Notas del curo Python de IBM, por Carlos Enriquez con fines academicos

Nada más no manden sus ejercicios idéntico, porfa.

M O D U L O 1

----------Contenido----------

A statement or expresión is a instruction tath computer execute

Una frase o una instrucción es una instrucción que ejecuta la computadora

El statmenent o funcion Print()

Print (“Hello World”)

El valor entre los paréntesis es llamado argumento

Parta hacer comentarios ponemos # (hash simbol)

Un error de sintaxis o syntactic error es cunado pýthon no comprende el código por un error de escritura

Un error de semántica o semantic error es cuando la lógica de lo escrito esta mal

----------Jupyter lab / Google Colabs----------

#Imprimir Hola mundo

print(“Hello World”)

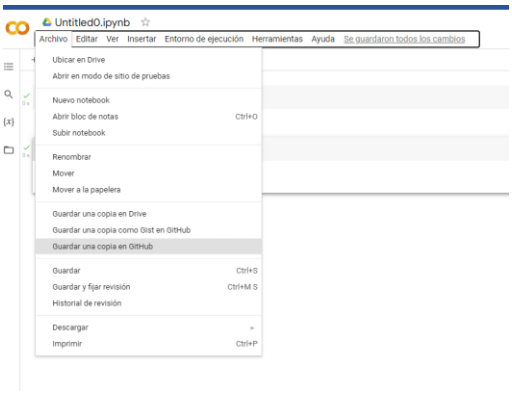
#Imprimir Hola Mundo en dos renglones diferentes

print(“Hello\nWorld”)

```
In [6]: #Imprimir Hola mundo
        print("Hello World")

        Hello World

In [5]: #Imprimir Hola Mundo en dos renglones diferentes
        print("Hello\nWorld")

        Hello
        World

In [ ]:
```

```
In [6]: #Imprimir Hola mundo
        print("Hello World")

        Hello World

In [5]: #Imprimir Hola Mundo en dos renglones diferentes
        print("Hello\nWorld")

        Hello
        World

In [ ]:
```

Nota para guardar de colab a github, tienes que seleccionar



Y después vinvular tu cuenta de colab con la de github

Si estas trabajando en jupyter, solo guarda el archivo y súbelo a tu repositorio de github

----------Quiz----------

Question 1

1/1 punto (no calificado)

Using the **Practice Lab**, find the result of executing `print("Hello\nWorld!")` statement.

- ○ Hello\nWorld!
- ○ Hello World!
- ◉ Hello
  World!

✔

Enviar Ha realizado 2 de 2 intentos

✔ Correcto (1/1 punto)

Question 2

1/1 punto (no calificado)

What is the output of executing the following statement: `# print('Hello World!')` ?

- ○ Hello World!
- ○ "Hello World!"
- ◉ There is no output as it is a comment.

✔

Guardar

Enviar Ha realizado 1 de 2 intentos

✔ Correcto (1/1 punto)

----------**Contenido**----------

Los tipos de información es la forma en que se representan diversos tipos de datos

- Enteros o Integers (35) pueden ser positivos y negativos
- Flotantes o floats (35.568) pueden ser positivos y negativos con punto decimal
- Cadenas o strings (Hola)
- Boleanos o booleans(true or false), es negativo o falso

Typecating, nos permite cambiar de un tipo a otro

Ponemos el tipo al que queremos cambiar y entre paréntesis el valor en su tipo original:

Para cambiar un entero a flotante: float(2), el resultado es 2.0

Para cambiar de floatante a entero: int(1.1), el resultado es 1, obsérvese que se pierde información

Para cambiar una cadena a entero: int (‘1’), el resultado es 1, considerar que no se puede hacer lo mismo con texto, por ejemplo, poner una letra nos devolverá un error

Para cambiar un boleano a entero: int(True), nos devolverá 1, el false nos devolverá cero, y viceversa de entero a boleano

----------Jupyter lab / Google Colabs----------

#Encontrar el tipo de la nformación

#”Hello World””1.1”

type (“Hello World”)

#Encontrar el tipo de la nformación

#1.1”

type (1.1)

#Casto o convertir el valor primero a entero int a booleano bool

#”1”

bool(int(“1”))

----------Quiz----------

Question 1

1/1 punto (no calificado)

What is the type of the following: int(1.0)?

