulate[x^3, {x, 0, 100, 3}]`

Out[25]=



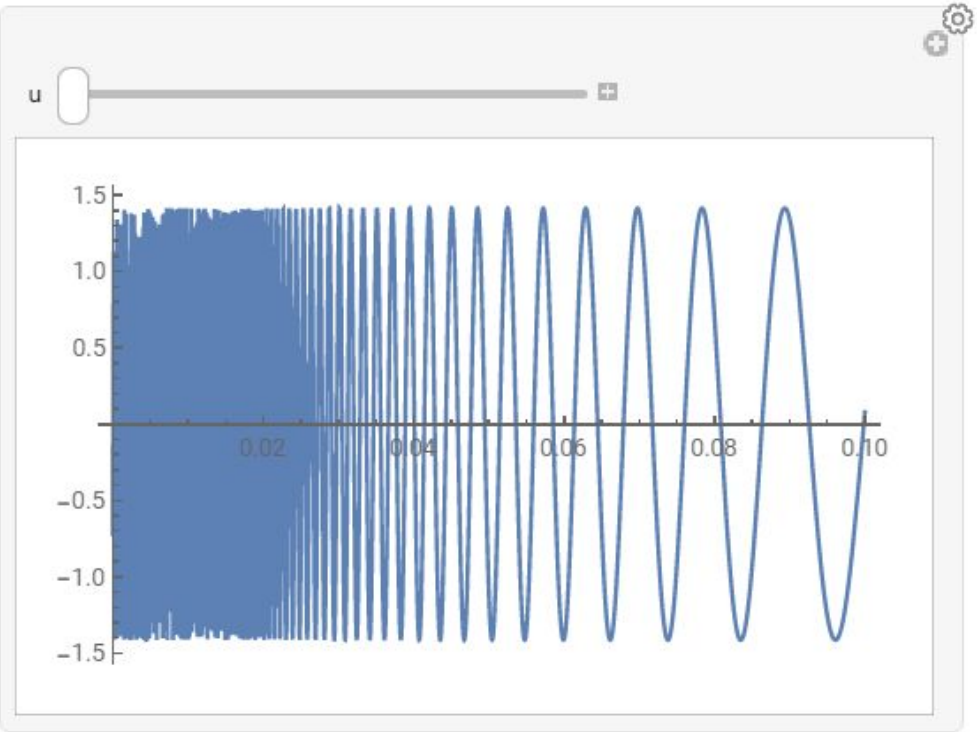
Note: ***The Manipulate command in its most basic form takes 2 arguments , separated by comma are:***

1.Describes the expression to be manipulated

2.An Iterator

In[27]:=
```
Manipulate[Plot[ Sin[4 / x] + Cos[4 / x],
  {x, 0, u}], {u, 0.1, 2}]
```

Out[27]=



In this example, we make a plot of the function Sin[4/x] + Cos[4/x] , the right endpoint of which is controlled with a slider, while the left end point is fixed at 0