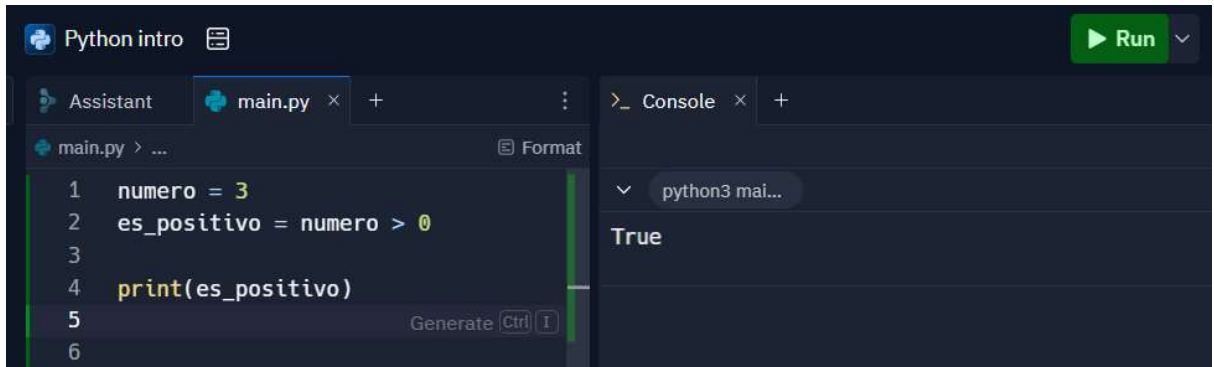


Checkpoint 3 preguntas teóricas:

¿Cuáles son los tipos de Datos en Python?

- **Booleans** : Una variable booleana es una variable que sólo puede tomar dos posibles valores: True (verdadero) o False (falso). Ej:

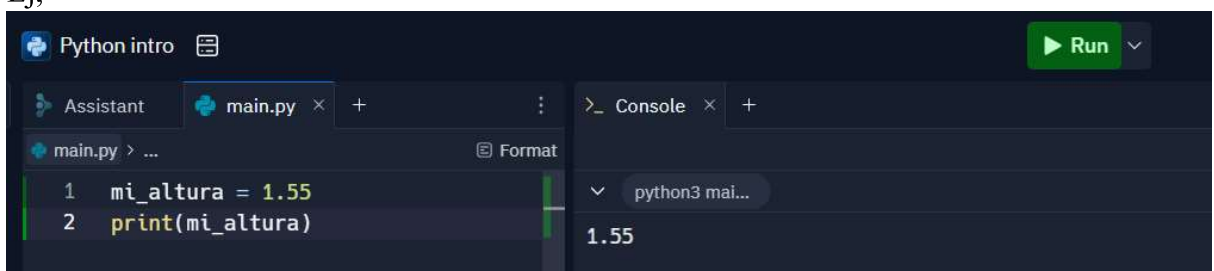


```
Python intro
Assistant main.py x +
main.py > ...
1 numero = 3
2 es_positivo = numero > 0
3
4 print(es_positivo)
5
6
```

Console

```
python3 mai...
True
```

- **Numbers**: Existen tres tipos numéricos distintos: números enteros, números de punto flotante y números complejos. Además, los booleanos son un subtipo de números enteros. Los números enteros tienen una precisión ilimitada. Ej: int, float, complex
Ej;

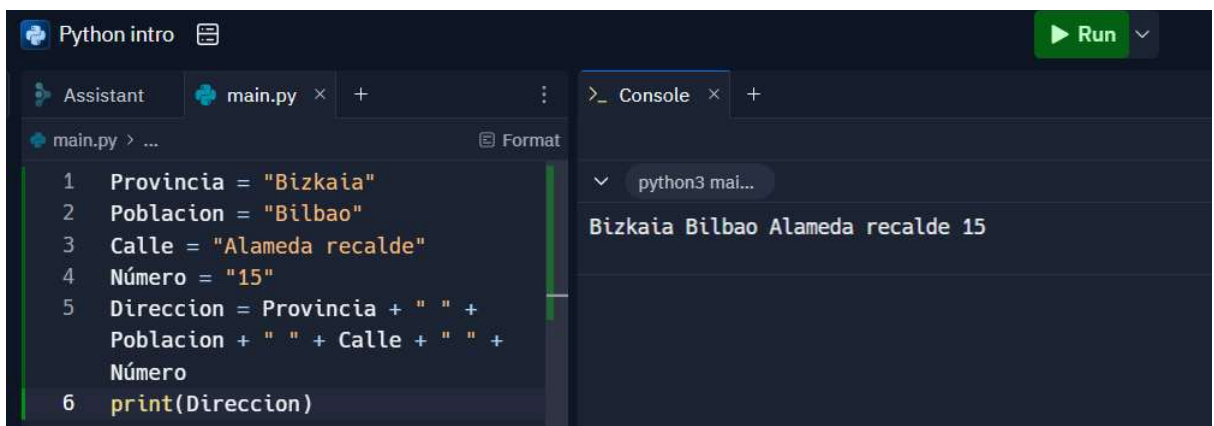


```
Python intro
Assistant main.py x +
main.py > ...
1 mi_altura = 1.55
2 print(mi_altura)
```

