# Real Valued Functions Of Two or More Variables:-

- **A real-valued function is a function whose values are real numbers.**
- **It is a function that assigns a real number to each member of its domain.**
- **A real valued function of two variables denoted z=f(x,y) is a rule that assigns to each point (x,y)∈D(f) a unique real number f(x,y)∈R(f).**

In[1]:=
```
f[x_, y_] := Sin[x^2 - y^2]
```

## To make it more convenient,we can write it as:-

In[2]:=
```
Clear[f, x, y];
f = Sin[x^2 - y^2];
```

## A function of three variable can be written as:-

In[4]:=
```
Clear[g, x, y, z];
g = x^2 * y^3 - 3 xz;
```

## To evaluate the value of a function at a given point:-

- **At $(0,\sqrt{(\pi/4)})$:**

In[7]:=
```
f /. {x → 0, y → √(π / 4)}
```

Out[7]=
$$-\frac{1}{\sqrt{2}}$$

- **At $(1-\pi,1+\pi)$:**

In[8]:=
```
f /. {x → 1 - π, y → 1 + π}
```

Out[8]=
$$\text{Sin}\left[(1-\pi)^2-(1+\pi)^2\right]$$

In[9]:=
```
Simplify[%]
```

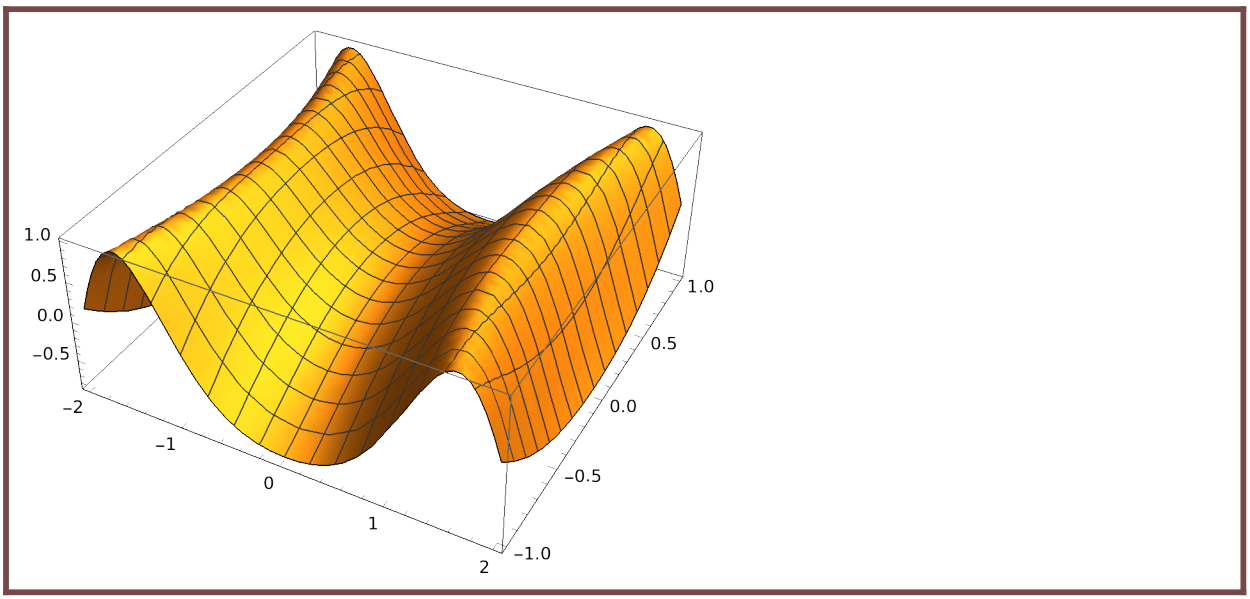
Out[9]=
```
0
```

# Plotting functions of two variables with Plot3D:-

- **The plotting of functions of two variables can be performed with the command Plot3D.**
- **Here we need an iterator specifying the span of values assumed by each of two variables. The plot will be shown over the rectangular domain in the plane determined by the two iterators.**

