

---

# **Julia for Digital Humanities**

***Release 0.1***

**M.Avillez**

**Jul 21, 2022**



**CONTENTS**

<b>1</b>	<b>Contents</b>	<b>3</b>
----------	-----------------	----------



**Julia for Digital Humanities** is a training programming for students and researchers in the field of digital humanities.

---

**Note:** This project is under active development.

---



## CONTENTS

### 1.1 About

Training course for students and researchers interested to learn the basics of Julia to use in their fields.

The course was initially designed for people working in Digital Humanities including the fields of Tourism, Literature, Heritage, just to mention a few.

### 1.2 Julia Command Line

### 1.3 Julia Jupyter Notebooks

### 1.4 Input/Output

#### 1.4.1 Opening and Reading a File in Julia

File handling in Julia is achieved using functions such as `open()`, `read()`, `close()`.

- `open()`: To open a file existing in an absolute path, provided as the parameter.
- `read()`: Read the contents of the file into a single string.
- `close()`: Close the file object or the variable holding the instance of an opened file.

Read the contents of a file use:

- `readline()`
- `readlines()`
- `read()`

## 1.4.2 Opening a file

### Method 1

```
f = open("cars.txt", "r")      # Opening a file in read_mode "r"
    # do some file operations
close(f)                      # close the file instance
```

### Method 2

```
open("cars.txt") do f          # opening a file in read_mode and cycle through the lines
    # do stuff with the open file instance 'f'
end
```

The opened file is automatically closed after the “do control” ends.

## 1.4.3 Reading the file contents

Read the file contents line by line (one line at a time) using `readline()` function

```
open("cars.txt") do f
    line = 0                      # line_number
    while ! eof(f)                # read till end of file
        s = readline(f)           # read a new line for every iteration
        println("$s")             # print the line
    end
end
```

obtaining

Brand	Price	Year
Honda Civic	22000	2015
Toyota Corolla	25000	2013
Ford Focus	27000	2018
Audi A4	35000	2018

Count the lines and print the line number

```
open("cars.txt") do f
    line = 0                      # line_number
    while ! eof(f)                # read till end of file

        s = readline(f)           # read a new line for every iteration
        line = line + 1
        println("$line . $s")
    end
end
```

giving

1 .	Brand	Price	Year
2 .	Honda Civic	22000	2015
3 .	Toyota Corolla	25000	2013

(continues on next page)



(continued from previous page)

```
4 . Ford Focus      27000  2018
5 . Audi A4        35000  2018
```

**Reading all the lines of a file into a String array using readlines()**

```
f = open("cars.txt", "r")      # opening a file in read_mode "r"
line_count = 0                 # to count total lines in the file
for lines in readlines(f)
    global line_count = line_count + 1    # Define the line_count variable global and
    increment it
    println(lines)                    # print the line
end
println("line count is $line_count")    # total lines in file
close(f)
```

obtaining

```
Brand      Price  Year
Honda Civic 22000  2015
Toyota Corolla 25000 2013
Ford Focus  27000  2018
Audi A4     35000  2018
line count is 5
```

**Read all contents of a file into a String at once using read()**

```
f = open("cars.txt", "r")      # opening a file in read mode "r"
s = read(f, String)           # read entire file into a string
print(s)
close(f)
```

obtaining

```
Brand      Price  Year
Honda Civic 22000  2015
Toyota Corolla 25000 2013
Ford Focus  27000  2018
Audi A4     35000  2018
```

**1.4.4 Writing into a file**

```
output=open("output_file.txt","w")    # opening a file in write mode "w"
write(output, "BMW 40000 2021\n")     # write some data into the file
close(output)
```

Check the presence of the file in the disk by changing to the *shell* environment by

```
julia>;
```

obtaining

```
shell>
```

Now just use *ls* to list the files in the directory and *cat output\_file.txt* to see the file contents

```
shell> ls
```

giving

```
DataFrames-RDatasets.pages
DataFrames-RDatasets.pdf
Dataframes.pages
Input_Output.pages
Input_Output.pdf
Julia_introduction.ipynb
Session 1 - Introduction Solved.ipynb
Session 2 - Files and Data.ipynb
Session-2
Session-3
cars.txt
output_file.txt
```

and

```
shell> cat output_file.txt
```

to give

```
BMW 40000 2021
```

**Warning:** Accessing the shell commands is a little bit different in Julia's command line interface REPL (read-eval-print-loop), that can be access through *Terminal* in MacOS and Linux or *PowerShell* in Windows, and in Jupyter notebook.

In REPL we only need to type ; in front of the julia> prompt

```
julia> ;
```

then appears the shell prompt

```
shell>
```

Now one can type the shell commands as wish.

In Jupyter notebook one need to type ; followed with the shell command in the same line, say list the folder contents using *ls*:

```
; ls
```