# Lesson 02

## Defining functions

Sometimes we may need to use the same function over and over again and it may be handy to give the function a name.

- ) Example: we need to compute the position y of an object in free fall at a certain instant of time t and for various initial positions yo and velocities vo.
Choosing a framework where up is positive, g (pointing down) will be negative. We can then define the function, as

```
Clear[y,g];
g = - 9.8;
y[t_, yo_, vo_] :=  yo + vo*t + (g/2.)*t^2;
```

And compute the position:

```
y[ 1, 50, 10 ]
y[ 2, 50, 10 ]
y[ 2 ,50, - 10]
```

## Lists, Table, Manipulating lists

### List

```
list = {4,6,2,3,9,7,1};
1/list
list!
3.*list

Total[ list ]

Max[ list ]
Min[ list ]

Sort[list]
Reverse[ Sort[ list ] ]

Length[ list ]
Position[ list, Max[ list ] ]
Position[ list, Last[ list ] ]
```

### Table (vector)

The Table function is used to generate lists

-) List of numbers from 5 to 10

```
t1=Table[k, {k, 5, 10}]
MatrixForm[t1]
```

-) List of the five first perfect squares
t2=Table[k^2, {k, 1, 5}]

-) List of odd numbers in decreasing order starting from 13
t3=Table[k, {k, 13, 1, -2}]

-) Construct a list of values of y[t_, yo_, vo_] for yo=50, vo=5, and t = 0, 0.5, 1.0, 1.5., 2.0, 2.5, 3.0, 3.5, 4.0
Table[ y[0.5 k, 50, 5], {k,0,8} ]

## Working with the list

In a table called "tab", tab[[ n ]] gives the nth element of the list, and tab[[ - n ]] gives the nth element from the end of the list. The command Part[ tab, n] also has the same role.

evens = Table[ k, {k,0,16,2}]
evens[[ 3 ]]
Part[ evens, 3 ]
evens[[ - 2 ]]
Part[evens, -2]

Other ways to manipulate a list:

Take[evens, 2]
Take[evens, -3]
Take[evens, {2, 4}]

Drop[evens, 3]
Drop[evens, - 2]

Union[ t1, t2 ]
Intersection[ t1, t2 ]
Join[ t1, t2 ]

# Matrices

## Table (matrix)

A matrix is an array of numbers arranged in rows and columns

m1 = { {1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12} };
MatrixForm[ m1 ]

m2=Flatten[ m1 ]
MatrixForm[ m2 ]

m = Table[Table[ a[row, column], {column, 1, 3 } ], {row, 1, 4} ]
MatrixForm[ m ]

Matrices can be combined using the operations of addition, subtraction, scalar, and matrix multiplication.
The operation of matrix multiplication is represented by a period (.).

REVIEW (do it in a piece of paper and then compare with *Mathematica*):
-) If you do not remember how to deal with matrices, *Mathematica* can help.
MM = { { m11, m12, m13}, {m21, m22, m23}, {m31, m32, m33} };
PP= { { p11, p12, p13 }, {p21, p22, p23}, {p31, p32, p33} };
u = { u1, u2 , u3 };
MatrixForm[ MM + PP ]
MatrixForm[ MM - PP ]
MatrixForm[ MM. PP ]
MatrixForm[MM.u]

A = { { 1, 2, 3}, {4, 5, 6}, {7, 8, 9} };
B= { { 2, 1, 5 }, {4, 7, 2}, {1, 3, 2} };
MatrixForm[ A + B ]
MatrixForm[ A - B ]
MatrixForm[ 3 A + B ]
MatrixForm[ A . B ]

-) In matrix B, what is the element in the 3rd row and in the 2nd column?
B[ [ 3, 2 ] ]
-) Add the element B in 3rd row and in the 2nd column with the element of A in the second row and in the first column
B[ [ 3, 2 ] ] + A[ [ 2, 1 ] ]
-) Multipply matrix A by the vector vec = { 5, 5, 5 }
B . vec

-) diag=DiagonalMatrix[ { 1,4, 7 } ]
MatrixForm[ diag ]
diag // MatrixForm

-) id=IdentityMatrix[ 4 ];
MatrixForm[ id ]