- ○ float
- ◉ int
- ○ bool

✔

Guardar

Enviar Ha realizado 1 de 2 intentos

✔ Correcto (1/1 punto)

Question 2

1/1 punto (no calificado)

Enter the code to convert the number 1 to a Boolean.

bool(1) ✔

Guardar

Enviar Ha realizado 1 de 2 intentos

✔ Correcto (1/1 punto)

----------**Contenido**----------

Las expresiones describen un tipo de operaciones que las computadoras ejecutan

Expresiones de operaciones matemáticas

+ suma

- Substraction

* Multiplication

/ division

// división entera, con resultado redondeado

Los números son llamados operands u operadons, y los símbolos son llamados operadores u operators

La realizaci[on sigue las reglas de las convenciones matemáticas, es decir priorizando multiplicacion y division y después suma y resta. Pero primero se le da priorida a las operaciones entre parentesis

Las variables guardan valores, les ponemos el nombre deseado y asignamos valor usando el operador de asignacion el símbolo =

En las variables no se puede poner espacio, usamos guion bajo, por ejemplo: Mi_variable=10

Las variables pueden ir cambiando y contenerse a sí mismas, por ejemplo:

asignamos un valor inicial de x=10

después lo reemplazamos x=x+15, ahora x vale 25, y así sucesivamente

----------Jupyter lab / Google Colabs----------

```
#Imprimir una salida

print('Hello, Python!')

#Revisar nuestra version de python

import sys

print(sys.version)

#Escribir comentarios

print('Hello, Python!') #Esta linea imprime una cadena

#Escribir un codigo erroneo

print("Hello, Python!)

#Escribir tres cadenas, una con error, para ver el orden

print("This will be printed")

frint("This will cause an error")

print("This will NOT be printed")

#Encontrar los tipos de informaci[on

type(12)

type(2.14)

type("Hello, Python 101!")

type(-1)

type(4)

type(0)

type(1.0)

type(0.5)

type(0.56)

type(False)
```

```
type(True)
type (6/2)
type(6//2)
#Conocer la informacion del sistema sobre el tipo flotante
sys.float_info
#Convertir los valores y ver a que tipo cambiaron
float(2)
type(float(2))
int(1.1)
type(int(1.1))
int('1')
float('1.2')
str(1)
str(1.2)
int(True)
bool(1)
bool(0)
float(True)
#Convertir un valor con error
int('1 or 2 people')
#Calculos
#Cuantas horas tiene un minuto
160/60
#Guardar valor en variables e imprimirlos
x = 43 + 60 + 16 + 41
x
y = x / 60
y
total_min = 43 + 42 + 57
total_min
total_hours = total_min / 60
total_hours
```

```
x = 3 + 2 * 2

y = (3 + 2) * 2

z = x + y

z
```

```
In [2]: #Imprimir una salida
        print('Hello, Python!')

Hello, Python!
```

```
In [3]: #Revisar nuestra version de python
        import sys
        print(sys.version)

3.10.7 (tags/v3.10.7:6cc6b13, Sep  5 2022, 14:08:36) [MSC v.1933 64 bit (AMD6
4)]
```

```
In [4]: #Escribir comentarios
        print('Hello, Python!') #Esta linea imprime una cadena

Hello, Python!
```

```
In [5]: #Escribir un codigo erroneo
        print("Hello, Python!)

  Cell In [5], line 2
    print("Hello, Python!)
          ^
SyntaxError: unterminated string literal (detected at line 2)
```

In [6]:
```
#Escribir tres cadenas, una con error, para ver el orden
print("This will be printed")
frint("This will cause an error")
print("This will NOT be printed")
```

```
This will be printed

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In [6], line 3
      1 #Escribir tres cadenas, una con error, para ver el orden
      2 print("This will be printed")
----> 3 frint("This will cause an error")
      4 print("This will NOT be printed")

NameError: name 'frint' is not defined
```

In [12]:
```
#Encontrar los tipos de informaci[on
type(12)
type(2.14)
type("Hello, Python 101!")
type(-1)
type(4)
type(0)
type(1.0)
type(0.5)
type(0.56)
type(False)
type(True)
type (6/2)
type(6//2)
```

Out[12]:
```
int
```

In [13]:
```
#Conocer la informacion del sistema sobre el tipo flotante
sys.float_info
```

Out[13]:
```
sys.float_info(max=1.7976931348623157e+308, max_exp=1024, max_10_exp=308, min=
2.2250738585072014e-308, min_exp=-1021, min_10_exp=-307, dig=15, mant_dig=53, e
psilon=2.220446049250313e-16, radix=2, rounds=1)
```

In [14]:
```
#Convertir los valores y ver a que tipo cambiaron
float(2)
type(float(2))
int(1.1)
type(int(1.1))
int('1')
float('1.2')
str(1)
str(1.2)
int(True)
bool(1)
bool(0)
float(True)
```

