

→ Python If... Else

Python supports the usual logical conditions from mathematics:

- Equals : $a == b$
- Not Equals : $a != b$
- Less than : $a < b$
- Less than or equal to : $a <= b$
- Greater than : $a > b$
- Greater than or equal to : $a >= b$

↳ IF

These conditions can be used in several ways, most commonly in "if statements" and loops.

Example:

a = 33

b = 200

if b > a:

print("b is greater than a")

output: b is greater than a

↳ Elif

The elif keyword is Python's way of saying "if the previous conditions were not true, then try this condition."

Example:

a = 33

b = 33

if b > a:

print("b is greater than a")

elif a == b:

print("a and b are equal")

output: a and b are equal.

↳ Else:

The else keyword catches anything which isn't caught by the preceding conditions

Example:

a = 200

b = 33

if b > a:

print("b is greater than a")

elif a == b:

print("a and b are equal")

else:

print("a is greater than b")

Output: a is greater than b

→ Python While Loops

With the while loop we can execute a set of statements as long as a condition is true

Example:

i = 1

while i < 6:

print(i)

i = i + 1

Output

1

2

3

4

5

↳ The break Statement

with the break statement we can stop the loop even if the while condition is true

Example:

Exit the loop when i is 3:

i = 1

while i < 6:

print(i)

if i == 3:

break

i = i + 1

Output: 1

2

3

↳ The continue Statement

with the continue statement we can stop the current iteration, and continue with the next

Example:

i = 0

while i < 6

i = i + 1

if i == 3:

continue

print(i)

Output: 1

2

4

5

6

→ Python For Loops

A for loop is used for iterating over a sequence (list, tuple, dictionary, set, string) with the for loop we can execute a set of statements, once for each item in a list.

Example:

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
    print(x) # Output: apple
              banana
              cherry
```

↳ The range() Function

- To loop through a set of code a specified number of times, we can use the range() function
- The range() function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and ends at a specified number

Example:

```
for x in range(6): # (6 is not include)
    print(x)
# Output: 0
          1
          2
          3
          4
          5
```

• we can change the starting value in a range

Example: $\text{range}(\text{start}, \text{end (not included)})$

for x in range (2, 6):

print(x) # output: 2

3

4

5

• we can also change the increment value by adding a third parameter

Example: $\text{range}(\text{start}, \text{end (not included)}, \text{increment})$

for x in range (2, 20, 3):

print(x)

output: 2

5

8

11

14

17