Console

```
python3 mai...
1.55
```

- **Strings**: Las cadenas (o strings) son un tipo de datos compuestos por secuencias de caracteres que representan texto. Estas cadenas de texto son de tipo str y se delimitan mediante el uso de comillas simples o dobles. Ej:



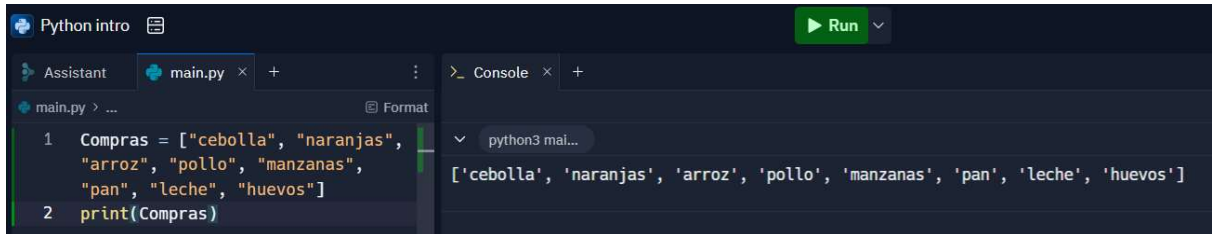
```
Python intro
Assistant main.py x +
main.py > ...
1 Provincia = "Bizkaia"
2 Poblacion = "Bilbao"
3 Calle = "Alameda recalde"
4 Número = "15"
5 Direccion = Provincia + " " +
6             Poblacion + " " + Calle + " " +
7             Número
8 print(Direccion)
```

Console

```
python3 mai...
Bizkaia Bilbao Alameda recalde 15
```

- **Bytes and byte arrays**: son tipos de datos utilizados para representar secuencias de bytes, es decir, datos binarios. Los bytes después de creados son inmodificables, mientras que los byte arrays son modificables.

- None: es un valor especial que representa la ausencia de un valor o un valor nulo. Ej: se utiliza para indicar que una variable no tiene un valor asignado o que una función no devuelve ningún valor explícito.
- Lists: Las listas en Python son un tipo de dato que permite almacenar datos de cualquier tipo. Son mutables y dinámicas. Ej:



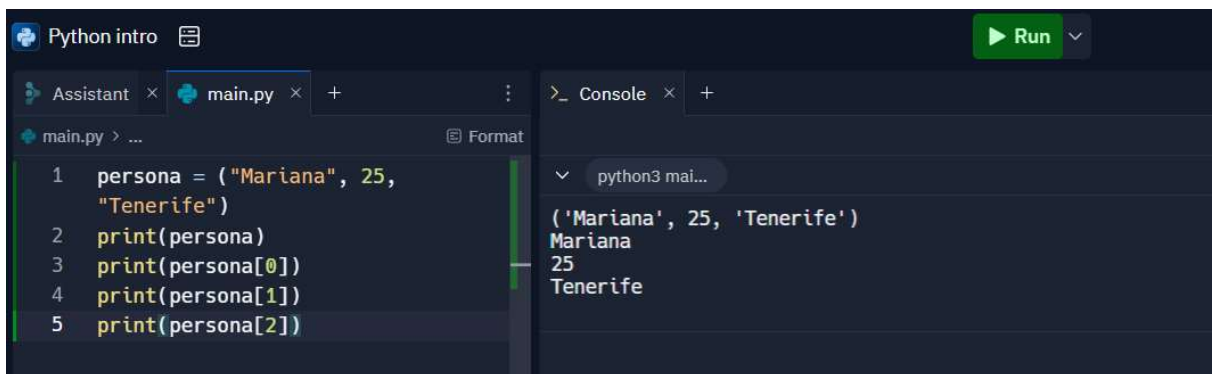
The screenshot shows a Python IDE with a file named 'main.py' containing the following code:

```
1 Compras = ["cebolla", "naranjas",
2            "arroz", "pollo", "manzanas",
3            "pan", "leche", "huevos"]
4 print(Compras)
```

The console output on the right shows the result of running the code:

```
python3 mai...
['cebolla', 'naranjas', 'arroz', 'pollo', 'manzanas', 'pan', 'leche', 'huevos']
```

