A Brief Introduction to Mathematica

Deobrat Singh

Department Of Physics & Astrophysics

University Of Delhi

Delhi - 110007

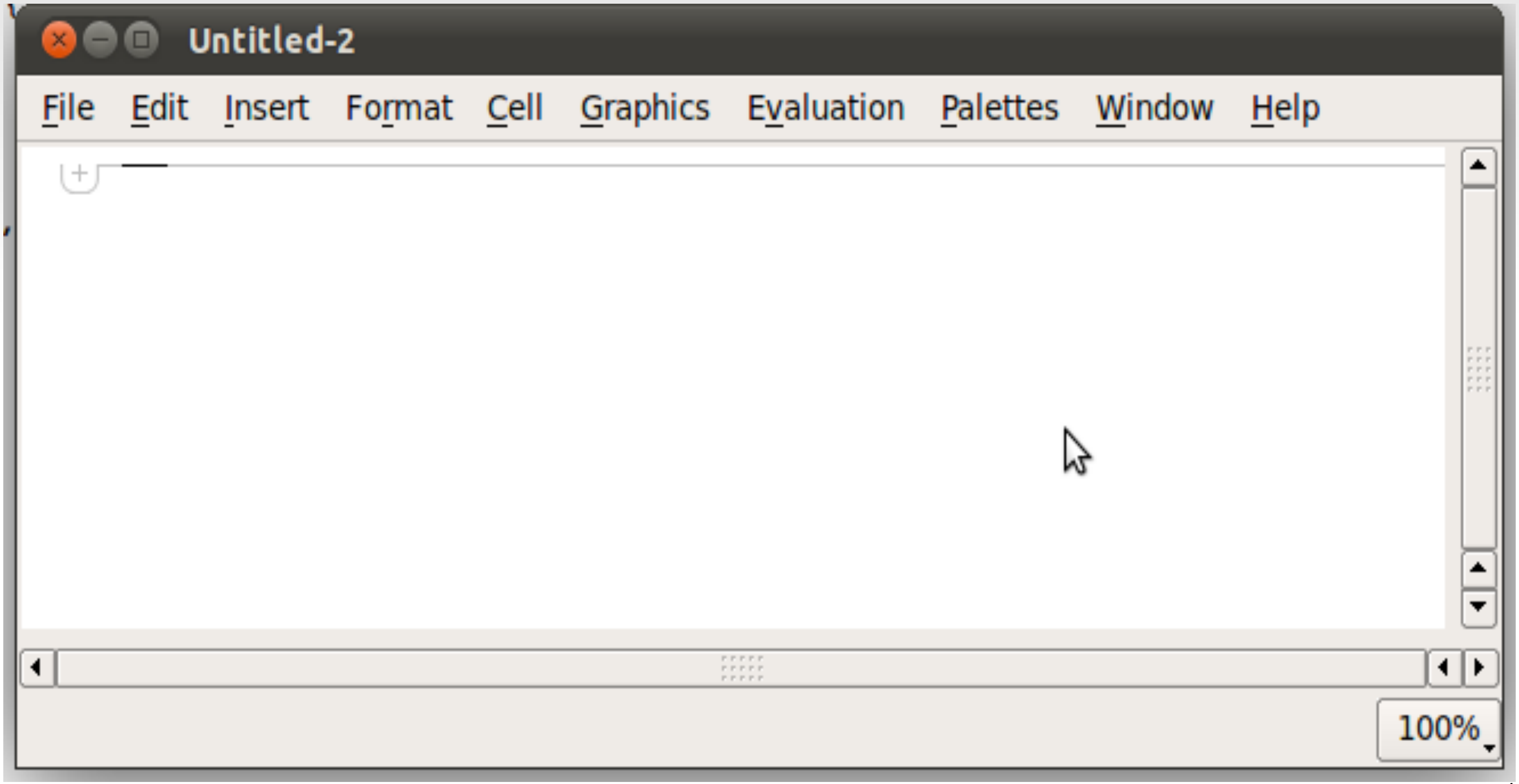
Outline:

- Running Mathematica
- Getting Started
- Working with the Notebook Interface
- Getting Information from Mathematica
- The Mathematica System

The Structure of Mathematica

- Mathematica →a modular software system
- The kernel which actually performs computations is separate from the front end which handles interaction with the user.
- To work on Mathematica is to use interactive documents known as notebooks.
- Notebooks mix Mathematica input and output with text, graphics, palettes and other material.
- Notebook interface interactive documents
- Text-based interface text from the keyboard
- MathLink interface communication with other programs
- It can interact not only with human users but also with other programs → achieved primarily through MathLink → a standardized protocol for two-way communication between external programs and the Mathematica kernel.

Screenshot of a blank notebook



Using a Notebook Interface

- A purely graphical interface → double-click the Mathematica icon to start Mathematica.
- A textually based operating system → type the command Mathematica to start.
- An empty notebook with a blinking cursor → by default will interpret your text as input →type Shift+Enter to make Mathematica process your input.
- Shift+Enter tells Mathematica that you have finished your input.

