Repetition Structures
For-Loops and While-Loops
Loops

Introduce nonlinearity into programs
Repeatedly execute blocks of code
For-loops
While-loops
For-Loops

Repeats a block of code a specified number of times

Like printing a list of numbers

```python
for i in range(n):
    print(i)
```
While-Loops

While loops are more flexible, but also more error-prone

Must initialize variable and set increment

```python
i = 0
while i < 10:
    print(i)
    i = i + 1
```
Repetition Structures

For-Loops and While-Loops

↓

initialization_block

↓

Is condition true?

(yes) ↓

body_block

(no)↓

after_block

↓
String Formatting Expressions

Python defines the % operator to work on strings

When applied to strings, % provides a compact way to code multiple string substitutions

Operator on left is conversion target

On the right, provide the object you want Python to insert in its place

print('%d %s % (i, 'str'))

Most common: %s %d %f
Break Statement

Lets you jump out of a loop from within the loop body

Allows you to skip over unnecessary statements

break

Should only be used when it makes your code simpler

Related to: continue
Repetition Structures
For-Loops and While-Loops