Python

ADA1: Investigación documental de lenguajes de programación. Equipo: "Programadores Insomnes"

Integrantes:

Campos Daguer Emilio Couoh Martin Reynaldo Graniel Arzat Aaron Isaac Herrera Herrera Adiel Elioenai Méndez Sierra Daniel

Instrucciones: En equipo, seleccionar un lenguaje de programación (Java, Python, C++, C#, JavaScript, VBScript, Ruby, etc.), realizar una investigación documental que sintetice la información que se solicita y elaborar una presentación (máximo 15 diapositivas).

Antecedentes de Phython

Guido van Rossum el creador de este programa empezó a idearlo a finales de los 80 para implementarlo en 1989.



Posteriormente en 1991 sería liberada la versión para el público (versión 0.9.0).



En octubre del 2000 sería publicada la segunda versión.



La versión 3.0 sería publicada en diciembre de 2008.

Cuenta con estructuras de datos eficientes y de alto nivel, así como un **enfoque** simple pero efectivo hacia la programación orientada a objetos.

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional.

Principales palabras reservadas

False: Valor booleano para operaciones de

comparación o lógicas que denota un valor falso.

None: Valor nulo

True: Valor booleano para operaciones

comparativas o lógicas que denota un valor

verdadero

And- Operador lógico

As- Alias

Assert- Utilizado con fines de depuración

Async- De la biblioteca "asyncio" de Python que es utilizado par código recurrente en el lenguaje

Await- De la biblioteca "asyncio" de Python que es utilizado par código recurrente en el lenguaje

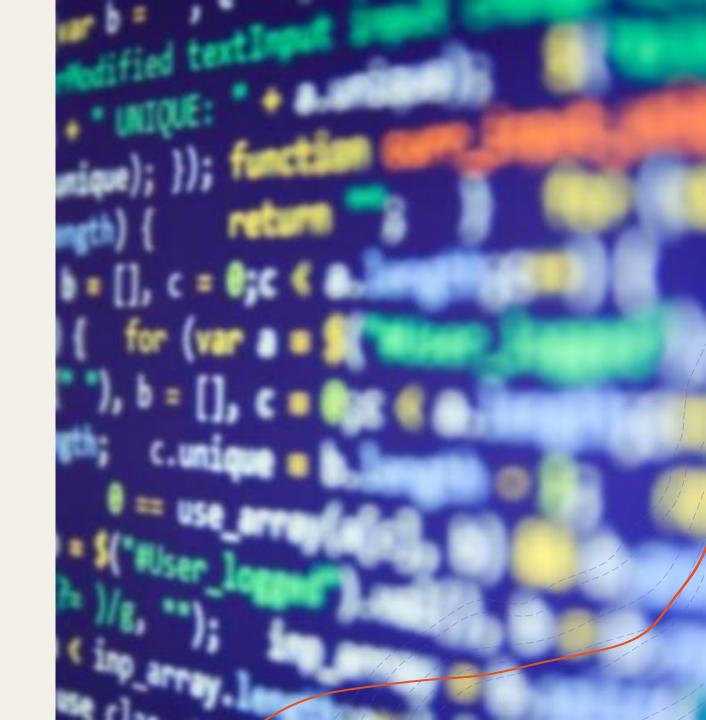
Break- Alterar el comportamiento de bucles for y while

Class- Definir una nueva clase

Def- Definir una función

Del- Eliminar

Elif, Else, If – Declaraciones u operadores condicionales



Except, raise, try - Crear excepciones **Finally -** Tienen la función de garantizar que un bloque de código se ejecute

For- Bucles

From- Importar partes específicas de un módulo

Global- Declarar variable global

Import- Importar un módulo

Iń- Comprobar si un valor está incluido

Is- Probar si dos variables se refieren al mismo objeto.

