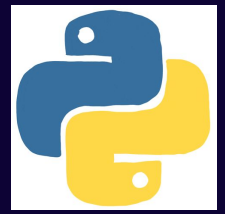
python™

CSULB-COB-SUMMER 2022

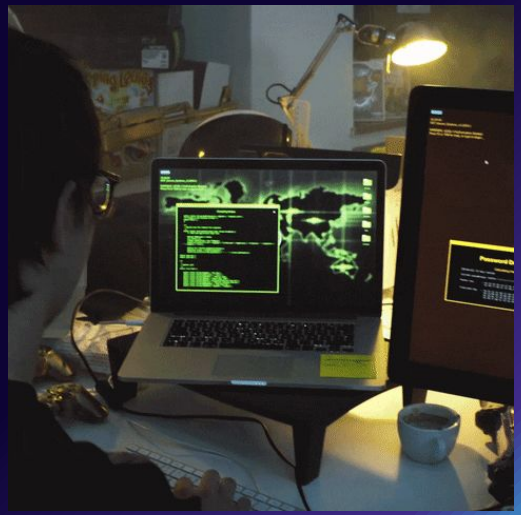
WHAT IS PYTHON?

- Is it a snake?
- Is it a food?
- Or is it good?

- **PYTHON** IS NOT A SNAKE. WELL, KINDA.



- **PYTHON** is a high-level programming language



- Help you tell your computer what to do anything in a **cooler way**.

1. What can PyThon do?

Some of my works/ projects?

Stock Prediction

V.R. Game

Making Robot/ A.I.

Most Important Usages

- Data Analysis (Mathplotlib, Tensor Flow, Pandas). 
- Web Development (Flask, Streamlit, Django).
- A.I./ Machine Learning. 

2. Why you need to learn PyThon ?

Boring Answers

- For Your class, to get your degree
- You just want to pass IS 340
- Because the school requires you to learn

Good Answers

- You want to have an attractive salary 😉
- You want to impress employers 👍
- You just want to learn tech 📖

3. Your first PyThon Program?

Install PyThon

- Please go to Google, type “Python 3 download” , and press enter
- For **Window:**
  - Just click the “Download Python 3.10.5”
- For **Mac:**
  - Look a little down and click “macOS”
  - Then look for “Stable Releases” click the link after the first Download keyword
- After Download, run the installation

Starting PyThon

- After the installation,
- For **Window:**
  - Search “IDLE” and enter to open Python development environment
- For **Mac:**
  - Go to “Launchpad” and and search for “IDLE” then enter
- You will see something like a terminal open up

“Hello World” PyThon

- **Congratulation**, you are now officially a programmer.

- The first think you need to do is to say “Hello” to the new world.
- Please type print(“Hello World”)
- And enjoy the magic of 21st century.

4. Main Things about PyThon

Data Types PyThon

- **Text Type:** str
- **Numeric Types:** int, float, complex
- **Sequence Types:** list, tuple, range
- **Mapping Type:** dict
- **Set Types:** set, frozenset
- **Boolean Type:** bool
- **Binary Types:** bytes, bytearray, memoryview
- **None Type:** None
- **Python** has a built-in data type detection. So we do not need to include variable's type when declare a variable.

- The most important thing to remember is to declare variables.
- Before we do anything, we need to declare variables to store data or result of a calculation.
- Ex:

```
x = 5
y = 4
z = x + y
print(z) #The result will be 9
```

Math PyThon

- **Python** has most of the mathematical operations that we can think of, even some complex modules.
- Normally, we do not need to import math module to use its functions, for more information: https://www.w3schools.com/python/module_math.asp
- Ex: Let write (5^2-3)÷11 + 2

```
x = ((5 ** 2) - 3 ) / 11 + 2
print(x)
```

- Some basic mathematical operations syntax:
  - **Addition :** x + y
  - **Subtraction:** x - y
  - **Multiplication:** x * y
  - **Division:** x / y
  - **To the power:** x ** y
  - **Minimum of two number:** min(x,y)
  - **Maximum of two number:** max(x,y)
  - **Greater than (or equal) comparison:** x > y , x >= y
  - **Less than (or equal) comparison:** x < y, x <= y
  - **Equal comparison:** x == y
  - **Sum of all number in a list:** sum([1,2,3,4])

If/elif/else PyThon

- **Python** supports the usual logical condition from mathematics:
- These conditions can be used in several ways, most commonly in "if, elif, else" statement
  - An "if statement" is written by using the if keyword.
    - ```
      if ( my_grade >= 90):
          print ("I get an A")
      ```
  - An elif keyword is **Python** way of saying "if the previous conditions were not true, then try this condition"
    - ```
      if (my_grade >= 90):
          print("I get an A")
      elif (my_grade >= 80 and my_grade < 90):
          print("I get a B")
      ```

- Finally, the else keyword catches anything which isn't caught by the preceding conditions:
  - if (my_grade >= 90):
        print("I get an A")
    elif (my_grade >= 80 and my_grade < 90):
        print("I get a B")
    elif (my_grade >= 70 and my_grade < 80):
        print("I get a C")
    elif(my_grade >= 60 and my_grade < 70):
        print("I get a D")
    else:
        print("I do not pass")

Loop PyThon

- There are two kinds of loop in **Python**: For loop and while loop

- **While loop** is used to execute a block of code repeatedly until a given condition is not satisfied:
  - ```
    count = 0
    while(count < 3):
        count = count + 1
        print(count)
    ```
  - Notice: We need to update while loop condition manually.
    ```

- **For loops** are used for going through all element in a list, set, dictionary or array.
  - for count in range(3): # similar to previous page while loop
    print(count)

  - Numlist = [1,2,3,4]
    for count in Numlist: # traversing a list
    print(count)

  - Dictionary = {“saber”: “Atoria”, “caster”: “Castoria”}
    for name in Dictionary: # traversing a dictionary
    print(name)

User Input PyThon

- Many times, we need to let user control the program. Hence, we need a way to take some inputs from them.
- **Python** provides us a built-in input() function.
- This function takes the input from user and convert it to a string by default.
- We can change the type of input by:
  - Integer input: int(input())
  - Float input: float(intput())

- Syntax:

```
x = input(“Please enter your name:”)
print(“Hello ”, x)
```

Or

```
print(“Please enter your name:”)
x = input()
print(“Hello ”, x)
```

Other examples:

```
Name = input(“Please enter your name:”) #string input
Age = int(input(“Please enter your age”)) #integer input
print(Name,“ are ”,Age,” year old”)
```

“File read/write” PyThon

- Python provides built-in functions for reading, writing and creating files.
- To read a file:
  - File = open(“file_path”, “r”)

```
# read all lines from the file and store as a big string
String = File.readlines()
print(String)
```

Or

```
# read line by line of the file
for line in File:
    print(line)# do something with each line
File.close()
```

- To erase data from a file and write to it (or create one if that file has not existed):
  - File = open(“file_path”, “w”)
    File.write(“write anything you want”)
- To write data to the end of the file:
  - File = open(“file_path”, “a”)
    File.write(“write anything you want”)

Notice: We need to close the file after finish processing it

File.close()

Function PyThon

- A function is a block of code which only runs when it is called
- We can pass data, known as parameters, into function
- **Python** detects if code is inside a function by **indentation**
- Syntax:

```
def function_name(parameter1,parameter2,etc):
    #code of the function
```

- Ex1:

```
def firstfunction ():
    print("Welcome to the Beach")
firstfunction() #calling the function
# this is how we call a function (its name follow by any parameter
you want to pass inside the "()"
```

- Ex2:

```
def addfunction (num1, num2):
    print(num1+num2)
    return num1+num2
#the return statement is used when we need to pass the result or computation of a function to a variable for later use.
X = 1
Y = 2
Z = addfunction(X, Y) # calling the function
# We pass X as num1 and Y as num2 parameter
# The parameter name can be anythin when you declare the function, but it needs to be exact with the variables' name that you want to pass in the function when calling it.
```

Complex example:

```
def findListAverage(numList):
    Sum = 0 # initialize the sum variable
    For number in numList:
        Sum = Sum+ number
    Average = Sum/len(numList)
    print(Average)
    Return Average
# find average by dividing sum for total number in numList
#len() function is used to find the total elements inside a list, array, dictionary
List = [1,3,53,43,5,12,-1]
listAverage = findListAverage(List) #store the average in a variable
```

THANK YOU