GUÍA DE LABORATORIO 01
Implementar Entorno de Desarrollo PYTHON-VSCode – Introducción Python

Asignatura	Datos del alumno	Fecha y Firma
Algoritmos y solución de problemas	Apellidos:	
	Nombre:	

Instrucciones:

Desarrollar las actividades que indica el docente en base a la guía de trabajo que se presenta.

1. **Objetivos:** Crear un entorno de trabajo eficiente y productivo para desarrollar aplicaciones en Python utilizando las herramientas y funcionalidades que ofrece VS Code:
   - Configuración del entorno.
   - Familiarización con las herramientas de VS Code: Explorar las características específicas para Python, como la ejecución de scripts, la depuración interactiva, la gestión de entornos virtuales.

2. **Equipos, Herramientas o Materiales**
   - Computador
   - Software: Python, Visual Studio Code
   - Extensiones de VS Code: Extensión de Python de Microsoft, Pylance, Python Indent, Python Test Explorer, Code Runner

3. **Fundamento Teórico**

   ¿Qué es Python?

   Python es un lenguaje de programación muy popular. Fue creado por Guido van Rossum y lanzado en 1991.

   Se utiliza para:
   - desarrollo web (del lado del servidor),
   - desarrollo de software,
   - matemáticas,
   - secuencias de comandos del sistema.

   ¿Qué puede hacer Python?
   - Python se puede utilizar en un servidor para crear aplicaciones web.
   - Python se puede utilizar junto con software para crear flujos de trabajo.
   - Python puede conectarse a sistemas de bases de datos. También puede leer y modificar archivos.
   - Python se puede utilizar para gestionar grandes cantidades de datos y realizar operaciones matemáticas complejas.
   - Python se puede utilizar para la creación rápida de prototipos o para el desarrollo de software listo para producción.

   ¿Por qué Python?

- Python funciona en diferentes plataformas (Windows, Mac, Linux, Raspberry Pi, etc.).
- Python tiene una sintaxis simple similar al idioma inglés.
- Python tiene una sintaxis que permite a los desarrolladores escribir programas con menos líneas que otros lenguajes de programación.
- Python se ejecuta en un sistema de interpretación, lo que significa que el código se puede ejecutar tan pronto como se escribe, lo que permite crear prototipos con gran rapidez.
- Python puede tratarse de forma procedimental, orientada a objetos o funcional.

Sintaxis de Python comparada con otros lenguajes de programación

- Python fue diseñado para ser legible y tiene algunas similitudes con el idioma inglés con influencia de las matemáticas.
- Python utiliza nuevas líneas para completar un comando, a diferencia de otros lenguajes de programación que a menudo utilizan punto y coma o paréntesis.
- Python se basa en la sangría, utilizando espacios en blanco, para definir el alcance, como el alcance de bucles, funciones y clases. Otros lenguajes de programación suelen utilizar llaves para este propósito.

Tipos de Datos

- Numérico: Integer, Float, Complex.
- Cadenas: String
- Booleanos: True or False
- Listas, Diccionarios, Tuplas y Conjuntos.

Funciones integradas

En Python tenemos muchas funciones integradas. Las funciones integradas están disponibles globalmente para su uso, lo que significa que puede hacer uso de las funciones integradas sin importar ni configurar. Algunas de las funciones integradas de Python más utilizadas son las siguientes: print() , len() , type() , int() , float() , str() , input() , list() , dict() , min() , max() , sum() , sorted() , open() , file() , help() y dir() . En la siguiente tabla verá una lista exhaustiva de funciones integradas de Python tomadas de la documentación de Python.

		Built-in Functions		
abs()	delattr()	hash()	memoryview()	set()
all()	dict()	help()	min()	setattr()
any()	dir()	hex()	next()	slice()
ascii()	divmod()	id()	object()	sorted()
bin()	enumerate()	input()	oct()	staticmethod()
bool()	eval()	int()	open()	str()
breakpoint()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	

Variables

Las variables almacenan datos en la memoria de una computadora. Se recomienda el uso de variables mnemotécnicas en muchos lenguajes de programación. Una variable mnemotécnica es un nombre de variable que se puede recordar y asociar fácilmente. Una variable se refiere a una dirección de memoria en la que se almacenan datos. No se permiten números al principio, caracteres especiales ni guiones al nombrar una variable. Una variable puede tener un nombre corto (como x, y, z), pero se recomienda encarecidamente un nombre más descriptivo (nombre, apellido, edad, país).

