

# Introduction to Ruby

<https://www.tutorialspoint.com/ruby/index.htm>

## Installation

### Linux

#### Ubuntu or Debian

```
$ sudo apt install ruby-full
```

#### CentOS, Fedora or RedHat

```
$ sudo yum install ruby
```

#### Other Linux Distributions

Check this website for instructions:

<https://www.ruby-lang.org/en/documentation/installation/#package-management-systems>

To verify, open a terminal and type

```
$ ruby -v
```

The Ruby version should be printed out.

### Windows

1- Go to <https://rubyinstaller.org/downloads> and download “[Ruby+Devkit 2.6.4-1 \(x64\)](#)”. If not available, go to <https://aveen.se/pinkprogramming/> and download “rubyinstaller-devkit-2.6.4-1-x64.exe”

2- Double click on the file and start the installation process. When presented with the option, make sure to choose the option containing the words “development toolchain”.

3- To verify, type the following in the command line interface (cmd). The Ruby version number should be printed out.

```
$ ruby -v
```

### Mac OS

Ruby comes pre-installed. To verify, open a terminal (Locate the terminal program in the Utilities folder) and run

```
$ ruby --version
```

The Ruby version number should be printed out.

# Ruby Basic

Every language has its rules of how to build sentences and how certain words behave when they are followed by other certain words. We call these rules a language's grammars. The same thing applies for programming languages. Each programming language has its own grammars, typically called syntax. While most programming languages have the same vocabulary (most of them have words like `if`, `for`, `while`, `break`, `continue`.. etc), every language has its own rules about how these words are placed in a sentence/a line of code and how they will behave if they were followed by certain other words.

Today, we will be learning about the syntax/grammars of the Ruby programming language. To take look at how Ruby words behave, we'll start by using the interactive interface. Start a terminal (on Windows, start the "cmd" program) and type the command

```
$ irb
```

Now we can write a command in that prompt and see the result immediately. So try the following commands and see what you get

```
$ puts "hello world"
$ a = 5
$ a + 3
$ puts a.to_s << " seconds"
$ a = [5, 7, 12]
$ a*2
$ puts a
$ b = a*2
$ puts a
$ puts b
$ c = (1..10).step(1).to_a
$ puts c
```

We can do more with each element in a list when we loop through it using a for-loop

```
$ a.each do |x|
$ puts x*2
$ end
```

Notice the difference in the output compared `a*2` above

Close the interactive window

To write something more elaborate, we can write the commands in a file and execute the file. Use any file editor you like, for example `gedit` on Linux or `Notepad` on Windows. The file has to have the extension `.rb`

1- Open a file and write the following

```
puts "hello world"
```

2- Save the file as "example.rb"

3- Open a terminal. Navigate to the directory where "example.rb" is and run the following

```
$ ruby example.rb
```

# Exercises

## Exercise 1:

Loop through a list of 1 to 10 and print each number followed by the word “seconds”

## Exercise 2:

List = [34, 4, 56, 11, 9, 46, 77]

Do the following:

- Print every element of the list.
- Print the number of element in the list
- Print every element in the list that is less than 10
- Print every element in the list that is more than 10 and less than 50

### Helpful phrases to google

“ruby if statement”

“ruby get list size”

## Exercise 3:

Write a program that picks a random number between 1 and 100. Then you make a guess and the program prints out the following depending on your guess

- The guess is correct => “Congratulations. You guessed the right number!”
- The guess is higher than the number => “Too high. Pick a lower number”
- The guess is lower than the number => “Too low. Pick a higher number”

Repeat this until you guess the right number

### Helpful phrases to google

“ruby generate random number”

“ruby read input from console”

“ruby while statement”

# Advanced

Install `exifr` gem

## Linux

First, install `rubygems`

1- Go to <https://rubygems.org/pages/download> and download the zip file. If not available, go to “<https://aveen.se/pinkprogramming/>” and download “`rubygems-2.6.13.zip`”. It is a slightly older version, but it works the same way

2- Unpack into a directory and navigate to that directory

3- Run

```
$ ruby setup.rb
```

Then install the `exifr` library by running:

```
$ sudo gem install exifr
```

## Windows

The gems should be installed with the Windows installer, but we still need to install the `exifr` library by running

```
$ gem install exifr
```

## Exercise

Download any image from the following website and store it in the same directory as your ruby code files

[https://aveen.se/figureskating\\_em\\_2015/](https://aveen.se/figureskating_em_2015/)

Write a ruby program that reads the `exif` data of the image, extracts the date from the `exif` data and then renames the image file so that it contains the date. For example, if the image file name is “`DSC_0209.jpg`” and the date in the `exif` data is October 15<sup>th</sup> 2015, then the new image file name should be “`DSC_0209-2015-10-15.jpg`”

### Helpful phrases to google

“ruby read command line arguments”

“ruby exifr read image date”

“ruby file rename”

“ruby string format”