# Programming fundamentals
## >lab classes

Fernando Manuel Soares
fernando.soares@ulusofona.pt

# Lab Classes Structure

**3 lab classes:**

1. 05/11
2. 03/12
3. 07/01

**How are classes structured?**

- Clarification and help on previous themes,
- Exercices,
- *“codealongs”*

# Lab Classes Structure

1. Continuous evaluation
2. Clarifications regarding the following subjects
    a. *While*
    b. *For*
    c. *Arrays* and *multidimensional Arrays*
    d. *Functions*
3. *Cheatsheets*
4. *Exercices*

# Avaliation
## Of the lab classes

**The continuous evaluation (20% of the final grade)** one through several exercises Hosted on the online platform - ***repl.it***

**Number of Exercices- 36 divided in 2 sets:**

1. **First set: 24 exercices**
2. **Second set: 12 exercices**

**Exercise Deadline:**

**10/12/2020** (first set of 24 exercises)
**09/01/2020** (second set of 12 exercises)

# Repl.it – Continuous avaliation

**The continuous evaluation (20% of the final grade)** one through several exercises Hosted on the online platform - ***repl.it***

Step-by-step guide to ingress in the repl.it classroom:

1. Acess: https://repl.it/classroom/invite/yrWB3DZ
2. Make the registration.
3. Create the user name.
4. Create your profile. - **Attention: use your real names, if I can't identify you, you run the risk of not being evaluated!**
5. Acess the classroom: “FundamentosPrograma̧ão_2020_2021_ULHT”

# Conditional cycles: While

```
while <boolean expression> :
    <instructions block>
```

***while*** cycles keep executing if the <boolean expression> is ***true***

# Conditional cycles: While

***while*** cycles keep executing if the <boolean expression> is ***true***

**Table with values that evaluate as true or false:**

https://bit.ly/2PbH8PS

# Conditional cycles: While

## Example :

```
user_input = None

while user_input != "Yes" :
    user_input = input("Exit the loop? ")

print("Goodbye")
```

# Conditional cycles: While

**Mini-exercise 1 :**

Based on the previous example, write a program in which the terminating condition is any configuration of: “Yes” (uppercase, lowercase, etc)

***Ex: Yes, yEs, ...***

# Conditional cycles: While

**Mini-exercise 2 :**

Write a program using **while** that prints the following pattern:

```
#
##
###
```

# Conditional cycles: While

**Mini-exercise 3 :**

Create a program that prints this pattern using **while:**

```
   #
  ##
###
```

# Conditional cycles: for

```
for <item> in <iterable_object> :
    <instruction block>
```

**for** cycles repeat the <instruction block> for each item contained in a <iterable_object> (lists, sets, strings,…)

# Conditional cycles: for

**Mini-exercise 4 :**

Make a program that print all numbers from 1 to 50.

# Conditional cycles: for

**Mini-exercise 5 :**

Make a program that prints all even numbers from 1 to 50.

# Conditional cycles: for

**Mini-exercise 6 :**

Make a program that print this pattern using the **for** loop:
#
##
###

# Arrays - Python Lists

Data structure consisting of a collection of elements

```
l = [89, 40.2, True]
l[2] = True

l [-1] = ?
```

# Arrays

**Mini-exercise 7:**

Order the following list:

```
l = [7,4,1,7,2,3,5,1]
```

So that it is like this:

```
l = [1,1,2,3,4,5,7,7]
```

# Arrays

**Mini-exercise 8:**

Now remove the duplicates:

```
l = [7,4,1,7,2,3,5,1]
```

Should look like this:

```
l = [7,4,1,2,3,5]
```

# Arrays

**Mini-exercise 9:**

Now delete even numbers, order and remove repeated elements:

```
l = [7,4,1,7,2,3,5,1]
```

So that it is like this:

```
l = [1,3,5,7]
```

# Multidimensional Arrays

Data structure consisting of a collection with one or more collections of elements

```
l = [[2,42,1], 40.2, True]

l [0][2] = 1

l = [[2,42,1], 40.2, (30,2.0, False ,[2,4,"My brain hurts"],
"Hello")]

l[2][3][2][5] = ?
```

# Multidimensional Arrays

## Mini-exercise 10:

Consider the following list:

```
num = [ [10], [1,2,3], [5,6], [0,2,3] ]
```

Create a program that returns another list, **multi**, that contains the multiplication of the numbers contained in each of the lists of **num.**

The program should return:

```
[10, 6, 30, 0]
```

# Multidimensional Arrays

## Mini-exercise 11:

Consider the following list:

```
Result = [[ Figo, João, Marta], [Figo, João, Marta], [Marta, João, Figo]]
```

in which the order of the names in each sub-list indicates the order of victory of a tournament. Each sublist indicates a tournament stage. Saying that, in the first stage Figo came first, João in second and Marta in third, and so on.

Create a program that indicates who achieved first place more often.

```
> <winner> Wins!
```

# Functions

```
def <name>(<parameters>) :
    <code block>
```

**Functions group blocks of code, making these blocks reusable.**

"They arrange" the code in a program, making it more organized and easier to understand, especially in larger programs.

# Functions

**Mini-exercise 12:**

Writes a function called **ten** that returns **True** if its argument is 10 and **False** otherwise. **You cannot use the if statement**.

# Functions

**Mini-exercise 13:**

Define a function that receives a list as a parameter and returns the last element of that list. If the list is empty, returns the **None** keyword

```
print(example([1,3,5]))

> 5
```

```
print(example())

> None
```

# Functions

## Mini-exercise 14:

Define a function that receives a parameter: a number **(int and float)** or **a string**:

**The function returns different things following these rules:**

1. If it is a number returns that number as a string.
2. If it is a string returns the count of characters as an int.
3. If the argument is not string, int, or float. Returns the message: "Sorry, that's not a float or a string"

**Tip: Use python's base function: "isinstance()"**

# Exercise – Rock, Paper, Scissors

Create a program in which a player plays against the computer the classic game: rock, paper or scissors.

Rules:

The program asks for the player input:

1. only accepts **"rock"**, **"paper"** or **"scissors"** or **"exit"**
2. If the input is "**exit**" the program should exit and display the message: "Thanks for playing".
3. In any other type of input. You should print the message “command not valid” and the program must restart

1. The computer response is random (see: **random.randint**).
2. The game should indicate the play of the computer and who is the winner.
3. After being declared the winner, or if the command is not valid, the program must restart.
4.

# Exercise - Resolution

# For those who want to go “an extra mile" – Exercises

Gamified python teaching platform with over 200 exercises

https://py.checkio.org/

# Cheatsheets

https://bit.ly/2N1PQ0m