Out[14]:
```
1.0
```

In [15]:
```
#Convertir un valor con error
int('1 or 2 people')
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In [15], line 2
      1 #Convertir un valor con error
----> 2 int('1 or 2 people')

ValueError: invalid literal for int() with base 10: '1 or 2 people'
```

In [16]:
```
#Calculos
#Cuantas horas tiene un minuto
160/60
```

Out[16]:
```
2.6666666666666665
```

In [17]:
```
#Guardar valor en variables e imprimirlos
x = 43 + 60 + 16 + 41
x
y = x / 60
y
total_min = 43 + 42 + 57
total_min
total_hours = total_min / 60
total_hours
x = 3 + 2 * 2
y = (3 + 2) * 2
z = x + y
z
```

Out[17]:
```
17
```

----------Quiz----------

No hay quiz

----------Contenido----------

Sring Mehods y String Operators

Una cadena o string, es una secuencia de caracteres, y se escriben entre comillas dobles o simples. Pueden ser letas o caracteres.

Cada letra tiene una posici[on

Si a la variable Name le damos el valor “Michael Jackson”, para obtener la letra m tenemos que acceder así>

Name[0] y esto nos devolverá la letra M, ya que es la que está en la posicion 0

También se puede hacer la indezacion inversa, es decir accediendo pero para atrás

Para acceder a varos valores,usamos slicing, o ponemos un rango desde que posición hasta que posición

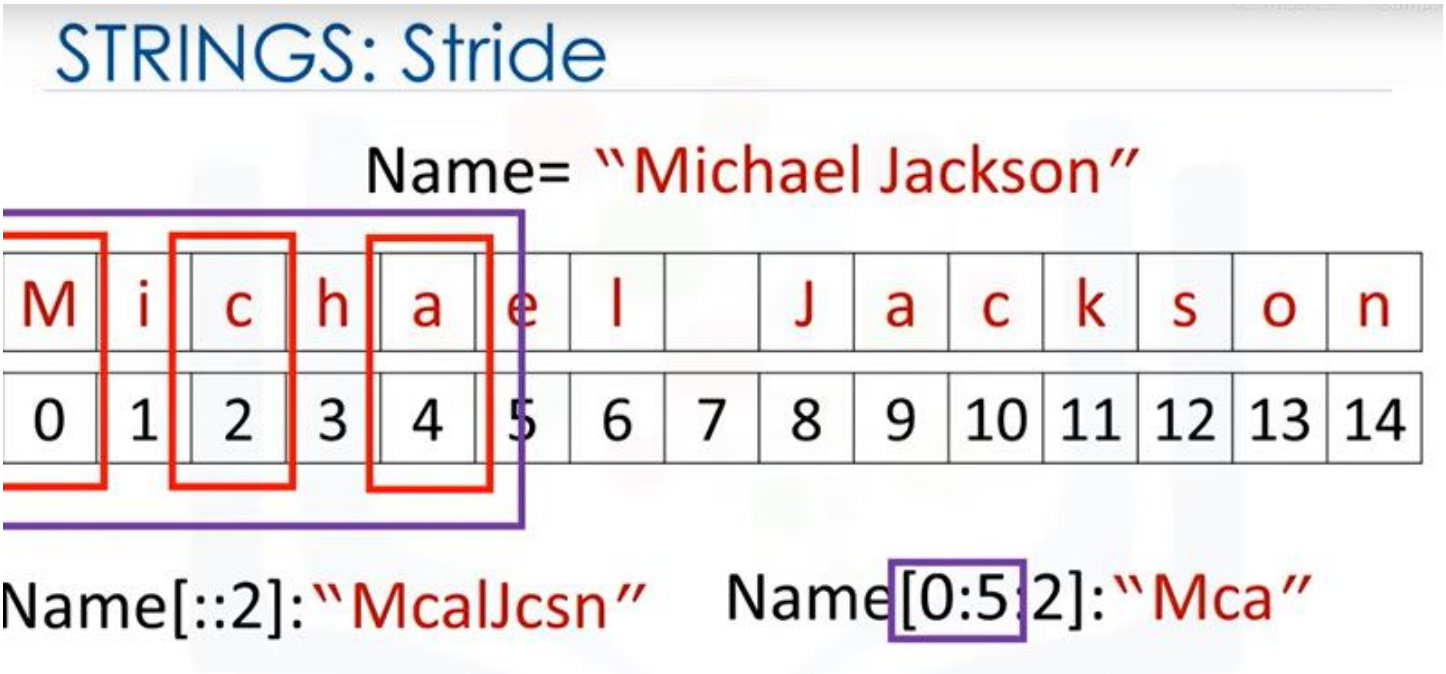
Name[0:4]

