**Ruby Quest02**

Remember to git add && git commit && git push each exercise!

We will execute your function with our test(s), please DO NOT PROVIDE ANY TEST(S) in your file

For each exercise, you will have to create a folder and in this folder, you will have additional files that contain your work. Folder names are provided at the beginning of each exercise under submit directory and specific file names for each exercise are also provided at the beginning of each exercise under submit file(s).

**My First If Multiple Conditions**

* Submit directory: ex00
* Submit file: ["my\_first\_if\_multiple\_conditions.rb"]

if statement is linked to else and writing the right condition can be quite complicated :D.

Replace/Complete the following code. (XX is what you need to replace)

**Function prototype** (ruby)

a = 10

b = 9

c = 11

d = 10

y = 9

z = 11

if (XX)

puts("a is bigger than b AND smaller than c AND equal to d")

end

if (XX)

puts("z OR y are bigger than a")

end

**Example 00**

Input:

Output: a is bigger than b AND smaller than c AND equal to d

z OR y are bigger than a

Return Value: nil

**My First Function**

* Submit directory: ex01
* Submit file: ["my\_first\_function.rb"]

Writing syntax of code is the small visible part of the Code Iceberg. Software Architecture (Design Software) are really the deal.

In order to "organize" your code function are the key. Let's dive in!

Replace/Complete the following code. (XX is what you need to replace)

**Function prototype** (ruby)

## Following XXXXXX will be a function that will print using puts("my\_first\_function")

XXXXXX

XXXXXX

XXXXXX

my\_first\_function()

**Example 00**

Input:

Output: my\_first\_function

Return Value: nil

**My First While**

* Submit directory: ex02
* Submit file: ["my\_first\_while.rb"]

Repeating is annoying? what is we could create a program for it? :)

Implemente a while loop to print 100 times "I want to code". (Don't forget to increment the index ;-))

Replace/Complete the following code. (XX is what you need to replace)

**Function prototype** (ruby)

index = 0

while (XX)

puts("I want to code")

XX

end

**Example 00**

Input:

Output: I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

I want to code

Return Value: nil

**My First Param Function**

* Submit directory: ex03
* Submit file: ["my\_first\_param\_function.rb"]

Function accepts parameters, let's send an integer to our function and print it!

Implemente a while loop to call a function detonation in...X secondes. Your loop will stop a 0. 10 included, 0 is not. (Don't forget to decrement the index ;-))

Replace/Complete the following code. (XX is what you need to replace)

**Function prototype** (ruby)

// function will puts("detonation in... #{seconds\_left} secondes.")

timer = 10

while (XX)

detonation\_in(timer)

XX

end

**Example 00**

Input:

Output: detonation in... 10 secondes.

detonation in... 9 secondes.

detonation in... 8 secondes.

detonation in... 7 secondes.

detonation in... 6 secondes.

detonation in... 5 secondes.

detonation in... 4 secondes.

detonation in... 3 secondes.

detonation in... 2 secondes.

detonation in... 1 secondes.

Return Value: nil

**My First Return Function**

* Submit directory: ex04
* Submit file: ["my\_first\_return\_function.rb"]

Function returns a value, let's print it!

Implemente a function which return a number (7)

Replace/Complete the following code. (XX is what you need to replace)

**Function prototype** (ruby)

# function my\_get\_seven() will return 7

puts(my\_get\_seven())

**Example 00**

Input:

Output: 7

Return Value: nil