In[11]:=
```
Plot3D[f, {x, -2, 2}, {y, -1, 1}]
```

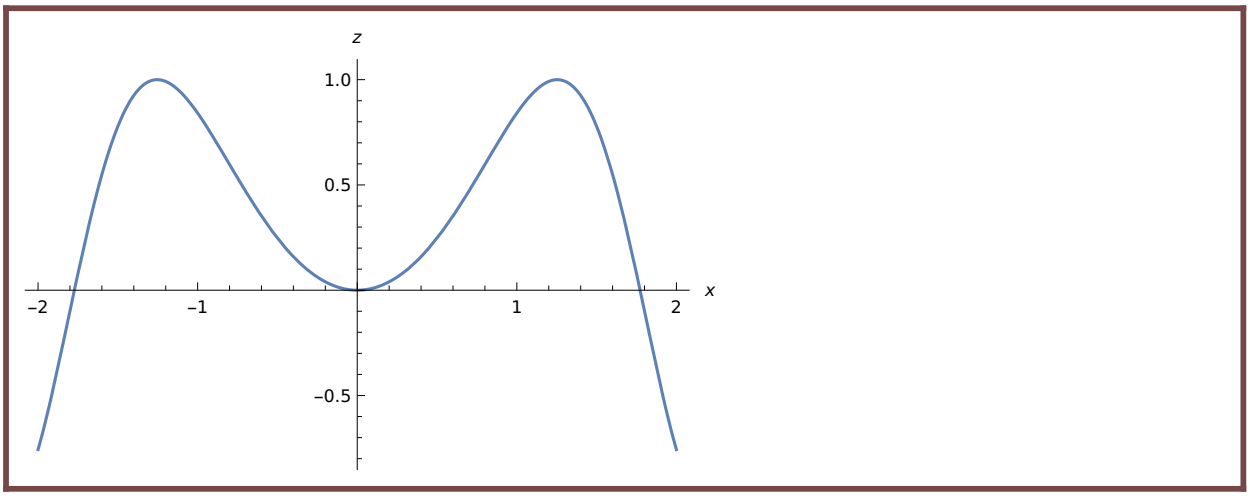
Out[11]=



- **To find traces or to sketch any vertical cross-section,we will use the following commands:-**

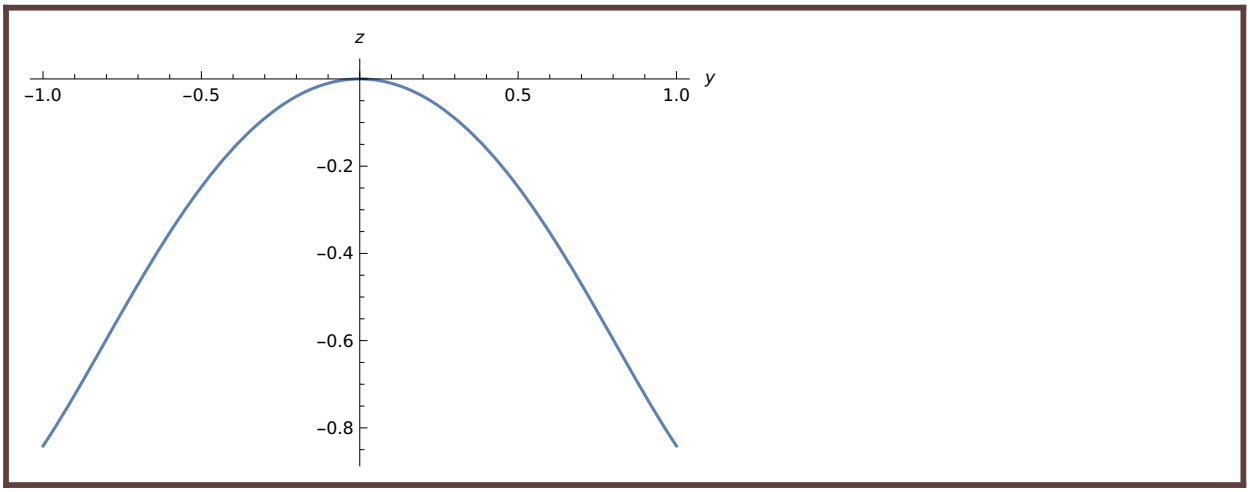
In[12]:=
```
Plot[f /. y → 0, {x, -2, 2}, AxesLabel → {x, z}]
```

Out[12]=




In[13]:=
```
Plot[f /. x → 0, {y, -1, 1}, AxesLabel → {y, z}]
```

Out[13]=



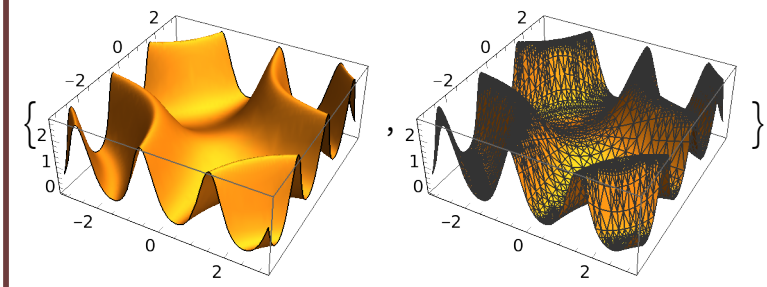
# To tweak the output of these plotting commands in some incredible ways.

