

Loops Cheat Sheet

Check out below for a run down of the syntax for while loops and for loops.

While Loops

A while loop executes the body of the loop while the condition remains True.

Syntax:

```
1 while condition:
2     body
```

Things to watch out for!

- **Failure to initialize variables.** Make sure all the variables used in the loop's condition are initialized before the loop.
- **Unintended infinite loops.** Make sure that the body of the loop modifies the variables used in the condition, so that the loop will eventually end **for all possible values of the variables.**

Typical use:

While loops are mostly used when there's an unknown number of operations to be performed, and a condition needs to be checked at each iteration.

For Loops

A for loop iterates over a sequence of elements, executing the body of the loop for each element in the sequence.

Syntax:

```
1 for variable in sequence
2     body
```

The range() function:

range() generates a sequence of integer numbers. It can take one, two, or three parameters:

- range(n): 0, 1, 2, ... n-1
- range(x,y): x, x+1, x+2, ... y-1
- range(p,q,r): p, p+r, p+2r, p+3r, ... q-1 (if it's a valid increment)

Common pitfalls:

- **Forgetting that the upper limit of a range() isn't included.**
- **Iterating over non-sequences.** Integer numbers aren't iterable. Strings are iterable letter by letter, but that might not be what you want.

Typical use:

For loops are mostly used when there's a pre-defined sequence or range of numbers to iterate.

Break & Continue

You can interrupt both while and for loops using the break keyword. We normally do this to interrupt a cycle due to a separate condition.

You can use the continue keyword to skip the current iteration and continue with the next one. This is typically used to jump ahead when some of the elements of the sequence aren't relevant.

If you want to learn more, check out this [wiki page on for loops](#).