Lesson 03

Do-loop

A loop allows for a set of instructions to be executed over and over again.

```
Do[ Print[ "hello" ], { k, 1, 4 } ]

Do[
Print[ "hello" ]
, { k, 1, 4 } ]

Do[
Print[ "hello" ] ;
, { k, 1, 4 } ] ;

Do[
Print[ k^2 ]
,{k,1,4} ]

(* Mathematica does not understand what I want to do below. I need to separate the lines with semicolons *)
Do[
a=k^2
Print[ a ]
,{k, 1, 4} ]

(* These alternatives will be fine *)
Do[
a=k^2;
Print[ a ]
,{k, 1, 4} ]

Do[
a=k^2 ;
Print[ a ] ;
,{k, 1, 4} ]

(* More general examples *)
Clear[ a, b, ab ];
Do[
a = k^2 ;
b = 1 + k ;
ab = a*b ;
Print[ { a, b, ab } ] ;
, { k, 1, 4 } ] ;

Clear[ a, b, ab ];
Do[
```

```
a[ k ] = k^2 ;
b[ k ] = 1 + k ;
ab[ k ] = a[ k ]*b[ k ] ;
Print[ { a[ k ], b[ k ], ab[ k ] } ] ;
, { k, 1, 4 } ] ;

a[ 2 ] +ab[ 4 ]
```

-) Construct a vector with 5 elements, where they are all zero.

```
Clear[vec]
vec= Table[ 0, {k,1,5} ]
```

```
Clear[ elem,  vec ];
Do[ elem[k] = 0, {k, 1, 5 } ];
vec = Table[ elem[ k ], {k, 1, 5} ]
```

-) Construct a vector with 30 elements, such that the first and last elements are equal to 10, the remaining elements at even positions are equal to 2 and the remaining elements at odd positions are equal to -1.

```
Clear[ elem,  vec ];
elem[ 1 ] =10;
elem[ 30 ] =10;
Do[ elem[ k ] = 2, { k, 2, 28, 2} ];
Do[ elem[ k ] = - 1, { k, 3, 29, 2} ];
vec = Table[ elem[ k ], { k , 1, 30 } ]
```

-) Construct a 5 X 5 upper triangular matrix of 1s with 0s below the main diagonal.

```
Clear[ elem, triag ];
Do[ Do[ elem[ j, k ] = 0, { k, 1, 5 } ], { j, 1, 5 } ];
Do[ Do[ elem[ j, k ] = 1, {k, j, 5 } ], { j, 1, 5 } ];
triag =  Table[ Table[ elem[ j, k ], { k, 1, 5 } ], { j, 1, 5 } ];
MatrixForm[ triag ]
```

-) Given the matrix

```
Clear[m1];
m1 = { { 1, 2, 3, 4 }, { 1, 2, 3, 4 }, { 1, 2, 3, 4 } };
MatrixForm[ m1 ]
```

```
tm1 = Transpose[ m1 ];
MatrixForm[ tm1 ]
```

Use do-loops to compute its transpose

```
Clear[ mt, mTrans];
Do[
Do[
mt[ j, k] = m1[ [ k, j ] ];
, { k, 1, 3 } ];
,{ j, 1, 4 } ];
mTrans = Table[ Table[ mt[ j, k ], { k, 1, 3 } ], { j, 1, 4 } ];
MatrixForm[ mTrans ]
MatrixForm[ Transpose[ m1 ] ]
```

If-Then-Else statement

If[condition, true, false]

-) Clear[a];
a=2;
If[a>0, Print[" a is positive "], Print[" a is negative "]]

-) Construct a 5 X 5 upper triangular matrix of 1s with 0s below the main diagonal.
Clear[triag];
triag = Table[Table[If[k >= j, 1, 0], { k, 1, 5 }], { j, 1, 5 }];
MatrixForm[triag]

-) Define a function g[x] which returns x/2 if x is even and 3 x + 1 if x is odd
Clear[g, x];
g[x_] := If[Mod[x, 2] == 0, x/2, 3 x + 1];
g[4]
g[3]

-) Define a function h[x] which is 1 when x>0, -1 when x<0, and 0 at x=0. Here we nest the If-function
Clear[h, x];
h[x_] := If[x>0, 1, If[x<0, -1, 0]];
h[2]
h[- 3]
h[0]

-) Compute the total number of numbers between 1 and 20 which are NOT multiples of 2, 3 or 5.
Clear[howmany];
howmany = 0;
Do[
If[Mod[k, 2] != 0 && Mod[k, 3] != 0 && Mod[k, 5] != 0, howmany = howmany +1];
,{ k, 1, 20 }]
Print["The total number is ", howmany];

-) Make a list with the numbers between 1 and 20 which are NOT multiples of 2, 3 or 5.
Clear[howmany, num];
howmany = 0;
Do[
If[Mod[k, 2] != 0 && Mod[k, 3] != 0 && Mod[k, 5] != 0, { howmany = howmany +1, num[howmany] = k }];
,{ k, 1, 20 }]
Table[num[k], {k, 1, howmany }]

Export and Import

Create a list with 10 pairs of even and odd numbers

Clear[list];
list = Table[{ k, k + 1}, { k, 2, 20, 2}]

Export this list to file named "PairEvenOdd.dat" which should be placed in the same directory where you have been saving your work.

First, check in which directory you currently are:
Directory[ ]

If this is not the one you want, set the correct one:
SetDirectory["Yeshiva"]
SetDirectory["ANGEL"]
SetDirectory["Presentations_Computational"]

Now we can export the data as
Export["PairEvenOdd.dat" , list]

Go to the folder and verify if it is really there. If it is, close *Mathematica* and restart it again.
Let us now import the data that we saved in the file PairEvenOdd.dat. Before doing that, we need to be sure again that we are in the correct directory. If so, then
Clear[newlist];
newlist = Import["PairEvenOdd.dat" , "Table"]

Try to solve the last problem of Assignment03 together in class.