INTRO TO PYTHON

OUTLINE

- Workshops Outline and Goal
- Python Data Types
- Conditions
- Introduction to Lists

GOAL OF THE WORKSHOP

GOALS

- Gain basic programing skills to go beyond Programing 1
- Apply the math with Python/R
  - Learn the needed libraries : numpy, pandas, scipy, sympy, sklearn, pytorch
- Go beyond in Data Science, ML, DL

HOW DOES PYTHON WORK!

WHICH CODE

I want to ask the user for his name and print hello + user. Then, I want to ask him for his age and then print your age is + age

```
name = input(“Please Enter you name: ”)
age = input(“Please Enter you name: ”)
print(age)
print(name)
```

```
name = input(“Please Enter you name: ”)
print(name)
age = input(“Please Enter you name: ”)
print(age)
```

Interpreter is a computer program that directly executes instructions written in a programming or scripting language

EXPLAIN THAT CODE

```
x, z = 34 - 23, 3.45                # Assigning int and float
y = “Hello”                         #ASSIGNING STRING.
L = [34 - 23, “Hello”, 3.45 ]

if z ==  3.45 or “Hello” in L:
      x = x + 1
      y = y + “World”          # String concat.
print(x)

print(L)
```

LETS UNDERSTAND THAT CODE

COMMENT

- **Start comments with # – the rest of line is ignored.**
- **Can include a “documentation string” as the first line of any new function or class that you define.**

```
“““This code does blah blah blah. python won’t
 see anything in that block”””

# The code would go on this line…

print(“Ahmed”)
```

BASIC DATA TYPES

- **Integers (default for numbers)**

```
z = 5 / 2  # Answer is 2,    integer division.
```

- **Floats**

```
x = 3.456
```

- **Strings**
  - Can use “” or ‘’ to specify.
    `“abc” ‘abc’` (Same thing.)
  - Unmatched can occur within the string.
    `“matt’s”`
  - Use triple double-quotes for multi-line strings or strings than contain both ‘ and “ inside of them:
    `“““a‘b“c”””`

NOTES FOR DATA TYPE

1- Type() function

2- Difference between

print(ahmed)   print(“ahmed”)   print(‘ahmed’)

3- I want to convert

- 10 to be a str and float
- 10.01 to be a str and int
- “ahmed” to be an int and float

VARIABLES

variables are small containers that stores your any data type

Case study, I want to store the user age to print it and print when is he going to reach 100 years

```
age = 18    # we assign the number 18 to the variable age
print(age)
print(100 - age)
```

ACCESSING NON-EXISTENT VAR

- **If you try to access a name before it’s been properly created (by placing it on the left side of an assignment), you’ll get an error.**

```
>>> y

Traceback (most recent call last):
  File "<pyshell#16>", line 1, in -toplevel-   y
NameError: name ‘y' is not defined
>>> y = 3
>>>

y

3
```

MULTIPLE ASSIGNING

- **You can also assign to multiple names at the same time.**

```
>>>x, y = 2, 3
>>>x
2
>>>y
3
```

NAMING RULES

- Create variables called **int, Harvard of Africa, and 1ahmed** then store integer, string, float values in them
- We can't use python keywords as name of variables or functions
    - import keyword
    - print(keyword.kwlist)

- **There are some reserved words:**

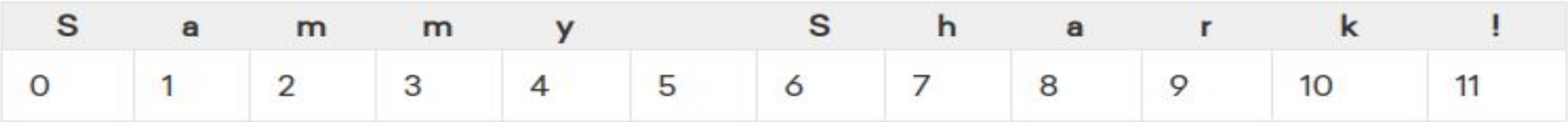
```
and, assert, break, class, continue, def, del, elif,  else, except,
exec, finally, for, from, global, if,  import, in, is, lambda, not,
or, pass, print, raise,  return, try, while
```

- **Names are case sensitive and cannot start with a number. They can contain letters, numbers, and underscores.**

```
bob Bob _bob    _2_bob_bob_2   BoB
```

STRING SLICING

- we have variable singer = “Sammy Shark!”, we want to print “Sammy is the best singer”



S	a	m	m	y		S	h	a	r	k	!
0	1	2	3	4	5	6	7	8	9	10	11

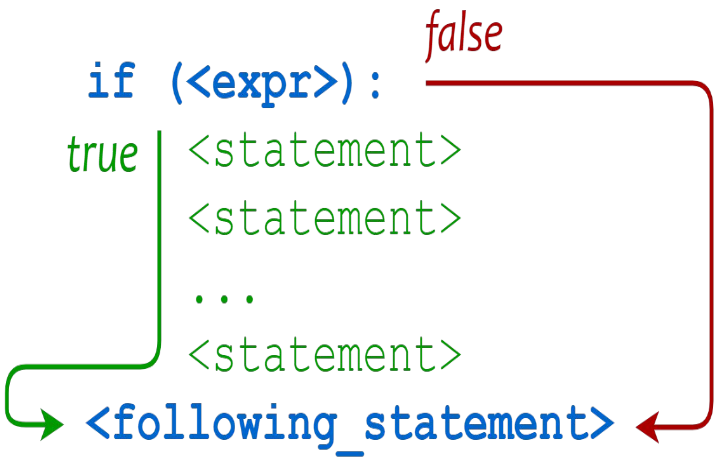
- Slice one value: variable[index]
  - singer[3]   “m”
  - singer[-2]   “k”

- Slice a part of str: variable[start: end -1 : step]
  - singer[0: 5: 1]   “Sammy”
  - singer[0:]   “Sammy Shark!”

IF STATEMENT

Problem: A variable has an int,

- if that integer equals to 10; print("the number is small").
- if it is equal to 20; print("number is medium").
- apart from that print("I don't know")



```
if (<expr>):
    <statement>
    <statement>
    ...
    <statement>
<following_statement>
```

```
if test_expression :
      statement(s)

elif test_expression:
      statement(s)

#it will include all of the other cases
else:
      statement(s)
```

Expression operations:

==

!=

> , < , >=, <=

test_expression **or** test_expression

t_expression **and** t_expression

Str **in** something(eg: list or int)

Str **not in** something

1- Whitespace is meaningful in Python: especially indentation and placement of newlines.

2- Colon after any test_statment in Python (proposition)

EXERCISE 1

Get a number from the user, if the number is even print {number} “is even”, otherwise print “it is negative”.

EXERCISE 2

Make a variable (N) and store your first name in it, then make that variable equal to another variable (FN). Then update FN with your full name. Now I want you to make an if statement such that the condition compares N with FN is true.

EXERCISE 3

Write code that takes in a sentence in mixed cases and outputs it in sentence case.

EXERCISE 2

In this challenge, the user enters a string and a substring. You have to print the number of times that the substring occurs in the given string. String traversal will take place from left to right, not from right to left.

input :

```
ABCDCDC
CDC
```

Output:

```
2
```

EXPLAIN THAT CODE

```
x, z = 34 - 23, 3.45            # Assigning int and float
y = “Hello”                     #ASSIGNING STRING.
L = [34 - 23, “Hello”, 3.45 ]

if z ==  3.45 or “Hello” in L:
    x = x + 1
    y = y + “World”       # String concat.
print(x)
print(L)
```

THANK YOU!!!

ANY QUESTIONS?