Reglas para nombrar variables en Python

- Un nombre de variable debe comenzar con una letra o el carácter de guión bajo.
- Un nombre de variable no puede comenzar con un número
- Un nombre de variable solo puede contener caracteres alfanuméricos y guiones bajos (Az, 0-9 y _)
- Los nombres de las variables distinguen entre mayúsculas y minúsculas (firstname, Firstname, FirstName y FIRSTNAME son variables diferentes)

A continuación, se muestran algunos ejemplos de nombres de variables válidos:

```
firstname
lastname
age
country
city
first_name
last_name
capital_city
_if # if we want to use reserved word as a variable
year_2021
year2021
current_year_2021
```

Nombres de variables no válidos

```
first-name
first@name
first$name
num-1
1num
```

Usaremos el estilo estándar de nombres de variables de Python que han adoptado muchos desarrolladores de Python. Los desarrolladores de Python usan la convención de nombres de variables snake_case (snake_case).

4. Implementar Entorno de Desarrollo

4.1. Instalar Python

- Visita la página oficial de Python (https://www.python.org/downloads/) y descarga la última versión estable para tu sistema operativo (Windows, macOS o Linux)..
- Ejecuta el instalador y asegúrate de marcar la casilla "Add Python to PATH" durante la instalación. Esto permitirá que VS Code encuentre e interprete tu código Python.

4.2. Instalar Visual Studio Code

- Descarga e instala VS Code desde su sitio web oficial (https://code.visualstudio.com/).

4.3. Configurar VS Code para Python

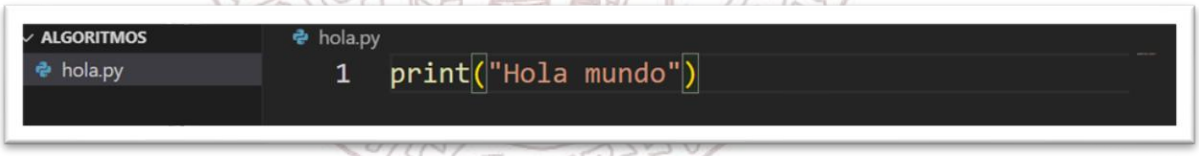
- Extensión de Python – Instalar extensión oficial de Microsoft.
- Seleccionar el intérprete de Python – Presiona CTRL+SHIFT+P y escribe: "Python: Select Interpreter" y selecciona la versión de Python que acabas de instalar.

4.4. Extensiones Recomendadas:

- **Pylance**: Proporciona autocompletado inteligente, verificación de tipos y navegación por el código.
- **Python Indent**: Formatea automáticamente tu código Python con la indentación correcta.
- **Python Test Explorer**: Facilita la ejecución y depuración de pruebas unitarias.
- **Code Runner**: Permite ejecutar fragmentos de código Python directamente desde el editor.
- **autoDocstring**: Genera automáticamente cadenas de documentación para tus funciones y clases.

4.5. Crear y Ejecutar tu Primer Programa

- Crea un nuevo archivo en VS Code (Ctrl+N) y guárdalo con la extensión .py (por ejemplo, hola.py).
- Escribe el siguiente código:



- Presiona F5 para ejecutar tu código. Deberías ver "¡Hola, mundo!" impreso en la terminal integrada de VS Code.

5. Implementación de Algoritmos en Python y Actividades.

Archivo Python

Dentro de nuestro directorio, crea un archivo llamado hola.py. Ahora, hagamos lo siguiente usando Visual Studio Code.

hola.py

```python
# LABORATORIO 01 ALGORITMOS Y SOLUCION DE PROBLEMAS

print(2 + 3)             # addition(+)
print(3 - 1)             # subtraction(-)
print(2 * 3)             # multiplication(*)
print(3 / 2)             # division(/)
print(3 ** 2)            # exponential(**)
print(3 % 2)             # modulus(%)
print(3 // 2)            # Floor division operator(//)

# CHECKING DATA TYPES
print(type(10))          # Int
print(type(3.14))        # Float
print(type(1 + 3j))      # Complex number
print(type('MSc Edem'))  # String
print(type([1, 2, 3]))   # List
print(type({'name':'Edem'})) # Dictionary
print(type({9.8, 3.14, 2.7}))    # Set
print(type((9.8, 3.14, 2.7)))    # Tuple
```