- Tuples: es una colección ordenada e inmutable de elementos. Esto significa que, una vez creada, no puedes modificar los elementos de una tupla (no puedes agregar, eliminar o cambiar elementos) Ej:



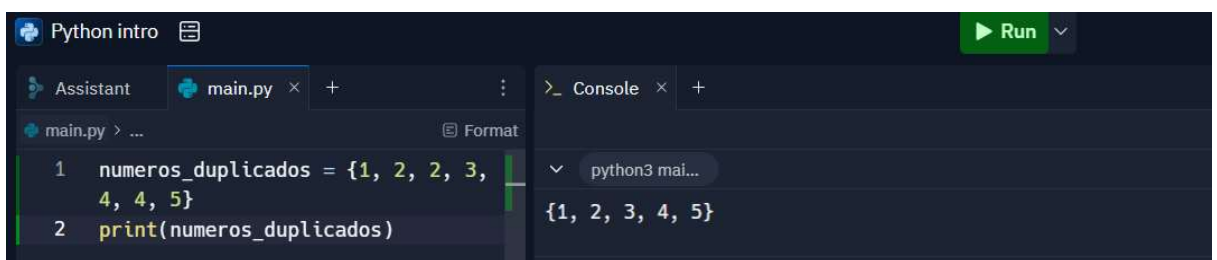
The screenshot shows a Python IDE with a file named 'main.py' containing the following code:

```
1 persona = ("Mariana", 25,
2            "Tenerife")
3 print(persona)
4 print(persona[0])
5 print(persona[1])
6 print(persona[2])
```

The console output on the right shows the result of running the code:

```
python3 mai...
('Mariana', 25, 'Tenerife')
Mariana
25
Tenerife
```

- Sets: es una colección no ordenada y mutable de elementos únicos. Los elementos de un set son único, lo que significa que no puede haber elementos duplicados. Ej:



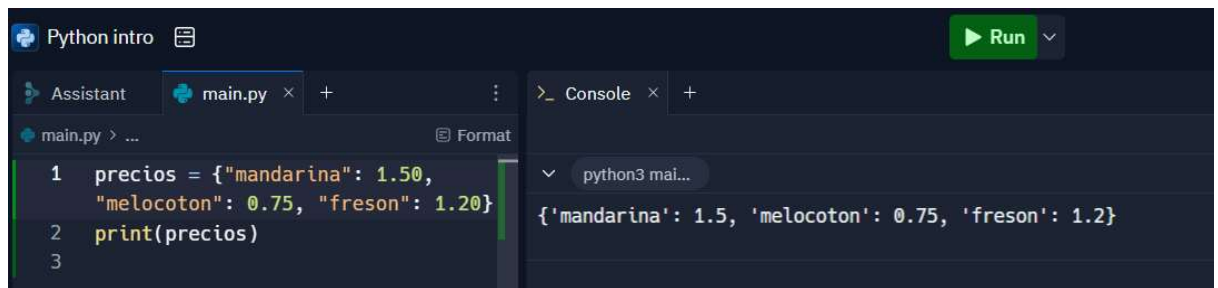
The screenshot shows a Python IDE with a file named 'main.py' containing the following code:

```
1 numeros_duplicados = {1, 2, 2, 3,
2                       4, 4, 5}
3 print(numeros_duplicados)
```

The console output on the right shows the result of running the code:

```
python3 mai...
{1, 2, 3, 4, 5}
```

- Dictionaries: es una colección no ordenada y mutable de pares clave-valor. Esto significa que almacena datos en forma de pares, donde cada par consta de una clave única y un valor asociado. Ej:



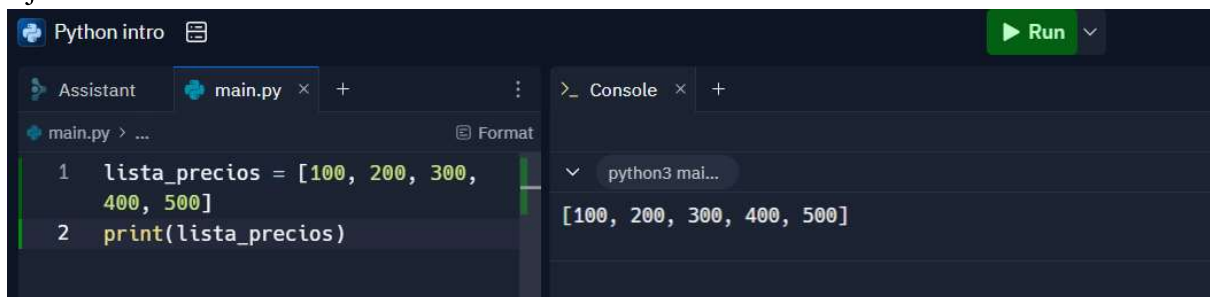
```
Python intro
Assistant
main.py x +
main.py > ...
1 precios = {"mandarina": 1.50,
2           "melocoton": 0.75, "freson": 1.20}
3 print(precios)

Console x +
python3 mai...
{'mandarina': 1.5, 'melocoton': 0.75, 'freson': 1.2}
```

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

Las variables se nombran en estilo `snake_case` : las palabras se escriben en minúsculas y se separan con el carácter de guión bajo `_`.