  If numeric keypad → its Enter key instead of Shift+Enter.
- After you send Mathematica input from your notebook, Mathematica will label your input with In[n]:=. It labels the corresponding output Out[n]=. Labels are added automatically.
- In[1]:= 2 + 2

  Out[1]= 4
- notebooks are part of the "front end" to Mathematica.

Begin.nb

```
2 + 2
```

```
4
```

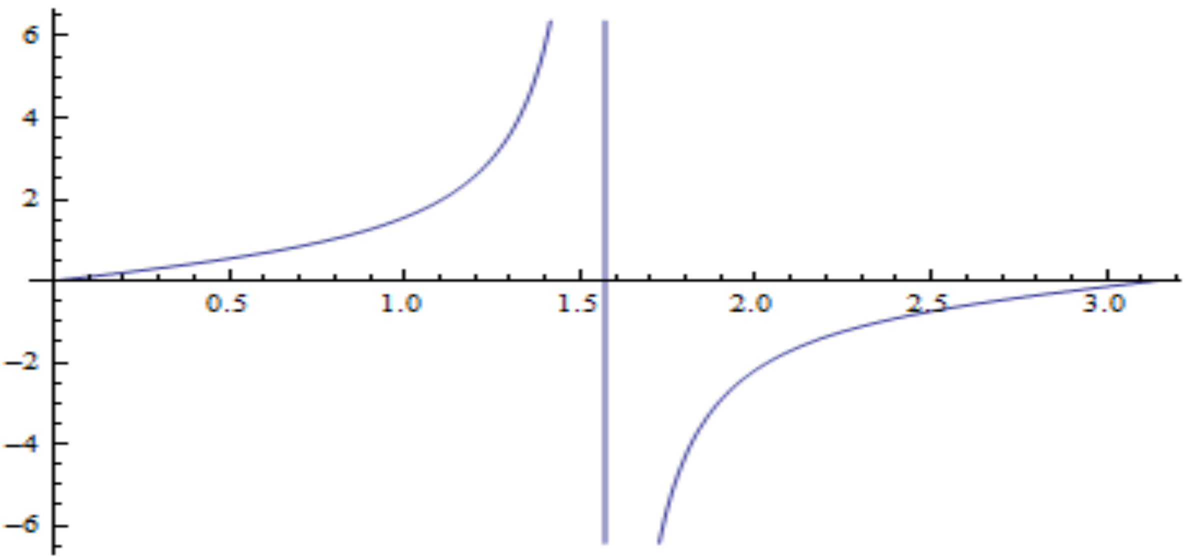
```
3 - 5
```

```
-2
```

```
Solve[x^2 - 2 x + 1 == 0, x]
```

```
{{x → 1}, {x → 1}}
```

```
Plot[Tan[x], {x, 0, Pi}]
```



100%

Getting Information about Mathematica Objects

- ?Log

  This gives information on the built-in function Log.

  Log[z] gives the natural logarithm of z (logarithm to base e).

  Log[b,z] gives the logarithm to base b.

- ?Name show information on Name

  ??Name show extra information on Name

  ?Aaaa* show information on all objects whose names begin with Aaaa

- You can put * anywhere in the string you ask ? about. For example, ?*Expand would give you all objects whose names end with Expand. Similarly, ?x*0 would give you objects whose names start with x, end with 0, and have any sequence of characters in between.

Untitled-1 *

In[2]:= ? Log

Log[*z*] gives the natural logarithm of *z* (logarithm to base *e*).

Log[*b*, *z*] gives the logarithm to base *b*. »

In[3]:= ?? Log

Log[*z*] gives the natural logarithm of *z* (logarithm to base *e*).

Log[*b*, *z*] gives the logarithm to base *b*. »

```
Attributes[Log] = {Listable, NumericFunction, Protected}
```

In[4]:= ?Export

Export["*file.ext*", *expr*] exports data to a file, converting it to the format corresponding to the file extension *ext*.

Export[*file*, *expr*, "*format*"] exports data in the specified format.

Export[*file*, *exprs*, *elems*] exports data by treating *exprs* as elements specified by *elems*. »

In[5]:= ?? Export

Export["*file.ext*", *expr*] exports data to a file, converting it to the format corresponding to the file extension *ext*.

Export[*file*, *expr*, "*format*"] exports data in the specified format.

Export[*file*, *exprs*, *elems*] exports data by treating *exprs* as elements specified by *elems*. »

```
Attributes[Export] = {Protected, ReadProtected}
```

100%

Export - Wolfram Mathematica

SEARCH Export

Search for all pages containing Export.

MORE INFORMATION

EXAMPLES

Basic Examples (3)

Export an image object:

In[1]:= Export["landscape.jpg", ]

Out[1]= landscape.jpg

Export a plot as a GIF:

```
In[1]:= Export["test.gif", Plot[Sin[x], {x, 0, 10}]]
Out[1]= test.gif
```

Export a formula as a GIF:

```
In[1]:= Export["test.gif", Integrate[1 / (x^4 - 1), x]]
Out[1]= test.gif
```

100%

Warnings and Messages

- If it looks as if Mathematica is doing something you definitely did not intend, Mathematica will usually print a message to warn you.

e.g.the square root function should have only one argument. Mathematica prints a message to warn you that you have given two arguments here.