REVIEW:
VECTORS
u = { u1, u2 , u3 };
v = { v1, v2, v3 };
-) Dot product (example: W=**F.d**)
u.v
-) Cross product (example: **F**=q **vxB**)
Cross[u,v]
-) Outer (tensor product)
ou=Outer[ Times, u, v ] ;
MatrixForm[ou]

REVIEW:
mat = { {1,2,3}, {4,2,2}, {5,1,7} };
MatrixForm[mat]

-) Transpose[ mat ]
MatrixForm[ Transpose[ mat ] ]

-) Determinant (system of linear equations has solution if det is nonzero, Jacobian in integrals, eigenvalues of a matrix)
Det[ mat ]

-) mat.mit=**1** implies that *mit* is the inverse matrix of *mat*
A matrix is invertible only if its det is nonzero
Inverse[ mat ]

-) Tr[ mat ]

## Table Form

-) Make a matrix with the values of y[t,50,5] for t = 0, 0.5, 1.0, 1.5., 2.0, ... 4.0, where the first column gives the time and the second the value y of the position.

ty=Table[ {0.5 k , y[ 0.5 k, 50, 5 ] }, { k, 0, 8} ]
TableForm[ ty ]

form = Table[ { PaddedForm[ 0.5 k, {3, 1} ] , PaddedForm[ y[ 0.5 k, 50, 5 ], { 7, 3 }] }, { k, 0, 8} ]
TableForm[ form ]
TableForm[ form, TableHeadings ->{None, { "  time", "  position"} } ]

ListPlot[ ty ]
ListPlot[ ty, Joined -> True]

---

# Random numbers

Table with 10 random numbers:

-) Integers 0 or 1
Table[ RandomInteger[ ] , { k, 1, 10 } ]

-) Integers between 2 and 5
Table[ RandomInteger[ { 2 , 5 } ] , { k, 1, 10 } ]

-) Reals between 0 and 1
Table[ RandomReal[ ] , { k, 1, 10 } ]

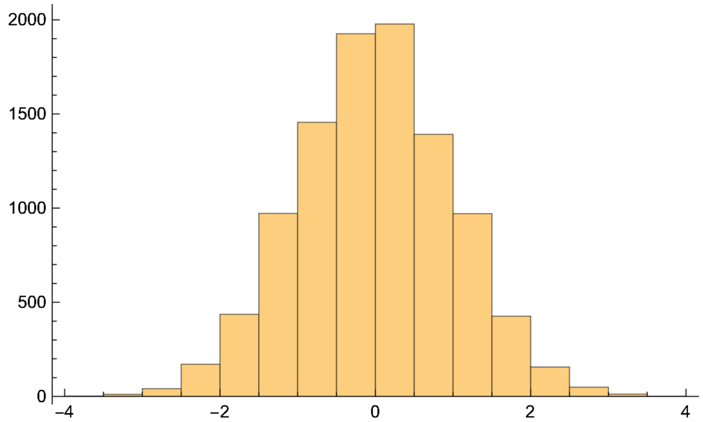
-) Reals between 1 and 4
Table[ RandomReal[ { 1 , 4 } ] , { k, 1, 10 } ]

-) Random numbers from a normal (Gaussian, Bell) distribution with mean 0 and standard deviation 1
Table[ RandomReal[ NormalDistribution[ 0, 1 ] ], { k, 1, 10 } ]

## Histogram

In[∘]:=
```
ran = Table[RandomReal[NormalDistribution[0, 1]], {k, 1, 10 000}];
Histogram[ran]
```

Out[∘]=




In[∘]:=
```
Sum[ran⟦k⟧, {k, 1, 10 000}] / 10 000
Mean[ran]
```

Out[∘]= -0.00896028

Out[∘]= -0.00896028

In[∘]:=
```
(*variance*)
Sum[ren⟦k⟧^2, {k, 1, 10 000}] / 10 000 - (Sum[ren⟦k⟧, {k, 1, 10 000}] / 10 000)^2

(*standard deviation *)
Sqrt[Sum[ren⟦k⟧^2, {k, 1, 10 000}] / 10 000 - (Sum[ren⟦k⟧, {k, 1, 10 000}] / 10 000)^2]
```

Out[∘]= 100.267

Out[∘]= 10.0133