

Loops

Have you ever had to do something more than once? Perhaps, clean your room, or, walk the dog. Wouldn't it be nice if you could do that one time and have it automatically done for you every time after that?

That's where loops come in. With a loop, you can write code one time, and then run that code over and over again.

Purpose:

Allow a program to run until a user is done with it. This may be a game, or a program that helps you do your homework.

References:

- Free online Editor and IDE: <https://repl.it/languages/python3>
- Free online help about Python: <https://www.w3schools.com/python/>
- These lessons:
<https://github.com/kaltinril/pythonTeaching/tree/master/assignments/4thgrade>

while loop

One of the basic types of loops is a `while` loop. This will run all the code inside it, until a condition is false.

A condition is a math statement that returns true or false (on or off, yes or no).

As an example, imagine you are told to read until you get to page 10.

In this situation, you would:

1. read page 1
2. Check if you are on page 10 yet, if not, keep reading

In code we would write something like this.

```
page = 1
while page < 10:
    print(page)
    page = page + 1
```

The above code does the following:

1. Create a new variable called `page` and set it to 1
2. Start a loop, and run the code in the loop, as long as page is less than 10
3. print the page number we are up to
4. increase page by 1. So the first time through the loop, page is 1, the second time, 2, etc.
5. Once page is 10, the `while page < 10` returns false, which exits the loop.

For loop

A `for` loop is similar to a `while` loop. The difference is with a for loop, you usually want the code inside it to run a specific number of times.

Lets re-write our while loop as a for loop:

```
for page in range(1, 10):
    print(page)
```

There are some nice things about using a `for` loop.

1. Notice we didn't have to create a page variable outside the loop.
2. We also didn't have to increase the page variable ourselves. The range is doing that for us automatically.

While vs for

When do you use a `while` loop, vs, when do you use a `for` loop?

- Generally, a for loop is used when you want to do something a certain number of times. A fixed amount that won't change.
- You would use a while loop if you are waiting for user input to exit, or, if the number of times you will run the loop is unknown.

- You can also use a while loop if you just want the program to run forever. `while true:`

What can you do with loops?

Here is a sample program that calculates perimeter for you. It uses both flavors of looping

`while` and `for`

```
keep_running = 'Y'
while keep_running == 'Y':
    sides = input("How many sides does your shape have? ")

    p = 0
    for side in range(0, int(sides)):
        s = input("Enter length of side " + str(side+1))
        p = p + int(s)

    print("The perimeter of your " + str(sides) + " sided shape is " + str(p))

    keep_running = input("Calculate another shape? [Y/N] ").upper()
```

NOTICE: There are 2 concepts used in the above code that have not been discussed yet.

1. `int(sides)`

- This is a way to convert the `Text` to a `number` so we can do math with it.

2. `str(shape)`

- This is a way to convert the `number` to `text` so we can print it.