- In[1]:= Sqrt[4, 5]

  During evaluation of In[1]:= Sqrt::argx: Sqrt called with 2 arguments; 1 argument is expected. >>

Out[1]= Sqrt[4, 5]

- You can suppress the message for one evaluation using Quiet.

  In[2]:= Quiet[Sqrt[4, 5]]

  Out[2]= Sqrt[4, 5]

Interrupting Calculations

- There will probably be times when you want to stop Mathematica in the middle of a calculation. Perhaps you realize that you asked Mathematica to do the wrong thing. Or perhaps the calculation is just taking a long time, and you want to find out what is going on.
- The way that you interrupt a Mathematica calculation depends on what kind of interface you are using.
- Alt+, notebook interfaces
- Ctrl+C text-based interfaces

The Mathematica System

- Getting Used to Mathematica
- Differences between Computer Systems
- The Limits of Mathematica

Getting Used to Mathematica

Some Important Rules:

- Arguments of functions are given in square brackets.
- Names of built-in functions have their first letters capitalized.
- Multiplication can be represented by a space.
- Powers are denoted by ^.
- Numbers in scientific notation are entered, for example, as 0.00025 or 2.5*10^-4.

Some examples: Sin[x]

Each of these represents multiplication.

a*b a b a(b+1); 2 x means 2*x

- Uppercase and lowercase letters are recognized as different characters. Lists are wrapped with curly brackets.
- Commas are used to separate arguments.

N[Pi, 50];

Differences between Computer Systems

- *Elements of Mathematica that are exactly the same on all computer systems:*

The language used by the Mathematica kernel

The structure of Mathematica notebooks

The MathLink communication protocol

- *Elements that can differ from one computer system to another:*

The visual appearance of windows, fonts, etc.

Mechanisms for importing and exporting material from notebooks

Keyboard shortcuts for menu commands

- Communications between the kernel and the front end are handled by MathLink.

The Limits of Mathematica

- In just one Mathematica command, you can easily specify a calculation that is far too complicated for any computer to do. For example, you could ask for *Expand[(1+x)^(10^100)]*. The result of this calculation would have *10^100+1* terms--more than the total number of particles in the universe.

- If your computer does run out of memory in the middle of a calculation, most versions of Mathematica have no choice but to stop immediately. As a result, it is important to plan your calculations so that they never need more memory than your computer has.

- Memory space is the most common limiting factor in Mathematica calculations. Time can also, however, be a limiting factor. You will usually be prepared to wait a second, or even a minute, for the result of a calculation. But you will less often be prepared to wait an hour or a day, and you will almost never be able to wait a year.

Core Language "How to" Topics

- Create Definitions for Variables and Functions »
- Clear My Definitions »
- Map a Function over a List »
- Work with Pure Functions »
- Enter Ranges and Options for Functions »

How to Create Definitions for Variables and Functions

- Here is a simple transformation rule. It says: whenever you see x, replace it by 3:

```
In[31]:= x = 3

Out[31]= 3
```

- The variable x has a value of 3.

```
In[32]:= x^2

Out[32]= 9
```

- Functions in Mathematica are defined by rules that act on patterns.

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

- The rule says: if you have f of any expression, replace it by that expression squared:

```
In[46]:= f[expr]
```

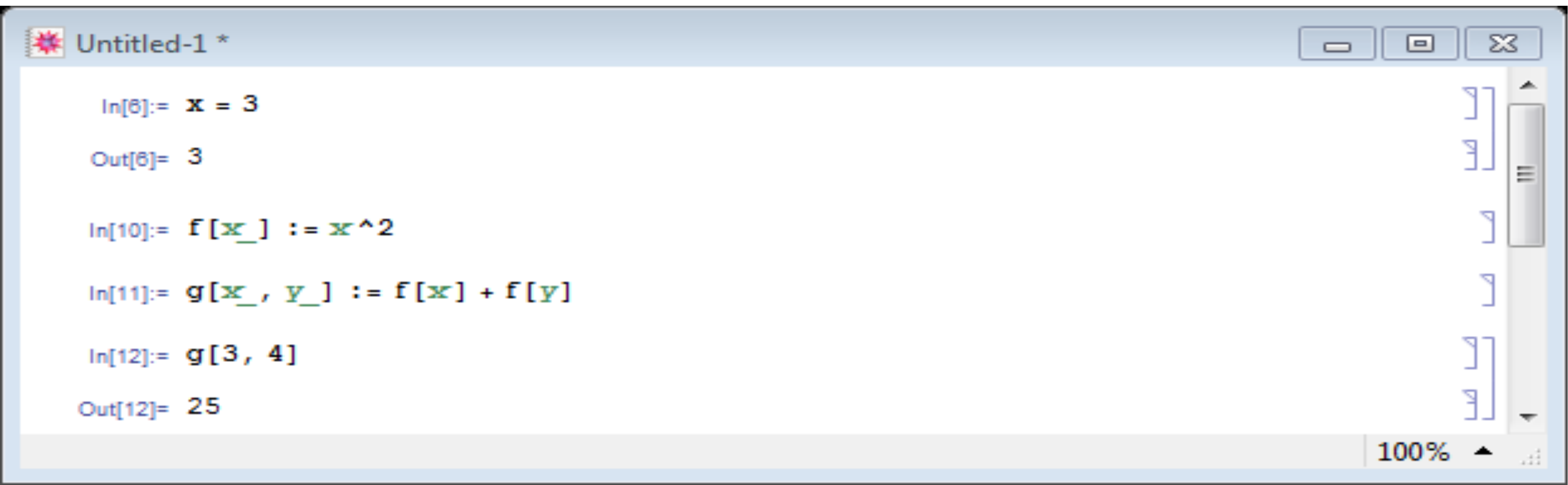
Out[46]= $expr^2$

- g[x_, y_] := f[x] + f[y]