Lambda- Crear función anónima

Nonlocal- Declarar variable no local

Not- Operador lógico

Or- Operador lógico

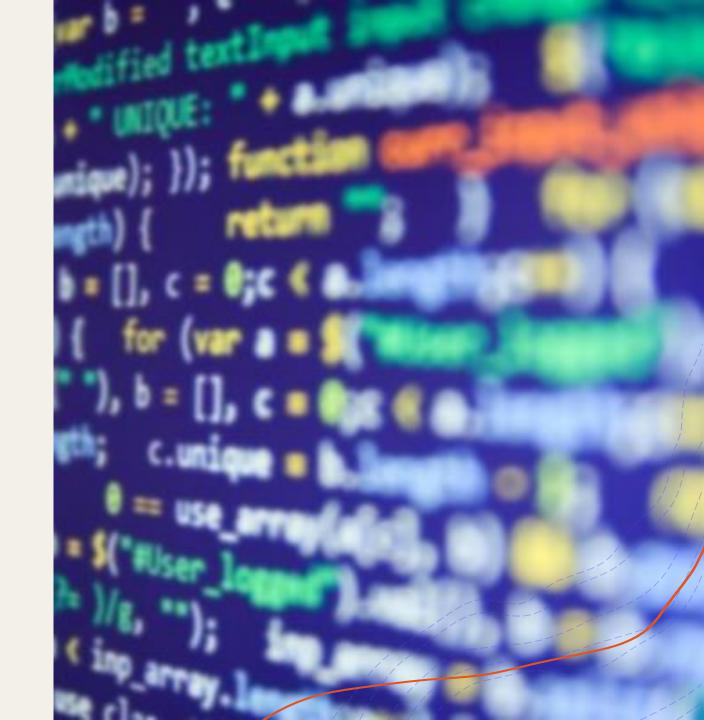
Pass- Declaración nula utilizado como marcador de posición

Return- Para salir y devolver el valor a una función.

While- Crear bucles

With- Simplificar el menejo de excepciones

Yield- Para salir y devolver el valor a una función, con la excepción que devuelve un generador.



Tipos de variables más importantes.

int: representan números enteros, positivos y negativos.

```
year = 2021
dia = 7
edad = 26
temperatura = -5
angulo = -45
```

bool: datos binarios útiles para expresiones condicionales y comparaciones

```
esta_frio = True
es_bajo = False
```

float: representan números de coma flotante o decimales (números reales)

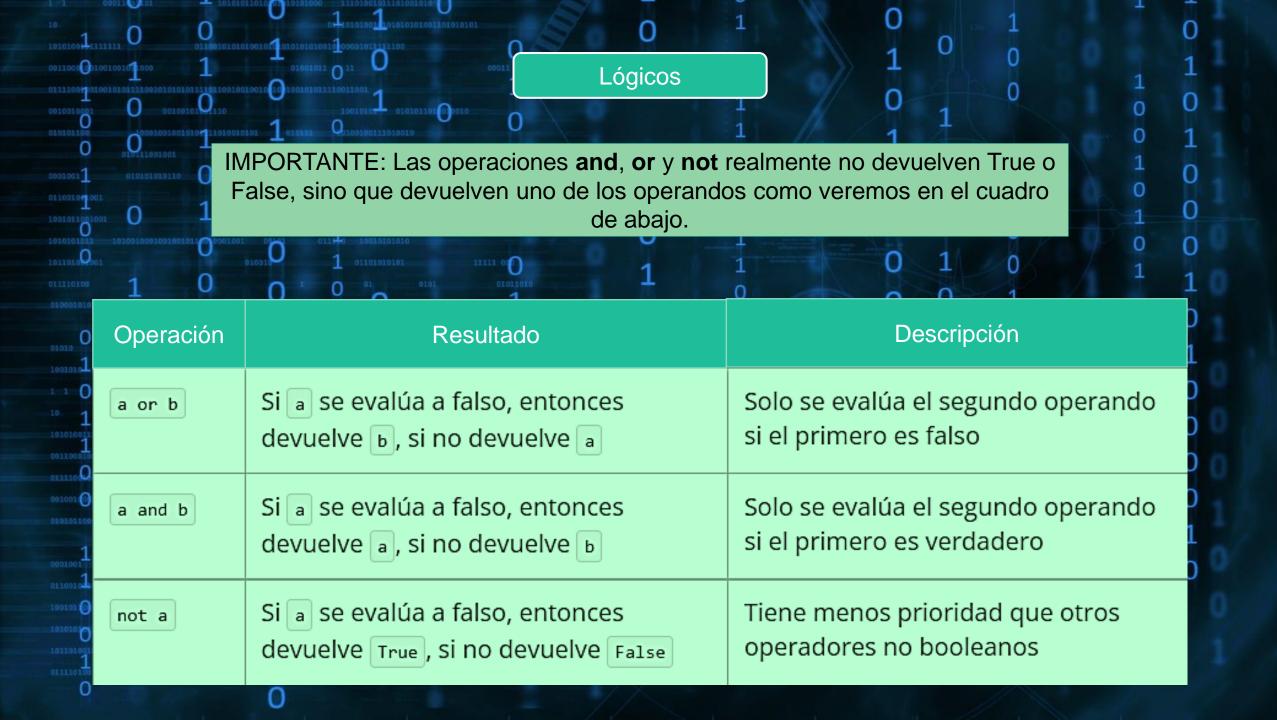
```
pi = 3.1416
estatura = 1.84
peso = 85.6
temperatura = -5.55
edad = 26.0
```

string: arreglo de caracteres que forman cadenas para formar un mensaje u oración generalmente. Se pueden crear usando comillas simples, dobles o triples.

```
profesor = "Sergio Castaño Giraldo"
web = 'Control Automático Educación'
cursos = """
1. Python
2. Matlab
3. MicroPython
4. Arduino
5. PIC
6. Control de Procesos
"""
```

