Python Functions: Built-in Functions and Multiple Return Statements: Takeaways

Syntax

- Initiate parameters with **default arguments**:

```
def add_value(x, constant=3.14):
    return x + constant
```

- Use **multiple return statements**:

```
def sum_or_difference(a, b, return_sum=True):
    if return_sum:
        return a + b
    else:
        return a - b
```

- Not using the else clause:

```
def sum_or_difference(a, b, return_sum=True):
    if return_sum:
        return a + b
    return a - b
```

Concepts

- We need to avoid using the name of a built-in function to name a function or a variable because this overwrites the built-in function. Also avoid naming variables using the names of the built-in functions because this also causes unwanted interference.
- Virtually every code editor highlights built-in functions.
- Each built-in function is well documented in the official Python **documentation**.
- It's possible to use **multiple** `return` **statements**. Combining `return` with an `if` statement and an `else` clause, for example.

Resources

- Python official documentation
- Style guide for Python code