Para dar saltos, usamos el stride que indica que se darán saltos cada determinado número>

Name [::2], esto devolverá todas los valores en cada dos posiciones

Podemos hacer stride, pero cortado, poniendo el valor inicial, el valor final, y los saltos

Name[0:5:2]



Para obtener la longitud usamos el comando len, ejemplo len("Hola") dará como resultado 4

Para combinar o concatenar cadenas usamos el símbolo +, ejemplo>

Nombre =Carlos, Saludo= Nombre + "Es el mas mejor", Saludo será igual a Carlos Es el mas mejor

Para replicar cadenas, usamos el símbolo de multiplicaci[on *

"Carlos"*3 , dará como resultado CarlosCarlosCarlos

Las cadenas son inmutables, ósea que no se pueden cambiar de valor ósea no se puede asignar así Nombre[0]="F", esto no dará como resultado Farlos, sino que dará un error

\, diagonal invertida representan secuencias de escape

\n representa enter o pasara la siguiente renglón

\t, es un espacio de tabulador

\\, Si necesitamos escribir solo una diagonal invertida, debemos de poner 2, y solo aparece una, o ponemos la letra r antes de la primera comilla doble de la cadena.

Los métodos

El método mayúscula o uppper(), cambia todo a mayúscula, ejemplo>

A="Hola", B=A.upper(), El valor de B ahora es "HOLA"

El método reemplazar o replace(, rremplaza una cadena por otra

Al usar el método, ponemos primero la palabra a reemplazarse y después la palabra que la sustituirá, ejemplo>

A='Carlos Sanchez', B=A.replace('Carlos', 'Juan'), el resultado será B igual 'Juan Sanchez'

El método encontrar, devuelve la posicion donde inicia un texto buscado también llamada substring, porque está dentro de una cadena, ejemplo

Si tenemos la variable Nombre="Raul Lopez", usamos el método Nombre.find('Lopez'), el resultado será el numero 6, porque lopez inicia en la posicion 5 de la cadena "Raul Lopez"

----------**Jupyter lab / Google Colabs**----------

```
#Crear diferentes strings de diversas formas

'Michael Jackson'

"Michael Jackson"

'@#2_#]&*^%$'

"1 2 3 4 5"

#Imprimir cadenas

print('Hola Mundo')

name = "Michael Jackson"

name

#Imprimir elementos con indices en diferentes posiciones

name="Rafael Oliveiros"

print(name[0])

print(name[6])

print(name[8])

#Imprimir elementos en posiciones invertidas

name="Rafael Oliveiros"

print(name[-2])

print(name[-9])

print(name[-1])

#Onbtener la longitud de una cadena

len("Michael Jackson")

#Obtener elementos d euna cadena en diferentes posiciones, slice

name='Pedrito Sola'

name[0:4]

name[3:8]

#Obtener elementos de una cadena dando determinados saltos

name="Thor Odinson"

name[::2]

#Obtener elementos de una cadena dando determinados saltos, pero dentro de un rango seleccionado

name='Septiembre patrio'

name[0:5:2]
```

```
#Concatenar dos o mas cadenas e imprimirlas
nombre='Carlos'
frase=nombre + 'es el mejors'
frase
#multiplicar una cadena
5 * 'frase repetida '
#Usar 3 secuencias de escape
print(" Michael Jackson \n is the best" )
print(" Michael Jackson \t is the best" )
print(" Michael Jackson \\ is the best" )
print(r" Michael Jackson \ is the best" )
#USar operadores de cadenas
a = "Hola mundo"
print("en minuscula es:", a)
b = a.upper()
print("en mayuscula es", b)
g='carlos'
g.upper()

a = "OScar is the best"
b = a.replace('OScar', 'Carlos')
b
g.replace('ar','al')

name = "Michael Jackson"
name.find('el')
g.find('al')
#Ver que ocurre cuando no encuentra
name.find('Jasdfasdasdf')
#imrprimir diagonal invertida
print("\\\\\\")
print(r"\ ")
```

In [1]:
```
#Crear diferentes strings de diversas formas
'Michael Jackson'
"Michael Jackson"
'@#2_#]&*^%$'
"1 2 3 4 5"
```

Out[1]: '1 2 3 4 5'

In [2]:
```
#Imprimir cadenas
print('Hola Mundo')
name = "Michael Jackson"
name
```

```
Hola Mundo
```

Out[2]: 'Michael Jackson'

In [3]:
```
#Imprimir elementos con indices en diferentes posiciones
name="Rafael Oliveiros"
print(name[0])
print(name[6])
print(name[8])
```

```
R

l
```

