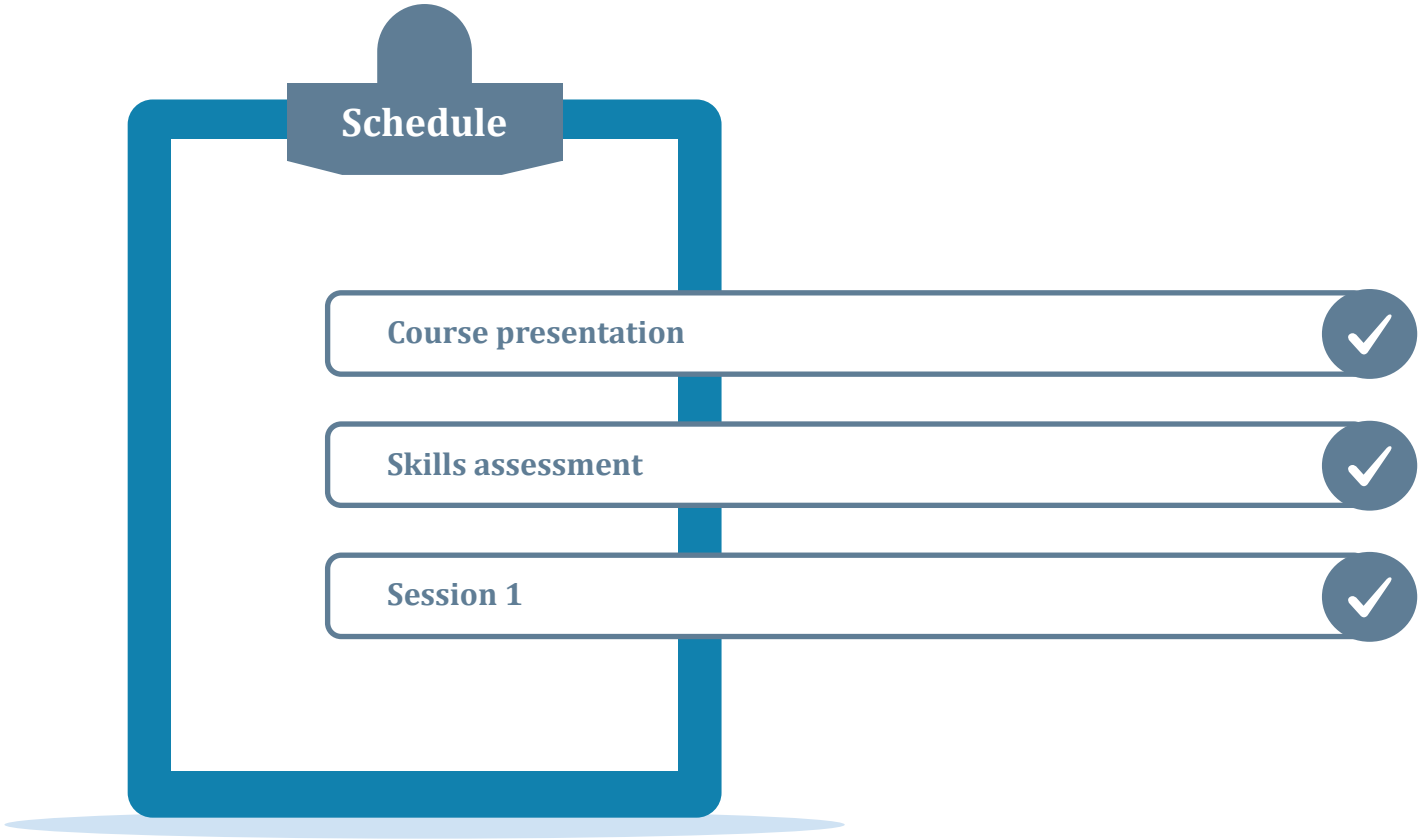
Fundamentals of Programming and Scripting

