```
(* - Lists*
 1. How to generate:  Table command; nested lists
 2. Length and Dimensions
 3. REARRANGING LISTS: Sort, Rotate, etc.
  4. Restructuring Lists: Flatten, Drop, etc.
  5. Combining Lists: Join, Union, Intersection
 6. Operating on lists: the Map command
 7. Lists as vectors and matrices

*)

(* 1. How to generate:

   Table[expr, {imax}] generates a list of imax copies of
    expr. Table[expr, {i, imax}] generates a list of the values of expr
    when i runs from 1 to imax. Table[expr, {i, imin, imax}] starts with i =
 imin. Table[expr, {i, imin, imax, di}] uses steps
  di. Table[expr, {i, imin, imax}, {j, jmin, jmax}, ... ]
  gives a nested list. The list associated with i is outermost
    *)


Table[i, {i, 10, 21, 3}]


{10, 13, 16, 19}

Length[%]

4

ColumnForm[%%]

10
13
16
19

Table[Sin[n x], {n, 1, 7, 2}]

{Sin[x], Sin[3 x], Sin[5 x], Sin[7 x]}
```

(*nested*)

```
Table[x^i + y^j, {i, 3}, {j, 4}]
```

$$\left\{\left\{x+y,\ x+y^2,\ x+y^3,\ x+y^4\right\},\ \left\{x^2+y,\ x^2+y^2,\ x^2+y^3,\ x^2+y^4\right\},\ \left\{x^3+y,\ x^3+y^2,\ x^3+y^3,\ x^3+y^4\right\}\right\}$$

```
Dimensions[%]
```

$$\{3,\ 4\}$$

```
TableForm[%%]
```

$$\begin{array}{cccc} x+y & x+y^2 & x+y^3 & x+y^4 \\ x^2+y & x^2+y^2 & x^2+y^3 & x^2+y^4 \\ x^3+y & x^3+y^2 & x^3+y^3 & x^3+y^4 \end{array}$$

```
MatrixForm[%%%]
```

$$\begin{pmatrix} x+y & x+y^2 & x+y^3 & x+y^4 \\ x^2+y & x^2+y^2 & x^2+y^3 & x^2+y^4 \\ x^3+y & x^3+y^2 & x^3+y^3 & x^3+y^4 \end{pmatrix}$$

```
(* REARRANGING LISTS *)
Sort[{abd, abc, ab, a, d, 10, a, 10, 5.5,
        1/4, 23/2, ada}]
```

$$\left\{\frac{1}{4},\ 5.5,\ 10,\ 10,\ \frac{23}{2},\ a,\ a,\ ab,\ abc,\ abd,\ ada,\ d\right\}$$

```
mylist = Table[i, {i, 10}]
```

$$\{1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9,\ 10\}$$

```
RotateLeft[mylist]
```

$$\{2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9,\ 10,\ 1\}$$

```
RotateRight[mylist, 3]
```

$$\{8,\ 9,\ 10,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7\}$$

```
Permutations[{a, b, c}]
```

```
{{a, b, c}, {a, c, b}, {b, a, c}, {b, c, a}, {c, a, b}, {c, b, a}}
```

(* Restructuring Lists *)

```
Clear[mylist, c]
```

```
mylist = Table[c[i, j], {i, 3}, {j, 4}]
```

```
{{c[1, 1], c[1, 2], c[1, 3], c[1, 4]},
 {c[2, 1], c[2, 2], c[2, 3], c[2, 4]}, {c[3, 1], c[3, 2], c[3, 3], c[3, 4]}}
```

```
Flatten[%]
```

```
{c[1, 1], c[1, 2], c[1, 3], c[1, 4], c[2, 1],
 c[2, 2], c[2, 3], c[2, 4], c[3, 1], c[3, 2], c[3, 3], c[3, 4]}
```

Flatten[list] flattens out nested lists. Flatten[list, n] flattens
to level n. Flatten[list, n, h] flattens subexpressions with head h.
*)

```
(*Drop[list, n] gives list with its first n elements dropped
 Drop[list, {n}] gives list with its n"th"element dropped.
  Take[list, n] gives the first n elements of list
 Take[list, {m, n}] gives elements m through n of list
 HW: create a simple list; try all this*)
```

