**Julia Scientific Programming**

**WEEK 1:**

* **Starting with Julia Notebooks:**

#### Programming languages and why Julia is special:

Outcome :

1. Show you how Julia compares to Python, MATLAB, C, and Fortran,
2. To say why Julia's easy to learn,
3. List a few of the technical aspects in Julia that make it very special.

Comparison with other programming languages:

1. Julia is easy to learn but many other programming languages which is also easy to learn in particular Python and MATLAB.
2. Julia is fast, but again there are other programming languages that are fast particularly C and Fortran.
3. Julia is free, but there are other languages that are free such as Python.

*Julia is unique in that it is free and fast, and easy to learn.*

Why Julia is easy to learn:

* Julia programs are short.
* Speed of execution

Some Technical aspects of Julia:

1. Julia is a strongly typed and dynamic language.
2. The type system also allows functions you have multiple dispatch
3. Julia supports parallel programming. It is concurrent. It has a data model suitable for huge data sets.
4. Julia is actually a very good language to use with other languages.

#### **Getting Ready: JuliaBox**

JuliaBox----- [juliabox.org](https://juliabox.com/)

IJulia on your own Machine:

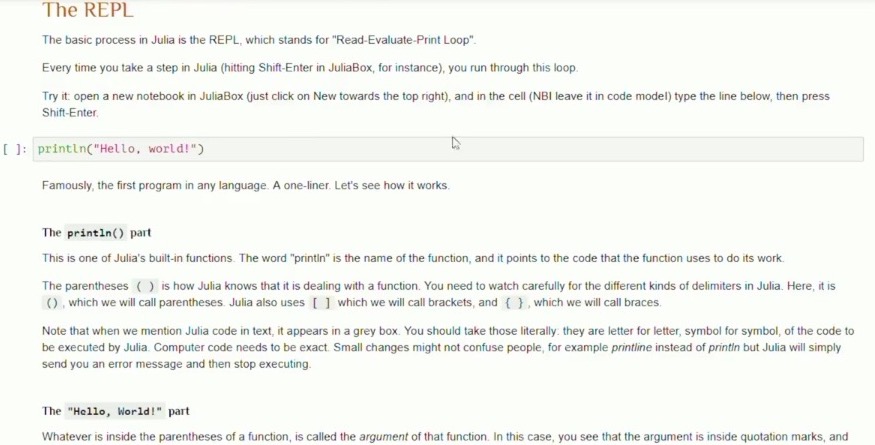
If you wanted to have it on your own machine and then why not do so, very nice capacity to have and you don't have to have to go online to use Julia.

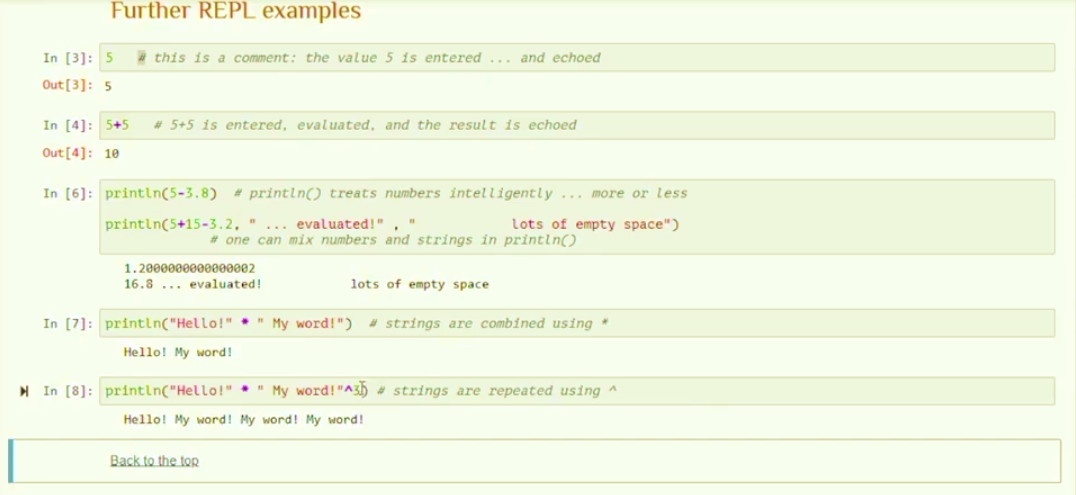
So then, you install Julia on your own computer

* 1: Start Julia on the command line and under command line,
* 2: You just add this bit of code, Pkg.add(“IJulia”)
* 3: "Using IJulia” to activate IJulia
* 4: The command line type, ("notebook") and then that will open a browser and to open a Jupyter notebook in a browser.

#### **The Julia REPL - Read, Evaluate and Print Loop:**

#### In Julia, the command line is usually called the R-E-P-L. And it stands for Read, Evaluate, Print Loop.



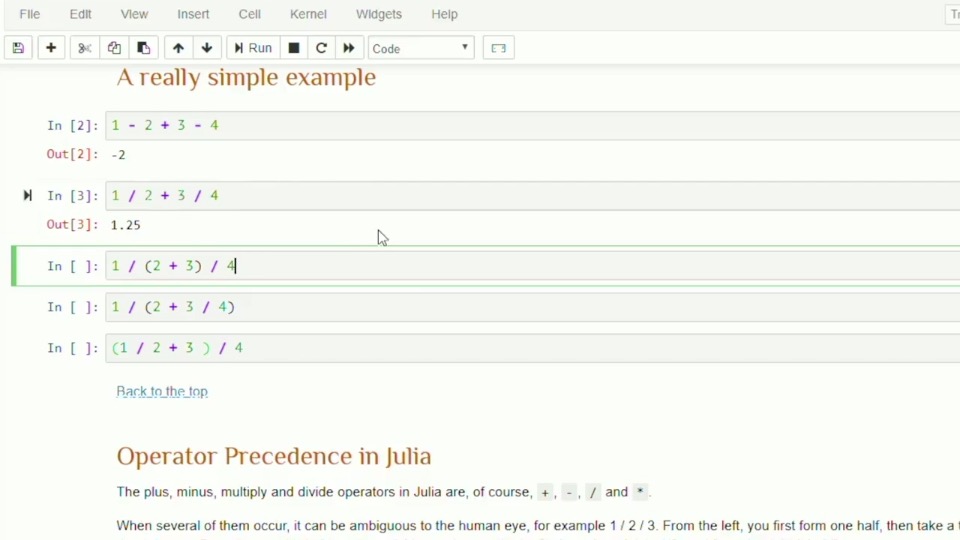


### Simple arithmetical and logical expressions in Julia

#### **Arithmetical expressions:**

Outcome:

1. To form arithmetical expressions with numbers and operators.
2. To state the order of preference for Julia's elementary arithmetical operators.
3. To work out exactly, how Julia would evaluate a given arithmetical expression.



Operator Precedence in Julia:

The Plus(+) , Minus(-), Multiply(\*) and Divide(/) are the operators in Julia.

When we have several of them then of occur it can be ambiguous, and then we can resolve the ambiguity in a number of different ways by putting parentheses in there. But Julia doesn't know

exactly where we want to put the parentheses.

So, in Julia, when we have several events so 1/2/3. So what Julia does is it simply goes from left to right. So you have 1/2 is a half and then you divide that by 3, and that should of course be 1/6 and from the right to it will give 3/2