Msc Ing. Estuardo Zapeta

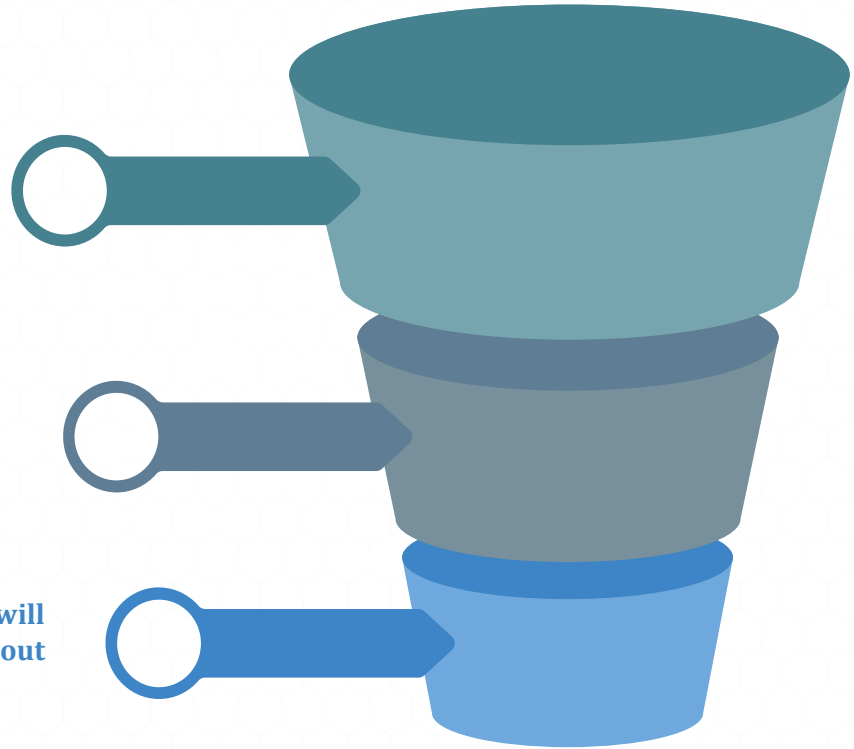

Schedule
Course presentation
Skills assessment
Session 1

Recommendations

Write by chat or enable the microphone to ask questions

Stay in session if there is a loss of connection

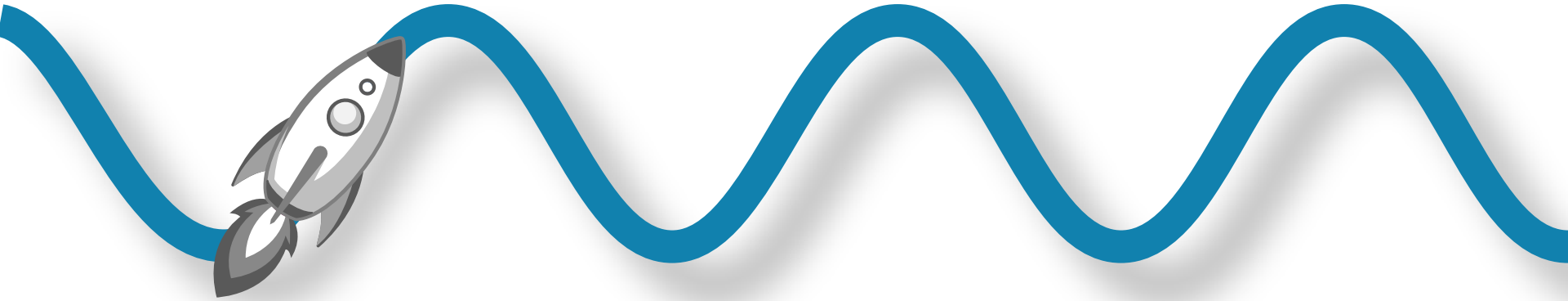
During the sessions, practical activities will be carried out



Email

estuardo.zapeta@gmail.com

Where do we start out?



Intro to basics

In its most basic form, R can be used as a simple calculator. Consider the following arithmetic operators:

Addition: +
Subtraction: -
Multiplication: *
Division: /
Exponentiation: ^
Modulo: %%

Variable assignment

A variable allows you to store a value or an object in R. You can then later use this variable's name to easily access the value or the object that is stored within this variable.

You can assign a value 10 to a variable my_var with the command:

```
my_var <- 10
```

Data types in R

R works with numerous data types. Some of the most basic types to get started are:

- Decimal values like `4.5` are called numerics.
- Whole numbers like `4` are called integers. Integers are also numerics.
- Boolean values (TRUE or `FALSE`) are called logical.
- Text (or string) values are called characters.
- Note how the quotation marks in the editor indicate that `"some text"` is a string.

Break

15 minutes

Vectors

Vectors are one-dimension arrays that can hold numeric data, character data, or logical data. In other words, a vector is a simple tool to store data.

In R, you create a vector with the combine function c(). You place the vector elements separated by a comma between the parentheses. For example:

```
numeric_vector <- c(1, 2, 3)
character_vector <- c("a", "b", "c")
```

Exercise

- Create a vector `my_numeric_vector` with values ranging from `1` to `12`
- Create a vector `my_character_vector` with 5 names of your classmates

Naming a vector

Vectors are one-dimension arrays that can hold numeric data, character data, or You can give a name to the elements of a vector with the `names()` function. Have a look at this example:

```
some_vector <- c("John Doe", "poker player")
names(some_vector) <- c("Name", "Profession")
```

This code first creates a vector `some_vector` and then gives the two elements a name. The first element is assigned the name `Name`, while the second element is labeled `Profession`.

Calculating

To select elements of a vector (and later matrices, data frames, …), you can use square brackets. Between the square brackets, you indicate what elements to select.

```
c(1, 2, 3) + c(4, 5, 6)
c(1 + 4, 2 + 5, 3 + 6)
c(5, 7, 9)
```

You can also do the calculations with variables that represent vectors:

```
a <- c(1, 2, 3)
b <- c(4, 5, 6)
c <- a + b
```

Vector selection

To select elements of a vector (and later matrices, data frames, …), you can use square brackets. Between the square brackets, you indicate what elements to select.

For example, to select the first element of the vector, you type poker_vector[1]. To select the second element of the vector, you type poker_vector[2], etc. Notice that the first element in a vector has index 1, not 0 as in many other programming languages.

Selection by comparison

The (logical) comparison operators known to R are:

< for less than
> for greater than
<= for less than or equal to
>= for greater than or equal to
== for equal to each other
!= not equal to each other

Exercise

- Create two vectors `x` and `y` with five random values each.
- Compare both vectors with logical operators (`>`, `<`, `==`).
- Multiply both vectors and store in `result` variable.

Matrices

In R, a matrix is a collection of elements of the same data type (numeric, character, or logical) arranged into a fixed number of rows and columns. Since you are only working with rows and columns, a matrix is called two-dimensional.

You can construct a matrix in R with the `matrix()` function. Consider the following example:

```
matrix(1:9, byrow = TRUE, nrow = 3)
```

Naming a matrix

Similar to vectors, you can add names for the rows and the columns of a matrix.

```
rownames(my_matrix) <- row_names_vector
colnames(my_matrix) <- col_names_vector
```