# Object Oriented Programming with Python

Gramsci Hermozo

Session 02

# Content

- Strings
- Lists
- Tuple
- Dictionaries
- Functions

# String

## Ex-1.Concatenate

```
name = "Gramsci"
lastname = "Hermozo"

# concatenate your name and lastname and print
full_name = ""
```

**Output:**

```
Gramsci Hermozo
```

# String

## Most common functions

**format()**

```
"{}".format(var)
"{0} {1}".format(var, var1)
f"{<var_name>}"
```

# String

## Most common functions

**count()**

```
txt = """
      This example is for python course
      for Jalasoft engineers
      """
# returns a number of times a specified
# value appears in the string
x = txt.count("for")
```

**output:** 2

# String

## Most common functions

**replace()**

```
txt = "This is a cpp course"
new_txt = txt.replace("cpp", "python")
print(new_txt)
```

**output:**

```
This is a python course
```

# String

## Most common functions

**strip()**

```
txt = "    Hello World     "
new_txt = txt.strip()
print(new_txt)
```

**output:**

```
Hello World
```

## Ex-2. Count

As a crazy politician, I would like to know, how many times is mentioned my Name into this post. My name is Adolf

**Post**

Adolf not was the only one in his politic, there was other tree adolf's aDolf Junior, adolF, middle and the big ADOLF.

# List

## Create a List

```
my_list = ["Python", "is", "so", "cool"]
print(my_list)
```

# List

## Most common functions

**len()** This function returns the size of a list

```
my_list = ["C", "Pascal", "Javascript"]
print(len(my_list))
```

**output:**

```
3
```

# List

## Most common function

**Access List Items**

```
my_list = ["cpp", "python", "c#"]
print(my_list[1])
```

**output:**

```
banana
```

# List

## Most common functions

**insert(index, value)**

```
my_list = ["This", "a", "list"]
my_list.insert(1, "is")
print(my_list)
```

**output:**

```
["This", "is", "a", "list"]
```

# List

## Most common functions

**append(value)** This funciton add an item into the end of the list.

```
my_list = ["c#", "java", "c#"]
my_list.append("python")
print(my_list)
```

**output:**

```
["c#", "java", "c#", python]
```

# List

## Most common functions

**remove(value)**

```
list = ["dog", "cat", "mouse", "lion"]
list.remove("mouse")
```

**pop()**

```
list = ["dog", "cat", "mouse", "lion"]
list.pop(2)
```

**del**

```
list = ["dog", "cat", "mouse", "lion"]
del list[2]
```

**clear**

```
list = ["dog", "cat", "mouse", "lion"]
list.clear()
```

## Ex-3. Concatenate lists (who I am)

I had a transit accident and I don't remember my name help me please!!

```
memory_1 = ["M", "na", "i", "Jo", "Bla"]
memory_2 = ["y", "me", "s", "e", "ck"]
```

**Expected Result**

```
['My', 'Name', "is", 'Joe', 'Black']
```

## Ex-4. Need numbers

A crazy trainer returns my grade into single string and I need to know the total and the average

```
"English=68 Logic=75 Uml=87 Code=80"
```

## Ex-4. Replace value

I would like to change my firts option that was 20 for 2000

```
options = [5, 10, 15, 20, 25, 30, 4, 20]
```

**Expected Result**

```
[5, 10, 15, 2000, 25, 30, 4, 20]
```

# Tuple

**Create a Tuple**

```
my_tuple = ("this", "is", "a", "tuple")
print(my_tuple)
```

**output:**

```
("this", "is", "a", "tuple")
```

The supported functions by Tuple class are almost the same than a List class

# Dictionary

**Create a Dictionary**

```
my_dict = { "type": "Car", "brand": "Ford",
           "model": "Mustang", "year": 2020 }
print(my_dict)
```

**output:**

```
{"type": "Car", "brand": "Ford",
"model": "Mustang", "year": 2020}
```

# Dictionary

## Accessing Items(1/2)

**Brackets**

```
my_dict = { "type": "Car", "brand": "Ford",
            "model": "Mustang", "year": 2020 }
model = my_dict["model"]
```

**Get function**

```
my_dict = { "type": "Car", "brand": "Ford",
            "model": "Mustang", "year": 2020 }
model = my_dict.get("model")
```

**Get Keys**

```
my_dict = { "type": "Car", "brand": "Ford",
            "model": "Mustang", "year": 2020 }
keys = my_dict.keys()
```

## Accessing Items(2/2)

**Get Values**

```
my_dict = { "type": "Car", "brand": "Ford",
            "model": "Mustang", "year": 2020 }
values = my_dict.values()
```

**Get Items**

```
my_dict = { "type": "Car", "brand": "Ford",
            "model": "Mustang", "year": 2020 }
items = my_dict.items()
```

# Dictionary

## How to change/add Values

**Brackets**

```
my_dict = { "type": "Car", "brand": "Ford",
           "model": "Mustang", "year": 2020 }
my_dict["year"] = "1988"
```

**Update function**

```
my_dict = { "type": "Car", "brand": "Ford",
           "model": "Mustang", "year": 2020 }
my_dict.update({"brand": "Toyoda"})
```

# Dictionary

## Remove Items

**Pop function**

```
my_dict = { "type": "Car", "brand": "Ford",
            "model": "Mustang", "year": 2020 }
my_dict.pop("type")
```

**Del function**

```
my_dict = { "type": "Car", "brand": "Ford",
            "model": "Mustang", "year": 2020 }
del my_dict["type"]
del my_dict # WARNING: if you no specify the key to
              # remove that could delete the dictionary
              # completely and cause an error
```

# Functions

**Definition**

```
def function_name(parameters):
  statement(s)
```