Boolean operators

Week 3

Boolean operators

- Numbers have operators like: plus, minus, product, etc
- Boolean operators in Python are used to perform logical operations on Boolean values (`True` and `False`).
- Boolean operators are essential for controlling the flow of a program, especially in **conditional statements** and **loops**.
- Python has Boolean operators: `not`, `or`, `and`
- These operators return Boolean values (`bool`)

Boolean operators

OPERATOR	DESCRIPTION	SYNTAX
`and`	`True` if both the operands are `True`; `False` if at least other of the operands is `False`	`x and y`
`or`	`True` if at least other of the operands is `True`; `False` if both the operands are `False`;	`x or y`

`not`	`True` if the operand is `False`
	`False` if the operand is `True`
		`not x`

- Boolean operators can also be combined (like mathematical operators).

File: `Example3.py`

Short circuiting

- Short-circuiting in boolean operators is a concept where the evaluation of a logical expression stops as soon as the result is determined.
- This can improve efficiency and prevent unnecessary computations or function calls.
- `x or y`: If `x` is `True`, then `x or y` returns `True` and `y`'s value is not even evaluated
- `x and y`: If `x` is `False`, then `x and y` returns `False` and `y`'s value is not evaluated

Short circuiting

Example.

```
has_permissions = is_admin or is_moderator
```

Example.

```
x != 0 and 10 / x > 0.3
```

File: `Example4.py`