

# CUESTIONARIO

## 1.- Cuales son los tipos de datos en Python?

- Tipos de datos numericos:

Int = abreviatura de numero entero 1, 2, 3... , se utiliza para hacer operaciones matemáticas

Float = abreviatura de flotante, representa a los números con decimales 1.88 , 8.12

Complex = abreviatura de numero complejo, representa a los numeros que tienen una parte real y una parte imaginaria  $8 + 2j$  ,  $7j$

- Tipos de texto:

Str = abreviatura de string/cadena de caracteres, representa a una secuencia de caracteres que pueden ser letras o numeros , etc.

Se representa con comillas simples o dobles

«Hi Word»

- Tipos booleanos

Bool = abreviatura de booleano, son valores verdaderos o falsos.

Se representa: True , False

Se usa principalmente en expresiones logicas IF o en declaraciones WHILE.

- Tipos de secuencia:

List = abreviatura de lista, representa una colección ordenada es decir se puede acceder a ellas por su posición, se usa [ ]

Ejemplo: [3,4,5] , [4, "z", False]

Tuple = abreviatura de tupla, es muy similar a listas, son inmutables, es decir una vez creada no pueden modificarse. Se representa mediante ( )

Ejm: (5,6,7,8,9) (l,m,n,o,p)

- Tipo de mapeo:

Dict = abreviatura de diccionario, almacena datos como pares clave/valor se pueden asociar, son útiles por que se pueden acceder a los datos de manera eficiente.

Se representa por llaves { }

Ejm: { 'nombre': 'Favio', 'edad': 40 }

- Tipo de conjunto:

Set= son elementos únicos e inmutables, es decir no hay duplicados ni pueden ser modificados.

Se utiliza { } o la función set( )

{7,8,9} = numeros                      { 'Hi', 'There' } = cadena

Set ([6,7,8,9])

## **2.- Que tipo de convención de nomenclatura deberíamos utilizar para las variables en Python ?**

- Cuando se tenga una variable que tenga varias palabras la convención es poner guion bajo entre las palabras:

a\_long\_sentence\_wiht\_numbers

- Para nombrar un método o variable se añadirá un guion bajo:

\_greeting = 'Hi'

- No se debe utilizar la: l (letra ele minúscula)  
O (letra o mayúscula)

Porque se confundirían con el uno y cero.

- Las clases deberán usar el formato UpperCamelCase (la primera letra deberá ser en mayúscula) y lowercase (todo en minúscula) lower\_case

IDENTIFICADOR	CONVENCIÓN
<b>Modulo</b>	<b>minúscula</b>
<b>Clase</b>	<b>CapWords</b>
<b>Funciones</b>	<b>minúscula</b>
<b>Variable de tipo</b>	<b>minúscula</b>
<b>Constantes</b>	<b>MAYÚSCULA</b>
<b>Paquete</b>	<b>minúscula</b>

### 3.- Que es un Heredoc en Python?

Es una forma de especificar un bloque de texto, manteniendo las sangrías, espacios, saltos de líneas en el texto.

Se representa por las comillas triples ('' '' '') para cadenas de varias líneas.

Ejemplo :

**#Heredoc**

**Content = '' '' ''**

Al contrario del pensamiento popular, el texto de Lorem Ipsum no es simplemente texto aleatorio.

Tiene sus raíces en una pieza clásica de la literatura del Latin, que data del año 45 antes de Cristo, haciendo que este adquiriera mas de 2000 años de antigüedad.

Richard McClintock, un profesor de Latin de la Universidad de Hampden-Sydney en Virginia, encontró una de las palabras más oscuras de la lengua del latín, "consecteur", en un pasaje de Lorem Ipsum, y al seguir leyendo distintos textos del latín, descubrió la fuente indudable. '' '' '' ''

**Print(content)**

**Esta es la forma de contruir un Heredoc**

### 4.- Que es una Interpolación de Cadenas?

Es el proceso de insertar valores o variables de forma dinámica en una cadena de texto.

La ventaja de utilizar la interpolación de cadenas hace que el código sea más legible, puede utilizarse con una amplia variedad de tipos de datos (números, fechas, objetos)

### **Ejemplo:**

```
name = 'Pedro'
age = 40
greeting = f'Hi, {name}! you have {age} years old'

Print(greeting)
```

## **5.- Como deberíamos usar los comentarios en Python?**

Los comentarios son como notas personales en el código, ayuda a explicar el propósito y también se puede usar como un recordatorio.

Los comentarios sirven para explicar las partes de un código, estos comentarios son ignorados por los ordenadores cuando se le ejecuta.

Se colocan de 2 formas: usando almohadilla al inicio del comentario o usando triple comillas al inicio y al final del comentario.

Ejem:

```
#Todo se comenta en Python
```

```
'''Este es un fragmento extenso'''
```

## **6.- Cuales son las diferencias entre aplicaciones Monolíticas y de Microservicios?**

### **Monolíticas**

Las aplicaciones monolíticas se han utilizados durante décadas, es decir es de una forma fácil de construir aplicaciones.

La interfaz de usuario, al acceso a los datos están todos acoplados se ejecutan en un solo proceso a la vez todos los módulos comparten los mismos recursos, son más difíciles de escalar al ser de una sola unidad. Esto hace que sea más difícil modificar las aplicaciones ya que estas deben ser actualizadas resultando ser un poco desafiante poder hacerlo.

La ventaja es la simplicidad en su desarrollo, se utiliza para pequeñas aplicaciones.

La desventaja es la ineficiente utilización de los recursos

## **Ejm: Plataformas tradicionales de e-commerce .**

### **Microservicios**

Es este tipo de arquitectura cada funcionalidad es independiente, lo cual lo hace más óptimo y mejora su rendimiento.

Una de las ventajas es su flexibilidad y su escalabilidad, promueve el desarrollo y despliegue independiente, reduciendo el riesgo de fallos.

Permitiendo innovar y experimentar nuevas tecnologías sin afectar a la aplicación.

Desventaja, gestionar una arquitectura de microservicios resulta complejo ya que requiere buenas herramientas para el monitoreo y registro de las aplicaciones,

**ejem: Netflix, Amazon, Spotify**

- **Factores a considerar al elegir entre la aplicación monolíticas o microservicios:**

Tamaño y Complejidad; para aplicaciones pequeñas y simples la aplicación monolíticas es la correcta. Para aplicaciones más complejas y de gran complejidad se usa la aplicación de microservicios.

La aplicación de Microservicio requiere mayor nivel de experiencia.