

Cuestionario M2C3

¿Cuáles son los tipos de Datos en Python?

- Primero tenemos Boolean, que se basa en True o False, después tenemos los números, que en el lenguaje python son muy complejos, y se dividen en varias secciones. Luego podemos encontrarnos con los 'string', que pueden pasar desde un simple nombre a todo un documento html. Las string son inmutables, es decir que una vez declaradas no se pueden modificar.

```
>name = 'Pablo'
```

Luego pasamos a otro tipo de dato muy complejo, que son los bytes y los byte arrays, que se usa por ejemplo, para alterar una foto, pero al ser byte por byte, es un tipo de información que necesita de formación para poder manejarla.

Luego tenemos el 'none', que tiene mucha utilidad cuando quieres declarar una variable, pero no tienes o no quieres ponerle la información. Ejemplo

```
>name = 'none'
```

Finalmente tenemos las listas, que a diferencia de los 'string', si se pueden modificar, y son dinamicas. Para diferencial de los 'string' voy a poner un ejemplo:

```
> list = [1, 2, 3, 4]
```

¿Qué tipo de convención de nomenclatura deberíamos utilizar para las variables en Python?

- La nomenclatura para Python tiene reglas de escritura para que los programadores, si tienen buenas prácticas, se puedan entender y no den lugar a confusiones. En primer lugar, los nombres de las variables, deben ir en minúsculas, y en segundo lugar, si el nombre de la variable tiene espacios, se expresaran con barra baja (_), para que sea mas claro pondré varios ejemplos:

```
>una_variable= ''
```

```
>AsiNoSeEscribe = ''
```

¿Qué es un Heredoc en Python?

- Los Heredocs en python son las “multiline string”, ya que estas, siguiendo la correcta escritura te permite procesar bloques de texto, pero la forma de declararlas es distinta a las de las “string”, en los siguientes ejemplos podemos ver como seria correcto escribirlas

```
>wrong heredoc = “ aaaa  
aaa  
aaaa”
```

```
>correct heredoc = “””
```

```
Nullam id dolor id nibh ultricies vehicula ut id elit. Nullam quis risus eget urna mollis  
ornare vel eu leo.
```

```
Vestibulum id ligula porta felis euismod semper. Cum sociis natoque penatibus et  
magnis dis parturient montes, nascetur ridiculus mus. Cras justo odio, dapibus ac  
facilisis in.
```

```
Integer posuere erat a ante venenatis dapibus posuere velit aliquet.
```

```
“””
```

¿Qué es una interpolación de cadenas?

- String interpolation es un método que permite insertar valores dentro de una variable, y la forma de hacerlo es la siguiente:

```
>name= ‘Pablo’
```

```
>school = ‘DevCamp’
```

```
> welcome_message = f”Hello {name}, welcome to {school}, thank you for choosing  
us”
```

```
>print(message)
```

```
output: Hello Pablo, welcome to DevCamp, thank you for choosing us
```

¿Cuándo deberíamos usar comentarios en Python?

- De primeras, es mejor evitar el uso de comentarios en el código, ya que puede dar a dudas y confusiones y provocar errores. Si que es verdad que se pueden usar comentarios para organizar el código o para explicar el flujo de trabajo, pero no debe ser algo permanente, ya que un comentario es significado de que no has sabido explicarte bien en el código.

¿Cuáles son las diferencias entre aplicaciones monolíticas y de microservicios?

- En este caso, lo que los diferencia es la forma en la que están organizadas las funciones.

Por una parte, las aplicaciones monolíticas tienen todas las funciones en un mismo lugar, bien tengan autenticación, publicaciones, log-ins. Eso quiere decir que si se hiciera un cambio en alguna de esas funciones, y saliera mal, toda la aplicación se vendría abajo.

Por otra parte, la aplicación tiene todas las funciones repartidas de forma individual, y para realizar su trabajo se comunican entre ellas, por lo tanto, si hiciéramos un cambio en una de ellas, como por ejemplo en la autenticación, y ese cambio resultara en error, y alguno de los otros módulos de la aplicación requiriera a la autenticación, simplemente resultaría en error, pero la aplicación seguirá funcionando.

Las aplicaciones monolíticas son normalmente aplicaciones pequeñas y con pocas secciones o menos complejas, o para hacer prototipos simples de las mismas.

En cambio, las aplicaciones microservice son aplicaciones grandes y complejas, además de requerir de más recursos que las monolíticas.

Muestro aquí un ejemplo gráfico de las diferencias más significativas:

