Loo[illegible]

We found more than one version of Functions in a Nutshell. You can view the alternate here. ×

Looping is the repetition of statements in a program. We'll generally be using two types of statements in this python journey. They are:

1. While Loop(Indefinite Loop) Because the no of times it'll be executed till the condition is filled is unknown.
2. For loop(Definite Loop) Because the no of times it'll be executed till the condition is filled is known.

A for loop will execute once even before testing the condition but a while loop will only execute if the condition is true

An example of an infinite loop

```
x = 9
while x > 0
   print("Help! I can't stop!")
```

This program will continuously execute because the condition is being always true. 9 is always greater than 0, always therefore it will keep going, of course until x > 0 is false.

Break reserve word

A break takes you to the immediate next statement outside the loop. It's useful when you need to enter a sentinel value to stop the code from executing. A sentinel value is a value that stops a loop.

```
while True:
    goodperson = input("Enter Your Name:")
    if goodperson == "Salim":
        break
print("Welcome to Stein's Gate", goodperson )
```

Continue reserve word

A continue reserve word takes you back to the top of the loop. This is useful for conditionals that require you to head back to the beginning of the loop.

See Loops.Py for more info and examples.

Example of a for loop

```
AgesL = [1,2,3,44]
for ages in AgesL:
   print(age)
```

This will print every age on a separate line. The part where it says for ages in AgesL is referring to each age in the list. We know that we want to print age, for all the ages in AgesL, therfore making it a a perfect use of a finite loop.

When should I use a for or while Loop?

The best way to think about it is this: If you know you're dealing with a file or list, use a for loop. If you're dealing with conditions such as "The principle wants at least n number of people to..." or if there's a "continue

dealing with conditions such as "The principle wants at least n number of people to..." or if there's a "sentinel value

From a theoretical point of view, you use a for loop when you know exactly how many times you want the loop to run for, but not in a literal sense. For example, you know that you want to count the no of lines in a file, and that at some point it will end. Right? In that case use a for loop.

For a while loop, lets say an n no of Students input scores. You have no idea how many students will input their scores. It could be 1 million, 500 thousand, 2 people, etc. The fact that you don't know is enough reason to use a while loop. You'd use a sentinel value to stop the loop.