(*Element extraction*)

```
list = {a, b, c, d};
list[[2]]
```

```
General::spell1 : Possible spelling error: new
   symbol name "list" is similar to existing symbol "List". More…
```

```
b
```

```
Clear[list]; list = {{a, b}, {c, d}}
```

```
{{a, b}, {c, d}}
```

```
list[[2]]
```

```
{c, d}
```

```
list[[1, 2]]
```

```
b
```

```
MatrixForm[list]
```

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

(* Combining Lists *)

```
list1 = Table[i, {i, 3}]
```

{1, 2, 3}

```
list2 = Table[i, {i, 7}]
```

{1, 2, 3, 4, 5, 6, 7}

```
Join[list1, list2]
```

{1, 2, 3, 1, 2, 3, 4, 5, 6, 7}

```
Union[list1, list2]
```

{1, 2, 3, 4, 5, 6, 7}

```
Intersection[list1, list2]
```

{1, 2, 3}

(*Operating on Lists*)

```
Clear[list]
```

```
list = Table[x^n, {n, 4}]
```

$\{x, x^2, x^3, x^4\}$

```
D[list, x]
```

$\{1, 2\,x, 3\,x^2, 4\,x^3\}$

```
Sin[list]
```

$\{\text{Sin}[x], \text{Sin}[x^2], \text{Sin}[x^3], \text{Sin}[x^4]\}$

```
Sqrt[list]
```

$\{\sqrt{x}, \sqrt{x^2}, \sqrt{x^3}, \sqrt{x^4}\}$

```
(* Not all fuctions are listable! - Check *)
```

```
Clear[f];
```

```
Map[f, list]
```

$\{f[x], f[x^2], f[x^3], f[x^4]\}$

```
f /@ list (*same thing*)
```

$$\{f[x], f[x^2], f[x^3], f[x^4]\}$$

```
? Apply
```

```
Apply[f, expr] or f @@ expr replaces the head of expr by f. Apply[f,
   expr, levelspec] replaces heads in parts of expr specified by levelspec. More...
```

```
Plus @@ list
```

$$x + x^2 + x^3 + x^4$$

```
(* HW: write a function av[y_] which
  would find an average of a list y of arbitrary length*)


(*Pure function*)

(*Note: in f[x_]:=
 Sin[x] the variable x does not matter and can be replaced by anything. In fact,
x_ indicates a slot in which the variable will be place when evaluating Sin. The idea is
  then to skip x alltogether - this leads to a "Pure function", analog of an operator*)


mysin := Sin[#] &
(*# explicitly indicates a slot and & indicates the pure function;
note that [] does not enter the definiton of mysin,
but will be needed when it is applied*)

mysin[z]
```

```
Sin[z]
```

```
mysin[Pi / 4]
```

$$\frac{1}{\sqrt{2}}$$

```
(* more useful for differential operators*)

der2 := D[#, x, x] &

der2[x^2]
```

```
2
```

```
Clear[a]; der2[E^(a x)]
```

$$a^2\, e^{a x}$$

```
(*other examples: anything can go into the slot,
not only numbers. Below are Lists and Graphics*)

myav := (Plus @@ #) / Length[#] & (*average of a list*)

list = {1, 2, 3, 4};

myav[list]
```

$$\frac{5}{2}$$

```
disp := Display["t.ps", #, "EPS"] &

(*this creates a nice postscript outside Mathematica. E.g., if you like graphics in fig,
you write disp[fig] and will get a postscript file t.ps with the picture*)


(*pure functions are very good for repeated applications*)

f := 1 / (1 + #) &

f[x]
```

$$\frac{1}{1+x}$$

```
f[%]
```

$$\frac{1}{1+\frac{1}{1+x}}$$

```
f[%]
```

$$\frac{1}{1+\frac{1}{1+\frac{1}{1+x}}}$$

```
(*etc.*)

(*there is special Nest command to do that*)

? Nest
```

Nest[f, expr, n] gives an expression with f applied n times to expr. More…

```
Nest[f, x, 4]
```

$$\cfrac{1}{1+\cfrac{1}{1+\cfrac{1}{1+\cfrac{1}{1+x}}}}$$

```
Inverse::sing : Matrix {{a, b, c}, {1, 2, 3}, {a, b, c}} is singular. More…
Inverse[{{a, b, c}, {1, 2, 3}, {a, b, c}}]
```

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$