  In[49]:= g[3, 4]

  Out[49]= 25

- Always use := to define functions, otherwise the variables on the right-hand side may not represent the associated expressions on the left-hand side, since they will be evaluated before the rule is defined.



How to Clear Definitions

- When you set a value to a symbol, that value will be used for the symbol for the entire Mathematica session. Assign values to two symbols (x and y) and observe their sum:

```
In[3]:= x = 5;
y = 7;
x + y
Out[3]= 12
```

- Use Clear to clear the definitions for x and y:

```
Clear[x, y]
```

- Observe that x and y no longer have values associated with them:

```
In[5]:= Expand[(x + y)^2]
```

Out[5]= $x^2 + 2\,x\,y + y^2$

- This command clears all the definitions made during the current Mathematica session:

  Clear["Global`*"]

- Use ClearAll to clear not only the values and definitions of symbols but also the attributes and messages associated with them.

- You can also use Unset (=.) to clear any values or definitions made to a symbol:

  In[20]:= x = 5

  Out[20]= 5

  x =.

  ?x

- Remove will remove a symbol completely until it is referenced again:

Remove[x]

?x

  Information::notfound: Symbol x not found. >>

How to Map a Function over a List

- First set up a list of the integers from 1 to 5:

  In[293]:= mylist = Range[5]

  Out[293]= {1, 2, 3, 4, 5}

- You can map a function over every element of the list using Map;

  In[294]:= Map[f, mylist]

  Out[294]= {f[1], f[2], f[3], f[4], f[5]}

- Most mathematical functions have the Listable property:

  In[297]:= Sin[mylist]

  Out[297]= {Sin[1], Sin[2], Sin[3], Sin[4], Sin[5]}

- Map does not just operate on lists. It can be used for any expression:

  In[301]:= Map[f, a + b + c + d]

  Out[301]= f[a] + f[b] + f[c] + f[d]

- Apply is another functional programming operation. It replaces the head of an expression:
- You can see how this works using two undefined functions, f and g:

```
In[302]:= Apply[f, g[a, b, c, d]]
Out[302]= f[a, b, c, d]
```

- In[304]:= {Plus[a, b, c], Times[a, b, c], List[a, b, c]}

```
Out[304]= {a + b + c, a b c, {a, b, c}}
```

- In[305]:= Apply[Times, a + b + c]

```
Out[305]= a b c
```

- Mod finds the remainder when dividing the first number of an ordered pair by the second:

```
In[307]:= Mod[10, 4]
Out[307]= 2
```

function1.pdf - Adobe Reader

File Edit View Document Tools Window Help

1 / 1 73% Find

```
In[18]:= mylist = Range[2, 6]

Out[18]= {2, 3, 4, 5, 6}

In[24]:= Range[10]

Out[24]= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

In[26]:= Clear[f]

In[27]:= Map[f, mylist]

Out[27]= {f[2], f[3], f[4], f[5], f[6]}

In[28]:= Sin[mylist]

Out[28]= {Sin[2], Sin[3], Sin[4], Sin[5], Sin[6]}

In[29]:= Map[f, a + b + c + d]

Out[29]= f[a] + f[b] + f[c] + f[d]

In[30]:= Apply[f, g[a, b, c, d]]

Out[30]= f[a, b, c, d]

In[31]:= Apply[Times, a + b + c]

Out[31]= a b c

In[32]:= Mod[10, 3]

Out[32]= 1
```

How to Work with Pure Functions

- The most transparent way to define a pure function is with Function. The first argument is a list of arguments, and the second is a function.

In[1]:= f = Function[{x, y}, x + y]

Out[1]= Function[{x, y}, x + y]

- In[2]:= f[3, 4]

Out[2]= 7

- You do not need to give the function a name to use it:

- In[3]:= Function[{x, y}, x + y][3, 4]
- Out[3]= 7
- In[4]:= g = (#1 + #2) &

Out[4]= #1 + #2 &

In[5]:= g[3, 4]

Out[5]= 7

- The advantage of a pure function is that it does not require a separate definition or a name:

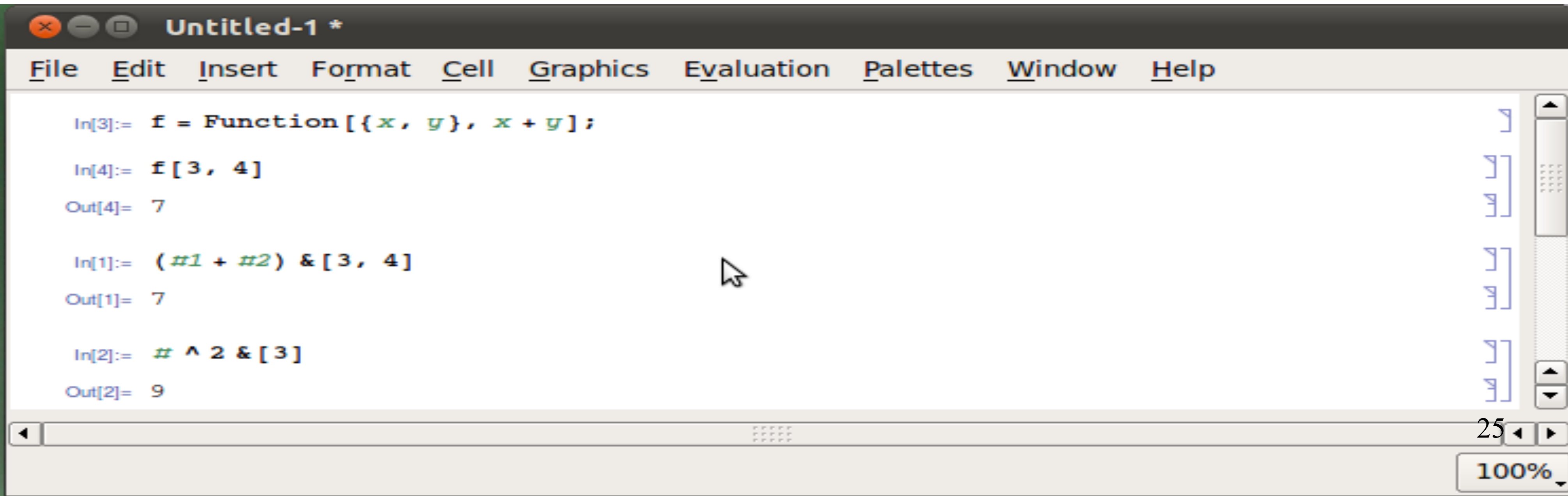
  In[6]:= (#1 + #2) &[3, 4]

  Out[6]= 7

- If the pure function only has one argument, you can use # instead of #1. This function squares its argument:

  In[7]:= #^2 &[3]

  Out[7]= 9

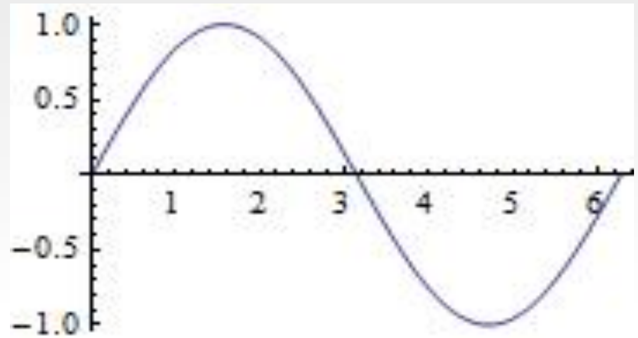


How to Enter Ranges and Options for Functions

- Here, Sin[x] is the first argument in Plot, while the second argument {x, 0,2π} gives the variable and the range for the plot:

In[7]:= Plot[Sin[x], {x, 0, 2π }]

