Basics of Mathematica Grammar (Easy)

Review of the elementary Mathematica usage

Basic input

Shift + Enter : evaluate expression
- compute 2+2

```
2 + 2

4
```

Symbols

To enter various symbols type: Esc + [something] + Esc
- try:

```
(* [something] = a *) α
(* [something] = t *) θ
(* [something] = inf *) ∞
```

[something] can be like in LaTeX. Try the following:

```
(* [something] = \in *)
```

Formulas

- Type Ctrl + / and see what it gives

$$\frac{\square}{\square}$$

- Type Ctrl + 2 and see what it gives

$$\sqrt{\square}$$

Go to "Selecting and Typing in Notebooks" to get more

Help

To get an immediate information about some function type ? before its name and run”
- try the following

```
? NIntegrate
```

The symbol * is very useful in the combination with ?”
- try this

```
? *Bessel*
```

When you don‘t remember a name of a function use Ctr + K : Auto-complete
- Type NIn and then Ctr + K

```
NIn
```

Syntax

= vs :=

There are two ways to assign a r.h.s. `=` and `:=` It is very important to fill the difference!
`=` immediately evaluates the r.h.s. and then result is assigned to l.h.s.
`:=` the r.h.s. will be evaluated each time we use the function

Compare the following definitions of functions `F1` and `F2`:

```
a = 1;
F1[x_] = x + a;
F2[x_] := x + a;
```

When evaluated they seem to be the same

```
{F1[π], F2[π]}
```

$\{1+\pi, 1+\pi\}$

But the result will be different when we change a

```
a = 2;
{F1[π], F2[π]}
```

$\{1+\pi, 2+\pi\}$

Since `:=` forces , *Mathematica*, to look back into the definition we can apply additional conditions. That is done using `/;`

The point is that when we evaluate `F2` *Mathematica* goes back to its definition.

- To make it more clear create a function which will print "I am evaluated" each time you run an expression with this function

```
a := b /; IntegerQ[b]
a := - b /; Not[IntegerQ[b]]
```

Substitutes (/. and //.) (→ and :→)

Substitutes are very important. Many calculations could be performed very efficiently by them

```
  x
—————— /. x → Zeta[3]
x^2 + 1
```

$$\frac{\zeta(3)}{\zeta(3)^2+1}$$

There are several variants of the substitute. `/.` replaces once, whereas `//.` repeatedly performs replacements until an expression no longer changes.

```
(1 + x (1 + x (1 + x))) /. 1 + x → 1
```

$$x\,(x+1)+1$$

```
(1 + x (1 + x (1 + x))) //. 1 + x → 1
```

```
1
```

Similar to `=` and `:=` one should clearly distinguish `->` and `:>`
`->` r.h.s. is immediately evaluated
`:>` r.h.s. will be evaluated each time when the substitute is applied

$$\texttt{Exp}\left[1-\frac{(1+\pi)*2-2-2\,\pi}{\pi}\right]\ \texttt{/. Exp[}a\texttt{\_] -> Exp[Simplify[a]]}$$

$$e^{1-\frac{-2-2\,\pi+2\,(1+\pi)}{\pi}}$$

The expression is not modified. That is because `Simplify` first applied to `a`, which obviously is as simple as possible. In order to tell Mathematica to simplify the expression *each time* it is replaced one should use `:>` instead

$$\texttt{Exp}\left[1-\frac{(1+\pi)*2-2-2\,\pi}{\pi}\right]\ \texttt{/. Exp[}a\texttt{\_] :> Exp[Simplify[}a\texttt{]]}$$

$$e$$

Apply Function (//, @ and /@)

Let us take some function - say `Cos`. We want to apply it to π. There are tree ways this can be done (this is convenient when you have many functions to apply)

```
Cos[π]
π // Cos
Cos@π
```

```
-1
```

```
-1
```

```
-1
```

When we want to apply a function to each term in some expression we use `/@`

```
Cos /@ (x + x^2 + x^3 + x^4)
```

$$\cos(x^4)+\cos(x^3)+\cos(x^2)+\cos(x)$$

Rather different and frequently used command `@@` - replace header

```
Sin @@ Cos[x]
Times @@ {1, 2, 3, 4, 5, 6, 7}
```

$$\sin(x)$$

```
5040
```

Patterns

Patterns are one of the most important ingredients in the Mathematica. They are used in the replacements as well as in many other circumstances. For example they can be used to reorganizing expressions

```
Collect[(Sin[x] + Exp[2 x] + Sin[x] Exp[y]) (Sin[x] + 12 Sin[3 x]), Sin[_]]
```

$$(e^y + 1)\sin^2(x) + \sin(x)\left((12\, e^y + 12)\sin(3\, x) + e^{2\, x}\right) + 12\, e^{2\, x}\sin(3\, x)$$

Patterns are used throughout Mathematica to represent classes of expressions. A simple example of a pattern is the expression `f[x_]`. This pattern represents the class of expressions with the form `f[anything]`

```
f[2, 3] + f[1] /. f[x_] → 0
```

$$f(2, 3)$$

There is a variant of this simple pattern with two underscores `f[x__]`. This pattern represents the class of expressions with the form `f[anything,...]`

```
f[2, 3] + f[1] /. f[x__] → 0
```

$$0$$

Mixed Problem

- Use only patterns and replacements to transform

```
{{M, a, {p, {}, {{l}, e}}}, S, {y, {r}}, u, p}
```

into

```
{M, a, p, l, e, S, y, r, u, p}
```

- Using help find the solution to

$$\frac{k}{k+2}y_{k+1} - y_k + \frac{k+2}{2k}y_{k-1} = 0 \quad , \quad y_0 = 0 \quad , \quad y_1 = 1$$

Hint: The function you need is called `RS....`

- Symmetrizer. Define a function Sym which symmetizes in arguments, e.g.

```
Sym[G[x, y, z]]
```

$$\frac{1}{6}\,(G[x, y, z] + G[x, z, y] + G[y, x, z] + G[y, z, x] + G[z, x, y] + G[z, y, x])$$

Hint: the function `Permutations` could be useful

Deck of Cards

- Create a list of the 52 cards of the deck `{Hearts[Ace],Hearts[2],Hearts[3],...}`

- Create a function `Shuffle:=...` which creates a shuffled deck saved as a list of 52 elements denoted `shuffledlist`
Hint: See `RandomSample`

- Create a function `take[n_]:=...` which takes the top `n` cards of the shuffled list (and displays them). You should remove those cards from `shuffledlist`. You should also be able to take n_1 cards, then n_2 more cards etc.

- Use `Mathematica` writing assistant to use ♣ instead of `Clubs` etc.
You can also use `Style[...,{Red,Large}]` etc to make the output look nicer.

Example:

```
Shuffle
```

```
extract[7]
```

```
{♠[9], ♡[9], ♢[King], ♡[2], ♡[Queen], ♢[Jacks], ♣[Jacks]}
```

```
extract[7]
```

```
{♠[Ace], ♣[Queen], ♣[4], ♢[6], ♢[Queen], ♠[10], ♢[5]}
```