Ej:

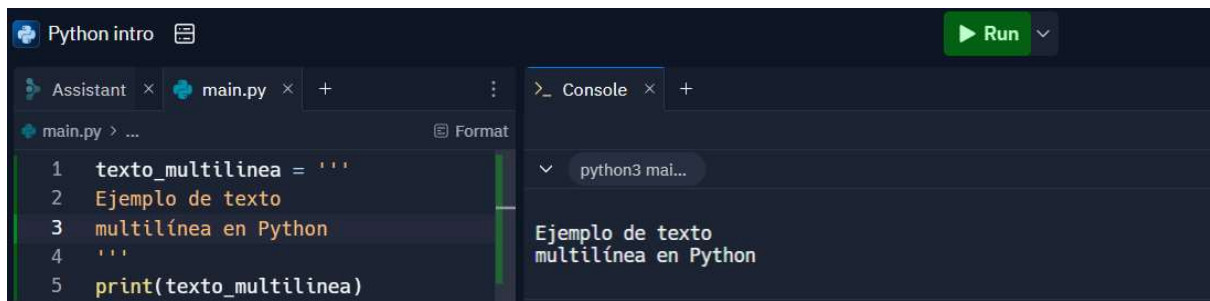


```
Python intro
Assistant
main.py x +
main.py > ...
1 lista_precios = [100, 200, 300,
2                 400, 500]
3 print(lista_precios)

Console x +
python3 mai...
[100, 200, 300, 400, 500]
```

¿Qué es un Heredoc en Python?

Es una forma de definir cadenas multilinea, se logra en Python utilizando cadenas de triple comillas. Ej:

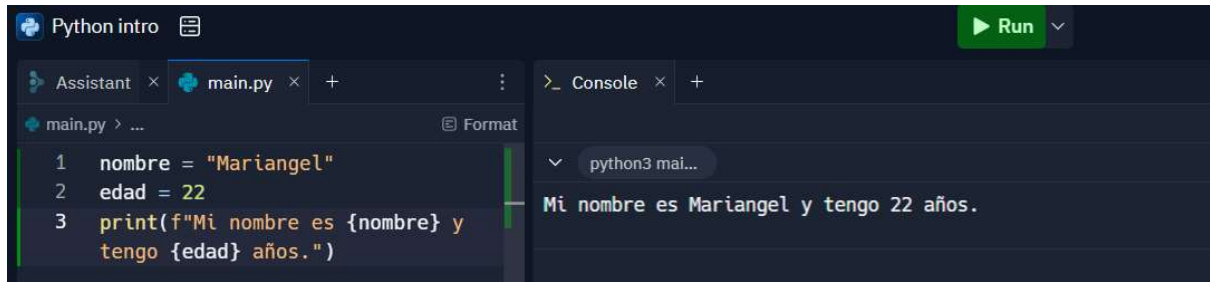


```
Python intro
Assistant
main.py x +
main.py > ...
1 texto_multilinea = '''
2 Ejemplo de texto
3 multilinea en Python
4 '''
5 print(texto_multilinea)

Console x +
python3 mai...
Ejemplo de texto
multilinea en Python
```

¿Qué es una interpolación de cadenas?

Es el proceso de insertar valores de variables o expresiones directamente dentro de una cadena de texto. Esto permite crear cadenas dinámicas que cambian según los valores de las variables. Ej:

A screenshot of a Python IDE interface. The top bar shows 'Python intro' and a 'Run' button. Below it, there's a tab for 'main.py'. The editor area shows three lines of Python code:

```
1 nombre = "Mariangel"
2 edad = 22
3 print(f"Mi nombre es {nombre} y
  tengo {edad} años.")
```

 To the right of the editor is a 'Console' tab. It shows the command 'python3 mai...' and the output 'Mi nombre es Mariangel y tengo 22 años.'

¿Cuándo deberíamos usar comentarios en Python?

Sirven para explicar a las personas que puedan leer el programa en el futuro, qué es lo que hace el programa, así como explicar algunas partes del código. Son beneficiosos específicamente cuando vamos a organizar el código y cuando usamos los comentarios automatizados como referencias.

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

En monolíticos todas las partes están funcionando siempre, ya que es un proceso que tiene todos sus módulos dentro, por lo que o toda la aplicación funciona o la aplicación no funciona. Mientras que en el caso de microservicios, al ser una arquitectura distribuida, los servicios están distribuidos en varias máquinas.