In [4]:
```
#Imprimir elementos en posiciones invertidas
name="Rafael Oliveiros"
print(name[-2])
print(name[-9])
print(name[-1])
```

```
o
O
s
```

In [5]:
```
#Onbtener la longitud de una cadena
len("Michael Jackson")
```

Out[5]: 15

In [6]:
```
#Obtener elementos d euna cadena en diferentes posiciones, slice
name='Pedrito Sola'
name[0:4]
name[3:8]
```

Out[6]: 'rito '

In [7]:
```
#Obtener elementos de una cadena dando determinados saltos
name="Thor Odinson"
name[::2]
```

Out[7]: 'To dno'

In [8]:
```
#Obtener elementos de una cadena dando determinados saltos, pero dentro de un ra
name='Septiembre patrio'
name[0:5:2]
```

Out[8]: 'Spi'

In [9]:
```
#Concatenar dos o mas cadenas e imprimirlas
nombre='Carlos'
frase=nombre + 'es el mejors'
frase
```

Out[9]: 'Carloses el mejors'

In [10]:
```
#multiplicar una cadena
5 * 'frase repetida '
```

Out[10]: 'frase repetida frase repetida frase repetida frase repetida frase repetida '

In [11]:
```
#Usar 3 secuencias de escape
print(" Michael Jackson \n is the best" )
print(" Michael Jackson \t is the best" )
print(" Michael Jackson \\ is the best" )
print(r" Michael Jackson \ is the best" )
```

```
 Michael Jackson
 is the best
 Michael Jackson         is the best
 Michael Jackson \ is the best
 Michael Jackson \ is the best
```

In [12]:
```
#USar operadores de cadenas
a = "Hola mundo"
print("en minuscula es:", a)
b = a.upper()
print("en mayuscula es", b)
g='carlos'
g.upper()

a = "OScar is the best"
b = a.replace('OScar', 'Carlos')
b
g.replace('ar','al')

name = "Michael Jackson"
name.find('el')
g.find('al')
```

```
en minuscula es: Hola mundo
en mayuscula es HOLA MUNDO
```

Out[12]:
```
-1
```

In [13]:
```
#Ver que ocurre cuando no encuentra
name.find('Jasdfasdasdf')
```

Out[13]:
```
-1
```

In [14]:
```
#imrprimir diagonal invertida
print("\\\\\\")
print(r"\ ")
```

```
\\\
\
```

----------Quiz----------

QUESTION 1

1/1 punto (no calificado)

Consider the following string:

```
Numbers = "0123456"
```

How would you obtain the even elements?

- ○ Numbers[::3]
- ◉ Numbers[::2]
- ○ Numbers[::6]

✔

Guardar

Enviar Ha realizado 1 de 2 intentos

✔ Correcto (1/1 punto)

QUESTION 2

1/1 punto (no calificado)

What is the result of the following line of code:

```
"0123456".find('1')
```

- ○ 0
- ◉ 1
- ○ '1'
- ○ True

✔

Guardar

Enviar Ha realizado 1 de 2 intentos

✔ Correcto (1/1 punto)

`3 + 2 * 2`

- ○ `10`
- ◉ `7`
- ○ `9`
- ○ `12`

✔

Enviar Ha realizado 2 de 2 intentos

Review Question 2

1/1 punto (calificado)

In Python, if you executed `name = 'Lizz'`, what would be the output of `print(name[0:2])`?

- ○ `Lizz`
- ○ `L`
- ○ `Liz`
- ◉ `Li`

✔

Enviar Ha realizado 2 de 2 intentos

✔ Correcto (1/1 punto)

Review Question 3

1/1 punto (calificado)

In Python, if you executed `var = '01234567'`, what would be the result of `print(var[::2])`?

- (•) `0246`
- ( ) `1357`
- ( ) `1234567`
- ( ) `8903`

✔

Enviar Ha realizado 2 de 2 intentos

✔ Correcto (1/1 punto)

Review Question 4

1/1 punto (calificado)

In Python, what is the result of the following operation `'1'+'2'`?

- ( ) `'2'`
- ( ) `'3'`
- (•) `'12'`
- ( ) `3`

✔

Guardar

Enviar Ha realizado 1 de 2 intentos

✔ Correcto (1/1 punto)

Review Question 5

1/1 punto (calificado)

Given `myvar = 'hello'`, how would you convert `myvar` into uppercase?

- `len(myvar)`
- `myvar.find('hello')`
- `myvar.upper()` ✔
- `myvar.sum()`

Guardar

Enviar Ha realizado 1 de 2 intentos

✔ Correcto (1/1 punto)