

Carolina Martínez García

## Checkpoint 3, preguntas teóricas

### ¿Cuáles son los tipos de datos en Python?

Hay nueve tipos de datos principales que se usan en Python, que son:

- **BOOLEAN**, que muestra un valor de verdadero o falso. Por ejemplo, un boolean **podría mostrar si un artículo se ha publicado o no** indicando "False" hasta que se publique.
- **NUMBER**: pueden ser una gran variedad de elementos diferentes y pueden ser enteros o decimales
- **STRING**: Las cadenas (string) pueden ser desde un nombre hasta un documento html, cualquier tipo de secuencia de bytes. Se encierran en comillas sencillas o dobles.
- **BYTES Y BYTES ARRAYS**. Estos tipos de datos son para el desarrollo complejo de Python para tareas como reorganizar secuencias de bytes o trabajar con imágenes a nivel bytes.
- **NONE**: Este tipo de datos en Python es muy útil cuando quieres definir una variable **pero todavía o no quieres determinar el valor**.
- **LIST, TUPLES, SETS, DICTIONARIES**: Estos tipos son para estructura de datos, para gestionar colecciones.

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

Se recomienda seguir la norma establecida en PEP8, una guía de estilos que hay que seguir para tener **una buenas prácticas**, en las que se indica, por ejemplo, la norma de poner **guión bajo entre las dos o más palabras que utilices** para escribir una variable, además de que hay que escribir dichas variables en minúsculas.

### ¿Qué es un Heredoc en Python?

Es una cadena de varias líneas, por ejemplo un conjunto de párrafos, como a continuación:

```
...
```

```
#Heredoc
```

```
content = """
```

```
I picked out these examples very intentionally. And the reason is because I
wanted to show you a great example of when you would use partition vs split.
So I'm going to continue our example. And let's imagine that we wanted to use
split with this heading.
```

```
Now technically it would work if we had started out with only this example or I
should say. Yeah so I'm just going to do everything from the comma I'm just
going to take out just like this
```

```
""").strip()
```

```
...
```

## ¿Qué es una interpolación de cadenas?

La interpolación sirve para ejecutar el código Python dentro de las cadenas, esto es muy útil para las páginas web donde los elementos van a cambiar dependiendo del comportamiento del usuario.

Por ejemplo:

```
...
```

```
number = 18
```

```
age = f'I am {number} years old
```

```
print(age)
```

```
#ejecutará
```

```
I am 18 years old
```

```
...
```

## ¿Cuándo deberíamos usar comentarios en Python?

Los comentarios a veces pueden ser confusos, así que es mejor poner los menos posible. La mejor opción es mejorar la denominación de los métodos y así será una mejor explicación del código.

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

Las aplicaciones monolíticas se crean en un solo código.

Las aplicaciones de microservicios se crean en muchos códigos independientes conectados mediante APIs.

Una de las ventajas de las aplicaciones de microservicios frente a las monolíticas es que si hay que hacer algún cambio y por lo que sea la web se bloquea, en la de microservicios sólo se va a bloquear la parte en la que estás trabajando, el resto seguirá funcionando. Si por ejemplo estás cambiando algo en la página de contacto, donde está simplemente nuestro teléfono, email y dirección física, y ahí surge un problema, si es una aplicación de microservicios el cliente podrá seguir sin problema haciendo su compra online ya que esa parte estará creada en otro código.

La monolítica suele ser más práctica si la página web a desarrollar no va a ser muy grande, si no es mejor hacer una de microservicios, aunque esta conlleve más presupuesto y mayor volumen de equipo de personas trabajando en ella.

Otra desventaja de la monolítica es que hay que codificar en un único lenguaje y la de microservicios acepta varios.