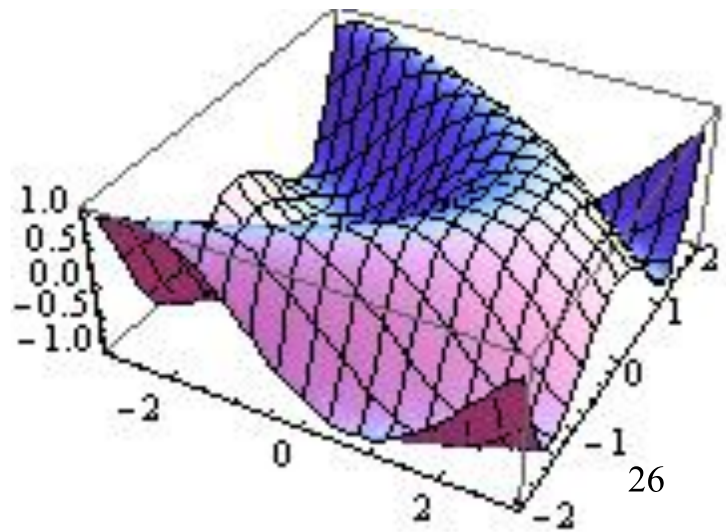
Out[7]=

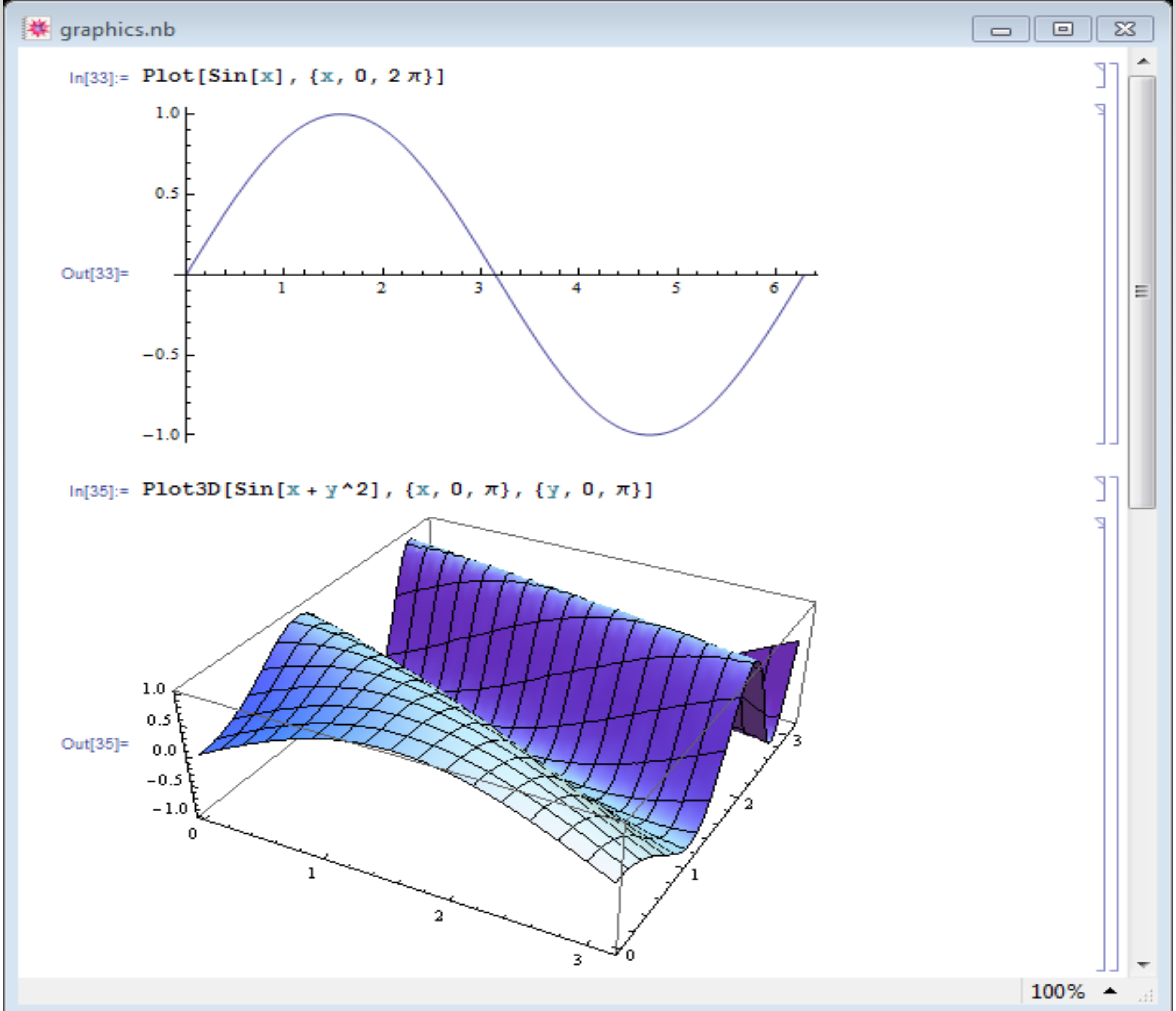


- When plotting functions of two variables, the ranges for each variable are entered in the second and third arguments:

In[6]:= Plot3D[Sin[x + y^2], {x, -3, 3}, {y, -2, 2}]

Out[6]=




graphics.nb
In[33]:= Plot[Sin[x], {x, 0, 2 π}]
Out[33]=
1.0
0.5
−0.5
−1.0
1
2
3
4
5
6
In[35]:= Plot3D[Sin[x + y^2], {x, 0, π}, {y, 0, π}]
Out[35]=
1.0
0.5
0.0
−0.5
−1.0
0
1
2
3
0
1
2
3
100%

How to Create Lists

- Use the shorthand notation {} to make a list:

In[1]:= {2, 3, 5, 6}

Out[1]= {2, 3, 5, 6}

- Or use List, which automatically is changed to {}:

In[2]:= List[2, 3, 5, 6]

Out[2]= {2, 3, 5, 6}

- Use Range with one argument to create a list of integers starting at 1:

In[3]:= Range[10]

Out[3]= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

- In[4]:= Range[5, 10]

Out[4]= {5, 6, 7, 8, 9, 10}

- In[5]:= Range[5, 10, 1/2]

Out[5]= {5, 11/2, 6, 13/2, 7, 15/2, 8, 17/2, 9, 19/2, 10}

- In[6]:= Range[10]^2

Out[6]= {1, 4, 9, 16, 25, 36, 49, 64, 81, 100}

- In[7]:= Table[i^2, {i, 10}]

Out[7]= {1, 4, 9, 16, 25, 36, 49, 64, 81, 100}

- In[9]:= NestList[f, x, 3]

Out[9]= {x, f[x], f[f[x]], f[f[f[x]]]}

- In[10]:= Array[f, 4]

Out[10]= {f[1], f[2], f[3], f[4]}

- Use List to create lists of strings:
- In[12]:= List[{"a", "b", "c"}, {"you", "are", "good"}]

