Boolean Values & Comparisons

`True` and `False` – the basis of all decisions.

Op	Meaning	Example	Result
`==`	Equal	`panels == 20`	`True`
`!=`	Not equal	`panels != 10`	`True`
`<`	Less than	`bill < 500`	`True`
`>`	Greater	`watts > 300`	`True`
`<=`	Less/eq	`size <= 6.6`	`True`
`>=`	Greater/eq	`bill >= 200`	`True`

if / elif / else

```
system_size = 6.6
if system_size <= 5:
    print("Small system")
elif system_size <= 10:
    print("Medium system")
else:
    print("Large system")
```

Only the **first** True branch runs. `else` catches the rest.

Logical Operators

Op	True when...	Example
`and`	Both True	`size <= 10 and owns_home`
`or`	At least one True	`bill < 200 or has_rebate`
`not`	Condition is False	`not has_solar`

```
if income < 180000 and owns_home and not has_solar:
    print("You qualify for the rebate!")
```

Nested Conditions

```
if system_size <= 10:
    if panel_quality == "premium":
        cost = cost * 1.3
```

Critical: Don't forget the colon `:` after `if`/`elif`/`else`/`while`/`for`!

Boolean Truth Tables

A	B	A and B
T	T	**T**
T	F	F
F	T	F
F	F	F

A	B	A or B
T	T	T
T	F	**T**
F	T	**T**
F	F	F

A	not A
T	F
F	T

for Loop + range()

Usage	Generates
`range(5)`	0, 1, 2, 3, 4
`range(1, 6)`	1, 2, 3, 4, 5
`range(0, 11, 2)`	0, 2, 4, 6, 8, 10

```
for year in range(1, 11):
    savings = savings * (1 + price_increase)
    total_savings = total_savings + savings
```

while Loop

Use when you **don't know** how many iterations.

```
remaining_cost = 5000
years = 0
while remaining_cost > 0:
    years = years + 1
    remaining_cost = remaining_cost - 1200
print(f"Payback in {years} years")
```

Input Validation Pattern

Set initial value to *fail* the condition so the loop enters at least once.

```
bill = -1                                    # Range check
while bill < 0 or bill > 5000:
    bill = float(input("Quarterly bill: "))

name = ""                                    # Not-empty check
while name == "":
    name = input("Panel name: ")

size = ""                                    # Membership check
while size not in ["small", "medium", "large"]:
    size = input("Size (small/medium/large): ")

panels = ""                                  # Integer check
while not panels.isdigit():
    panels = input("Number of panels: ")
panels = int(panels)
```

Accumulator Pattern

Running total inside a loop:

```
total_savings = 0
for bill in quarterly_bills:
    total_savings = total_savings + bill * 4
print(f"Total: ${total_savings}")
```

for vs while

Use **`for`**	Use **`while`**
Known iterations	Unknown iterations
Iterating a list	Input validation
Counting a range	Waiting for condition

Common Errors

`=` vs `==` – `=` assigns, `==` compares.
`if bill = 500` → SyntaxError! Use `==`.

Missing colon – `if bill > 500` → SyntaxError.
Must write `if bill > 500:` with the colon.

Infinite loop – forgetting to update the condition.
Always change the variable tested by `while`.

Wrong range bounds – `range(1, 5)` gives 1–4, not 1–5.
Use `range(1, 6)` to include 5.

`and` vs `or` in validation –
"reject if < 0 *or* > 100" → `while val < 0 or val > 100`
(not `and` – a value can't be both at once!)