

	<b>Carátula para entrega de prácticas</b>	
Facultad de Ingeniería	Laboratorio de docencia	

# Laboratorios de computación salas A y B

<i>Profesor:</i>	MARCO ANTONIO MARTINEZ QUINTANA
<i>Asignatura:</i>	ESTRUCTURA DE DATOS Y ALGORITMOS I
<i>Grupo:</i>	17
<i>No de Práctica(s):</i>	9
<i>Integrante(s):</i>	José Luis Arroyo Chavarría
<i>No. de Equipo de cómputo empleado:</i>	1
<i>No. de Lista o Brigada:</i>	5
<i>Semestre:</i>	2
<i>Fecha de entrega:</i>	01/03/2020
<i>Observaciones:</i>	

**CALIFICACIÓN:** \_\_\_\_\_

**Objetivo:**

Conocer y desarrollar este nuevo lenguaje de programación que es Python

**Introducción:**

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma.

Python usa tipado dinámico y conteo de referencias para la administración de memoria.

Una característica importante de Python es la resolución dinámica de nombres; enlaza un método y un nombre de variable durante la ejecución del programa.

Se pueden escribir nuevos módulos fácilmente en C o C++. Python puede incluirse en aplicaciones que necesitan una interfaz programable.

**Cadenas:**

Son definidas usando comilla simple (') o comilla doble ("). Una característica especial de las cadenas es que son inmutables, esto quiere decir que no se pueden cambiar los caracteres que contiene. El carácter \ sirve para escapar caracteres como \n o \t.

**Operadores:**

Aritméticos: +, -, \*, /

**Desarrollo y resultados:**

- **Código**

**1. Hola Mundo**

```
>>> #Se puede utilizar comillas dobles o simples para crear una cadena
>>> cadena = "Hola Mundo" #Variable de tipo cadena
>>> print(cadena)
Hola Mundo
```

**2. Variables**

```
#Iniciando variables
>>> x = 10 #variable de tipo entero
>>> print(x) #función para imprimir los valores de las variables
10
```

```
>>> x = y = z = 10
>>> print(x,y,z)
10 10 10
```

### 3. Tipo de variable

```
>>> #La funcion type() permite conocer el tipo de una variable
>>> type(x)
<class 'int'>
>>> type(cadena)
<class 'str'>
```

### 4. Cadenas

```
>>> #Inicializando cadenas
>>> cadena1 = 'Hola'
>>> cadena2 = "Mundo"
>>> print(cadena1)
Hola
>>> print(cadena2)
Mundo
>>> concat_cadenas = cadena1 + cadena2 #Concatenación de cadenas
>>> print(concat_cadenas)
HolaMundo
```

### 5. Función str()

```
>>> #Para concatenar un número y una cadena se debe usar la función str()
>>> num_cadena = concat_cadenas + ' ' + str(3) #Se agrega una cadena vacía
para agregar un espacio
>>> print(num_cadena)
HolaMundo 3
```

### 6. {} en la cadena

```
>>> #El valor de la variable se va a imprimir en el lugar donde se encuentre {}
en la cadena
>>> num_cadena = "{} {} {}".format(cadena1, cadena2, 3)
>>> print(num_cadena)
```

Hola Mundo 3

## 7. Función format()

>>> #Cuando se agrega un número dentro de {#}, el valor la variable que se encuentra en esa posición

>>> #dentro de la función format(), será impreso.

>>> num\_cadena = "Cambiando el orden: {1} {2} {0} #".format(cadena1, cadena2, 3)

>>> print(num\_cadena)

Cambiando el orden: Mundo 3 Hola #

## 8. Operadores

>>> #Para el exponente se puede utilizar asterisco

>>> print( 1 + 5 )

6

>>> print( 6 \* 3 )

18

>>> print( 10 - 4 )

6

>>> print( 100 / 50 )

2.0

>>> print( 10 % 2 )

0

>>> print( ((20 \* 3) + (10 +1)) / 10 )

7.1

>>> print( 2\*\*2 )

4

- Captura de pantalla

```
>>> #Iniciando variables
>>> x = 10 #variable de tipo entero
>>> print(x) #funcion para imprimir los valores de las variables
10
```

```
>>> #Se puede utilizar comillas dobles o simples para crear una cadena
>>> cadena = "Hola Mundo" #Variable de tipo cadena
>>> print(cadena)
Hola Mundo
>>> #Asigna un mismo valor a tres variables
```

```
>>> x = y = z = 10
>>> print(x,y,z)
10 10 10
>>> #La funcion type
```

```
>>> #La funcion type() permite conocer el tipo de una variable
>>> type(x)
<class 'int'>
>>> type(cadena)
<class 'str'>
```

```
>>> #Inicializando cadenas
>>> cadena1 = 'Hola'
```

```
>>> cadena2 = "Mundo"
>>> print(cadena1)
Hola
>>> print(cadena2)
Mundo
>>> concat_cadenas = cadena1 + cadena2 #Concatenacion de cadenas
>>> print(concat_cadenas)
HolaMundo
```

```
>>> #Para concatenar un número y una cadena se debe usar la función str()
>>> num_cadena = concat_cadenas + ' ' + str(3) #Se agrega una cadena vacía para agregar un espacio
>>> print(num_cadena)
HolaMundo 3
>>>
>>> #El valor de la variable se va a imprimir en el lugar donde se encuentre {} en la cadena
>>> num_cadena = "{} {} {}".format(cadena1, cadena2, 3)
>>> print(num_cadena)
Hola Mundo 3
>>>
>>> #Cuando se agrega un número dentro de {#}, el valor la variable que se encuentra en esa posición
>>> #dentro de la función format(), será impreso.
>>> num_cadena = "Cambiando el orden: {1} {2} {0} #".format(cadena1, cadena2, 3)
>>> print(num_cadena)
Cambiando el orden: Mundo 3 Hola #
```

```
>>> #Para el exponente se puede utilizar asterisco
>>> print( 1 + 5 )
6
>>> print( 6 * 3 )
18
>>> print( 10 - 4 )
6
>>> print( 100 / 50 )
2.0
>>> print( 10 % 2 )
0
>>> print( ((20 * 3) + (10 +1)) / 10 )
7.1
>>> print( 2**2 )
4
```

## Conclusión:

Durante la practica vi las diferencias con los lenguajes C y C++, en el desarrollo de esta vi que Python es una buena opción por lo fácil de utilizar ya que otros lenguajes son un pocos complejos, aunque solo lo he utilizado gracias a los ejemplos puede

que se utilice este lenguaje para trabajos sencillos o para más complejos

**Bibliografías y Cibergrafías:**

- [https://github.com/eegkno/FI\\_UNAM/blob/master/02\\_Estructuras\\_de\\_datos\\_y\\_algoritmos\\_1/P09/EDyA09\\_I.ipynb](https://github.com/eegkno/FI_UNAM/blob/master/02_Estructuras_de_datos_y_algoritmos_1/P09/EDyA09_I.ipynb)
- <https://www.python.org/downloads/release/python-382/>
- <https://es.wikipedia.org/wiki/Python>