Out[12]= {{"a", "b", "c"}, {"you", "are", "good"}}

- A matrix in Mathematica is a list of lists.
- Use RandomInteger to create a 4X4 matrix of random integers between 0 and 10 (stored as m):
- In[13]:= m = RandomInteger[10, {4, 4}]

Out[13]= {{5, 0, 2, 5}, {3, 2, 8, 6}, {9, 9, 8, 0}, {2, 2, 3, 4}}

- Use MatrixForm to see m as a 2D matrix:
- MatrixForm[m]

Out[14]//matrixform=

$$\begin{pmatrix} 5 & 0 & 2 & 5 \\ 3 & 2 & 8 & 6 \\ 9 & 9 & 8 & 0 \\ 2 & 2 & 3 & 4 \end{pmatrix}$$

- You can directly apply math functions to a list:

  In[15]:= Sqrt[{1, 2, 3, 4}]

  Out[15]= {1, √2, √3, 2}

In[16]:= 1 + {{a}, {a, b}, {a, b, c}}^2

Out[16]= {{1 + a^2}, {1 + a^2, 1 + b^2}, {1 + a^2, 1 + b^2, 1 + c^2}}

- In[17]:= Max[{1, 2, 3, 4}]

  Out[17]= 4
- In[18]:= Length[{a, b, c}]

  Out[18]= 3
- This uses Map to apply Length to each sublist:
- In[20]:= Map[Length, {{a}, {a, b}, {a, b, c}}]

  Out[20]= {1, 2, 3}

How to Get Elements of Lists

- In[256]:= v = Range[10]^2

  Out[256]= {1, 4, 9, 16, 25, 36, 49, 64, 81, 100}
- In[257]:= Part[v, 3]

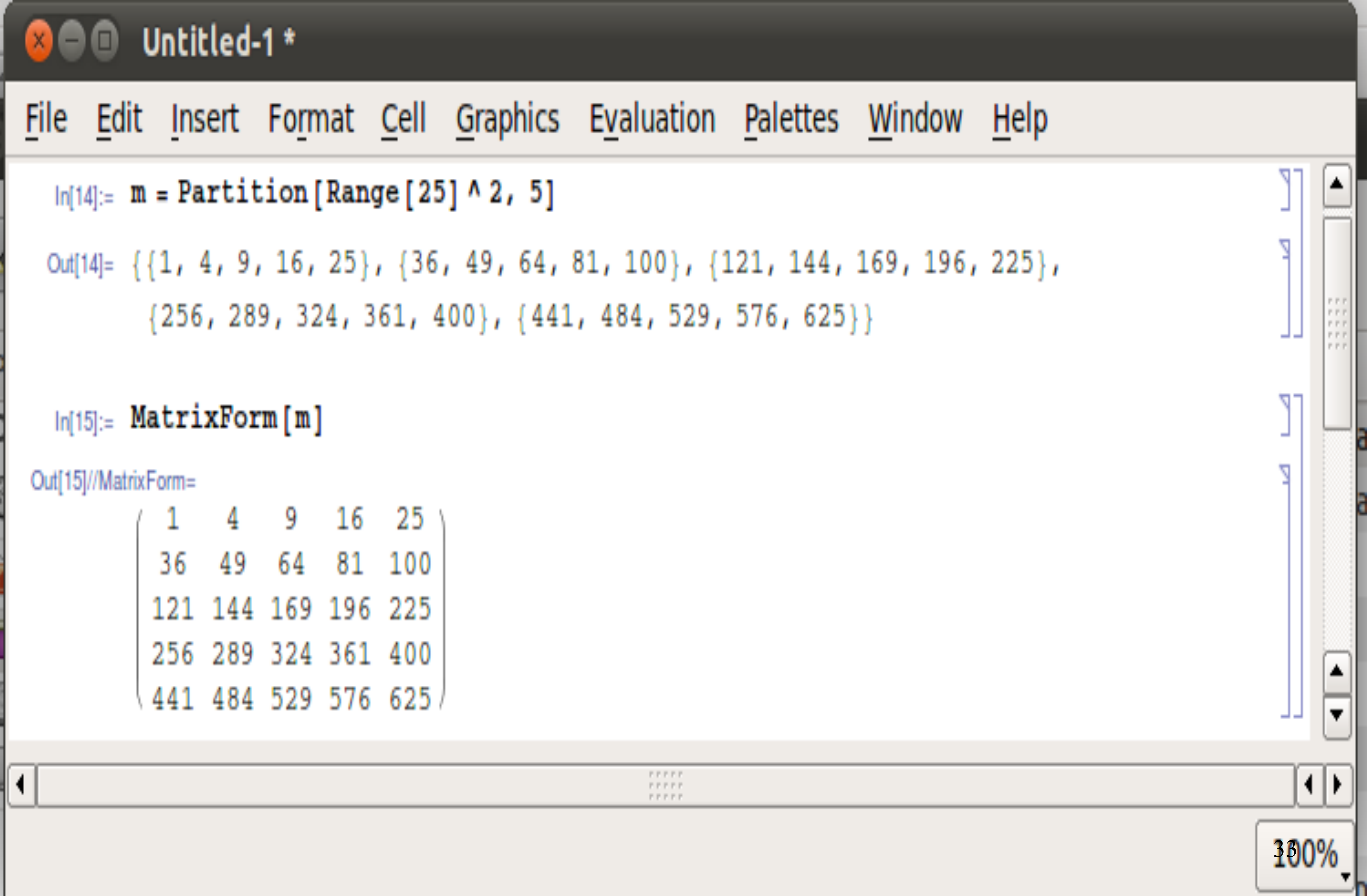
  Out[257]= 9
- In[260]:= v[[5 ;; 8]]