Operadores

0 010111001001	Aritméticos		0	Relacionales	1 0	
+	Suma dos operandos	0	>	izquierda es estricta	Mayor que. True si el operando de la izquierda es estrictamente mayor que el de la derecha; False en caso	
	Resta al operando de la izquierda el valor del de la derecha. Utilizado sobre un único operando, le cambia el signo.	1		contrario.	e en caso	
		0	>=	Mayor o igual que. T i de la izquierda es m	ayor o igual que el	
*	Producto de dos operandos	1		de la derecha; False e		
1	Divide el operando de la izquierda por el de la derecha. El resultado siempre es un float	1	<	Menor que. True si el izquierda es estricta el de la derecha; Fals contrario.	mente menor que	
%	Operador módulo. Obtiene el resto de dividir el operando de la izquierda por el de la derecha	0	<=	Menor o igual que. To de la izquierda es mo de la derecha; False e	enor o igual que el	
//	Obtiene el cociente entero de dividir el operando de la izquierda por el de la derecha		==	Igual. True si el opera	ındo de la izquierda	
**	Potencia. El resultado es el operando de la izquierda elevado a la potencia del operando de la derecha.			caso contrario.		
		1	!=	Distinto. True si los o _l distintos; False en cas		



Instrucciones de control de flujo

Condicionales:

Permiten comprobar condiciones para ejecutar o no acciones. Analizan si es verdadera o falsa la condición (True/False).

(If, Elif, Else)

```
1 compra = 160
2 if compra >= 300:
3    print ("20% de descuento")
4    compra = compra*0.8
5 elif compra > 150 and compra < 300:
6    print ("10% de descuento")
7    compra = compra*0.9
8 else:
9    print ("No aplica descuento")</pre>
```

Instrucciones de control de flujo

Repetitivas:

Permiten ejecutar un mismo código, o fragmento del mismo, repetidas veces mientras se cumpla una condición o iterar sobre una variable compleja

(While, For)

```
1  x = 1
2  while x <= 5:
3     print ("Prueba #", x)
4     x += 1
5
6  lista_1 = ['aguacate', 'tomate', 'perejil']
7  for z in lista_1:
8     print (z)
9
10  for z in "abc":
11     print (z)</pre>
Prueba # 1
Prueba # 2
Prueba # 3
Prueba # 4
Prueba # 5
aguacate
tomate
perejil
a

tomate
perejil
```

Bibliotecas/Funciones más relevantes.



Librería que te permite hacer peticiones por http de una manera sencilla.



Scrapy es un framework que te permitirá rastrear sitios web y extraer datos estructurados.



Herramienta que agrega soporte para abrir, manipular y guardar muchos formatos de imágenes diferentes.



Biblioteca de código abierto que proporciona estructuras de datos y herramientas de análisis de datos.

Bibliotecas/Funciones más relevantes.



Biblioteca que produce gráficas de buena calidad en una variedad de formatos y entornos interactivos.



Plataforma líder para la creación de programas en Python para trabajar con datos de lenguaje humano.



Biblioteca más usada para computación científica, contiene funciones sofisticadas, como herramientas para integración de código, etc.

Funciones básicas de entrada y salida.

Entrada de datos con input:

Permite obtener información de la terminal escrita el teclado, se ingresa el texto que se necesite y se presiona la tecla enter.

```
print("¿Cómo se llama?")
nombre = input()
print(f"Me alegro de conocerle, {nombre}")
```

```
nombre = "Pepe"
edad = 25
print("Me llamo", nombre, "y tengo", edad, "años.")
```

Salida de datos con print:

Permite mostrar información por consola en la pantalla como mensajes, números o valores de una variable. El texto a mostrar se escribe como argumento de la función.

```
Ejercicio 1
    # Ejercicio_Promedio
    num1 = float(input("Escriba un número: "))
3
    num2 = float(input("Escriba un número: "))
    promedio = (num1+num2)/2
6
    print(promedio)
```

Ejercicio 2

```
#Ejercicio_While
    num = 0
3
    While num<10
5
        contador = num + 1
6
    print(contador)
8
```

Referencias

- + https://www.mclibre.org/consultar/python/otros/historia.html
- + https://www.iartificial.net/librerias-de-python-para-machine-learning/
- + https://www.mclibre.org/consultar/python/lecciones/python-entradateclado.html
- https://www.mclibre.org/consultar/python/lecciones/python-salidapantalla.html
- + https://j2logo.com/python/tutorial/operadores-en-python/#operadores-logicos
- https://www.codigofuente.org/variables-en-python/https://eiposgrados.com/blog-python/palabras-reservadas-python/