Ejemplo 1 Variables:

Cuando asignamos un determinado tipo de datos a una variable, se denomina declaración de variable. Por ejemplo, en el ejemplo siguiente, mi nombre se asigna a una variable first_name. El signo igual es un operador de asignación. Asignar significa almacenar datos en la variable. El signo igual en Python no es igualdad como en Matemáticas.

variables.py > ...

```python
# VARIABLES EN PYTHON
first_name = 'Edem'
last_name = 'Terraza'
country = 'Perú'
city = 'Ayacucho'
age = 32
is_married = False
skills = ['HTML', 'CSS', 'JS', 'React', 'Python']
person_info = {
    'firstname':'Edem',
    'lastname':'Terraza',
    'country':'Perú',
    'city':'Ayacucho'
    }
```

Utilicemos las funciones integradas print() y len() . La función print acepta una cantidad ilimitada de argumentos. Un argumento es un valor que se puede pasar o colocar dentro de los paréntesis de la función, vea el ejemplo a continuación.

```python
print('Hello, World!') # El texto ¡Hola, mundo! es un argumento
print('Hello',',', 'World','!') # puede tomar múltiples argumentos, se han pasado cuatro argumentos
print(len('Hello, World!')) # solo se necesita un argumento
```

Actividad – Estudiante:

Imprimir y encontrar también la longitud de las variables declaradas.

Declarar múltiples variables en una línea

```python
first_name, last_name, country, age, is_married = 'Edem', 'Terraza', 'Peru', 32, False

print(first_name, last_name, country, age, is_married)
print('First name:', first_name)
print('Last name: ', last_name)
print('Country: ', country)
print('Age: ', age)
print('Married: ', is_married)
```

Obtención de la información del usuario mediante la función incorporada input() . Asignaremos los datos que obtenemos de un usuario a las variables first_name y age. Ejemplo:

```python
first_name = input('What is your name: ')
age = input('How old are you? ')

print("El nombre es:", first_name)
print("La edad es:", age)
```

Comprobación de tipos de datos y conversión

Para comprobar el tipo de datos de ciertos datos/variables utilizamos el tipe Ejemplo:

```python
# DIFERENTES TIPOS DE DATOS PYTHON
# Declaremos variables con varios tipos de datos.

first_name = 'Edem'     # str
last_name = 'Terraza'       # str
country = 'Perú'         # str
city= 'Ayacucho'            # str
age = 32                   # int

# IMPRIMIENDO TIPOS DE DATOS
print(type('Asabeneh'))     # str
print(type(first_name))     # str
print(type(10))             # int
print(type(3.14))           # float
print(type(1 + 1j))         # complex
print(type(True))           # bool
print(type([1, 2, 3, 4]))     # list
print(type({'name':'Edem','age':32, 'is_married':32}))    # dict
print(type((1,2)))                                          # tuple
print(type(zip([1,2],[3,4])))                               # set
```

EJERCICIOS:

1. Usando la función incorporada len() , encuentre la longitud de su nombre
2. Compara la longitud de tu nombre y tu apellido
3. Declara 5 como num_one y 4 como num_two
    - Sume num_one y num_two y asigne el valor a una variable total
    - Resta num_two de num_one y asigna el valor a una variable diff
    - Multiplica num_two y num_one y asigna el valor a una variable producto
    - Dividir num_one por num_two y asignar el valor a una variable división
    - Utilice la división de módulo para encontrar num_two dividido por num_one y asigne el valor a un resto variable
    - Calcula num_one a la potencia de num_two y asigna el valor a una variable exp
    - Encuentra la división del piso de num_one por num_two y asigna el valor a una variable floor_division
4. El radio de un círculo es de 30 metros.
    - Calcula el área de un círculo y asigna el valor a una variable llamada area_of_circle
    - Calcula la circunferencia de un círculo y asigna el valor a una variable llamada circum_of_circle
    - Tome el radio como entrada del usuario y calcule el área.
5. Utilice la función de entrada incorporada para obtener el nombre, apellido, país y edad de un usuario y almacenar el valor en sus nombres de variable correspondientes.