

Python Flow Control Loops

Control the flow of a program

For Loop

We use a loop to iterate over a sequence of elements (list , tuple, string,..) .

Syntax :

```
for val in sequence:  
    Loop body / statements
```

The val is the variable that takes the value of the item inside the sequence on each iteration.

Range() function

The range function generates a sequence of numbers . It has a start , stop , and step size . To allow you to generate the wanted sequence of numbers.

Syntax:

range(start, stop,step_size)

Example:

```
for number in range(2, 10):  
    print ("I like this number: ", number)
```

While loop

We use the while loop to iterate over a block of code as long as the expression is true . We usually use it if we are not sure how many times we need to iterate over a block of code.

Syntax :

```
while test_expression:  
    statement / s
```

While loop with else

We can use an optional “else” of block of code , that runs when the while loop is finished (expression evaluated to false) .

Syntax:

```
while test_expression:  
    statement / s
```

```
else:  
    statement / s
```

Break & Continue

We use break & continue to alter the flow of a loop .

To exit (terminate) a loop , we use break

Syntax:

Break

To skip a block of code inside the loop and go to the next iteration , we use continue.

Syntax:

continue

Python pass

We use the pass statement as a placeholder for future implementation of code . To replace a body if code inside a function , a loop , a class etc.

Syntax:

pass