**Some common settings are : AxesLabel, PlotLabel, PlotPoints, MaxRecursion PlotRange,Mesh and MeshFunctions**

- **Mesh command will display the polygons produced by Plot3D to render the image.**
- **PlotPoints settings control how many equally spaced points are initially sampled in each direction**
- **MaxRecursion controls the number of recursive subdivisions permitted to fine-tune the image.**

In[18]:=
```
Table[Plot3D[e^Sin[x*y], {x, -π, π},
  {y, -π, π}, Mesh → m, MaxRecursion → 4], {m, {None, All}}]
```

Out[18]=

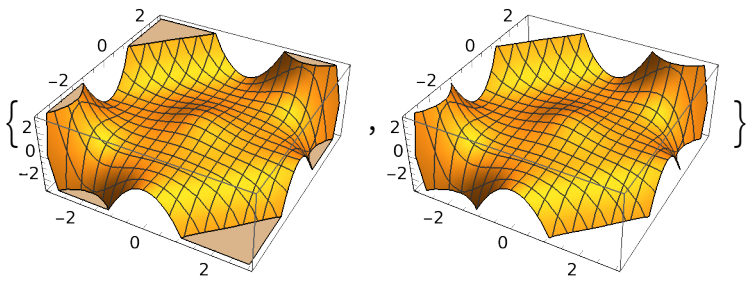


# Adjusting the PlotRange and BoxRatios:-

- **The option setting ClippingStyle -> None will remove the default horizontal planes placed into the clipped areas.**
- **PlotRange is an option for graphics functions that specifies what range of coordinates to include in a plot.**
- **PlotRange -> All will show the entire graph.**
- **BoxRatios determines the relative dimensions of the bounding box.**
- **BoxRatios-> Automatic will scale the bounding box so that all axes have the same scale.**

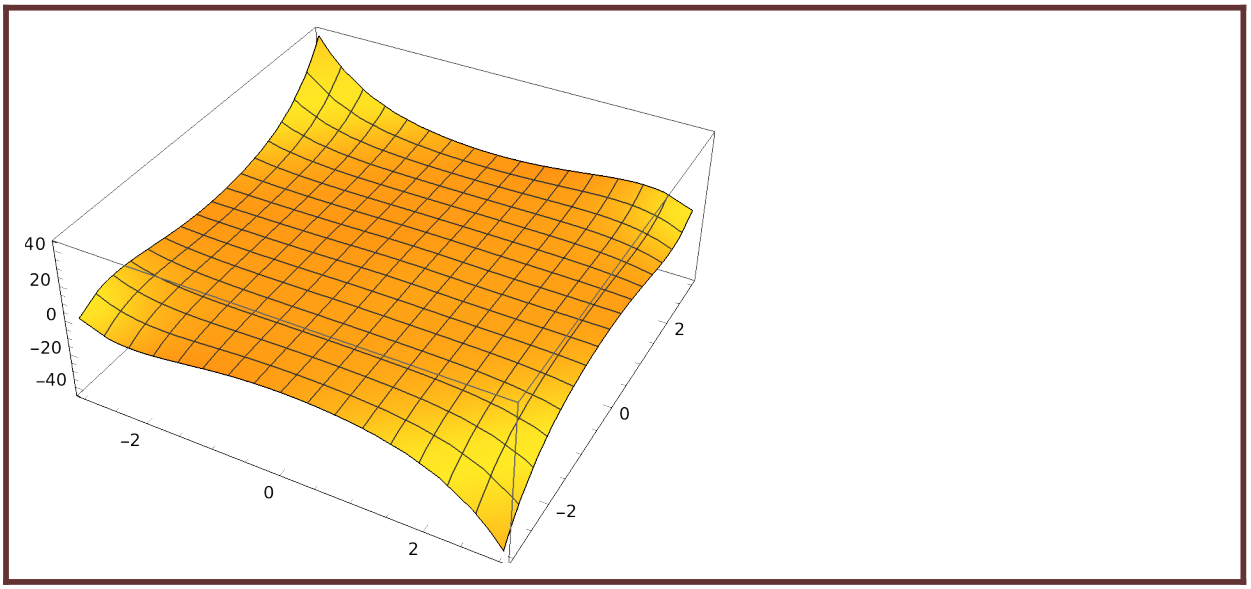
In[19]:=
```
Table[Plot3D[{x^2 y^5 - x^5 y^2}/100 + e^{-(x^2 + y^2)},
  {x, -3, 3}, {y, -3, 3}, ClippingStyle → k], {k, {Automatic, None}}]
```