  Out[260]= {25, 36, 49, 64}
- In[261]:= v[[-7 ;;]]

  Out[261]= {16, 25, 36, 49, 64, 81, 100}
- You can pick out elements from a matrix just like you would from a list.
- In[262]:= m = Partition[Range[25]^2, 5]

  Out[262]= {{1, 4, 9, 16, 25}, {36, 49, 64, 81, 100}, {121, 144, 169, 196, 225}, {256, 289, 324, 361, 400}, {441, 484, 529, 576, 625}}
- In[264]:= m[[1]]

  Out[264]= {1, 4, 9, 16, 25}


Untitled-1 *
File Edit Insert Format Cell Graphics Evaluation Palettes Window Help
In[14]:= m = Partition[Range[25] ^ 2, 5]
Out[14]= {{1, 4, 9, 16, 25}, {36, 49, 64, 81, 100}, {121, 144, 169, 196, 225}, {256, 289, 324, 361, 400}, {441, 484, 529, 576, 625}}
In[15]:= MatrixForm[m]
Out[15]//MatrixForm=
1 4 9 16 25
36 49 64 81 100
121 144 169 196 225
256 289 324 361 400
441 484 529 576 625
100%

How to Combine and Rearrange Lists

In[1]:= v = {3, 1, 3, 2, 5, 4}

Out[1]= {3, 1, 3, 2, 5, 4}

In[2]:= Sort[v]

Out[2]= {1, 2, 3, 3, 4, 5}

Use Union to sort v and remove any duplicates:

In[3]:= Union[v]

Out[3]= {1, 2, 3, 4, 5}

In[5]:= v = Union[v]

Out[5]= {1, 2, 3, 4, 5}

In[7]:= Reverse[v]

Out[7]= {5, 4, 3, 2, 1}

In[11]:= Partition[v, 2]

Out[11]= {{1, 2}, {3, 4}}

In[15]:= Join[{a, b}, {c, d, e}, {f, g, h}]

Out[15]= {a, b, c, d, e, f, g, h}

How to Perform Operations on Lists

You can add two lists of the same length element by element:

In[49]:= {a, b} + {x, y}

Out[49]= {a + x, b + y}

In[50]:= c + {a, b}

Out[50]= {a + c, b + c}

In[51]:= k {a, b}

Out[51]= {a k, b k}

In[43]:= v = Range[5]

Out[43]= {1, 2, 3, 4, 5}

In[45]:= Append[v, x]

Out[45]= {1, 2, 3, 4, 5, x}

In[44]:= Prepend[v, x]

Out[44]= {x, 1, 2, 3, 4, 5}

In[46]:= Insert[v, x, 3]

Out[46]= {1, 2, x, 3, 4, 5}

How to Create and Use Rules

- The short form for a rule uses a right arrow, which you get by typing ->
- Use /. to use a rule with an expression:

In[57]:= 2 x + 1 /. x -> 3

Out[57]= 7

- In[59]:= 2 x + y /. {x -> 3, x -> 4}

Out[59]= 6 + y

- You can also use a rule to replace larger parts of an expression:

In[62]:= 9 + x^2 - 9 x^3 /. x^2 - 9 x^3 -> 3 y

Out[62]= 9 + 3 y

- In[18]:= 1 + f[x] + f[y] /. f[x_] -> x^2

Out[18]= $1 + x^2 + y^2$

How to Use Rule Solutions

- In[77]:= t = Solve[x^2 == 9, x]

Out[77]= {{x -> -3}, {x -> 3}}

- In[2]:= v = Solve[{x^2 + y^2 + z^2 == 9, y == x, y == z}, {x, y, z}]

Out[2]= {{x -> -√3, y -> -√3, z -> -√3}, {x -> √3,y -> √3, z ->√ 3}}

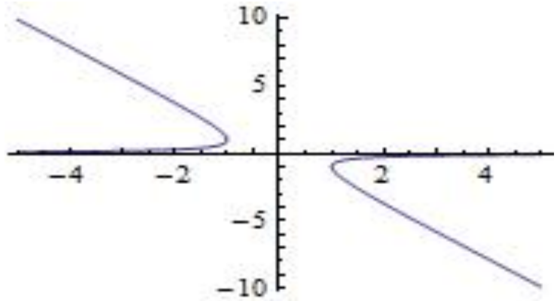
- eqn = x^2 + 2 a x + 1 == 0;

In[97]:= sol = Solve[eqn, x]

Out[97]= {{x -> -a - √-1 + a²}, {x -> -a + √-1 + a²}}

- Plot[x /. sol, {a, -5, 5}]

Out[70]=



Thank You