Out[19]=



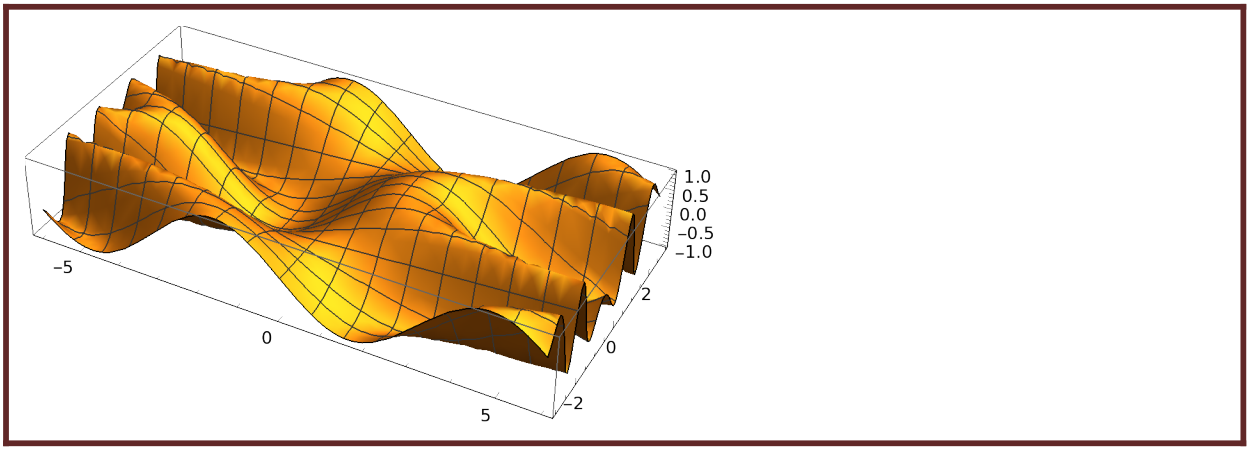
In[20]:=
```
Plot3D[{x^2 y^5 - x^5 y^2}/100 + e^{-(x^2 + y^2)}, {x, -3, 3}, {y, -3, 3}, PlotRange → All]
```

Out[20]=




In[22]:=
```
Plot3D[Sin[x * Cos[y]], {x, -6, 6}, {y, -3, 3}, BoxRatios → Automatic]
```

Out[22]=

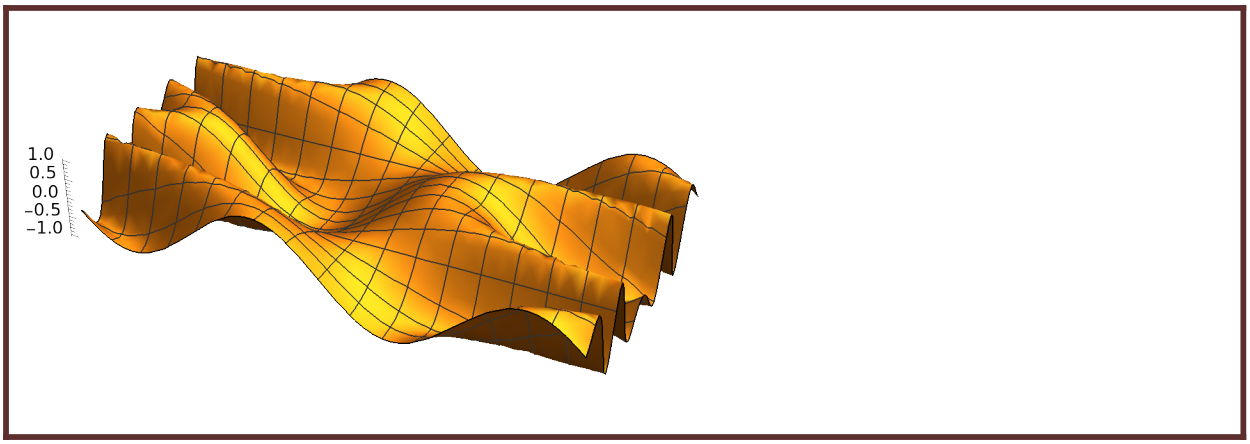


# The Bounding Box, Axes, and ViewPoint:-

- **The options Boxed and Axes can be used to modify the appearance of the bounding box and the tick marks that appear on three of its sides.**
- **By default, both options are set to True. To remove the bounding box entirely, set both to False.**
- **Axes can also be set to a list to display only selected axes.**
- **AxesEdge controls in each of the three coordinate directions which of the four parallel sides of the bounding box in that direction are to be used as an axis.**
- **ViewPoint specifies the position in space from which it is seen.**

In[24]:=
```
Plot3D[Sin[x * Cos[y]], {x, -6, 6}, {y, -3, 3}, BoxRatios → Automatic, Boxed → False,
 Axes → {False, False, True}, AxesEdge → {Automatic, Automatic, {-1, -1}}]
```

Out[24]=



In[25]:=
```
Plot3D[Sin[x * Cos[y]], {x, -6, 6}, {y, -3, 3},
 BoxRatios → Automatic, ViewPoint → {3, 0, 1}]
```

Out[25]=