

More with Loops: While-loops

The other kind of loop is an *indefinite loop* or *while-loop*. This has format

```
while <condition>:  
    <body>
```

For example,

```
x = 0  
while x < 10:  
    x = x+1  
    print(x)
```

When a while-loop is executed, the body is evaluated over and over until the condition is **False**. If the condition never becomes False, the loop never terminates.

Here is a very common programming issue:

Enter data until some condition is met.

To make this simple, we will enter strings until we get the empty string:

```
done = False
```

```
while not done:
```

```
    myInput = input( "type something: " )
```

```
    if myInput == "":
```

```
        done = True
```

```
    else:
```

```
        print( "Hmmm. How interesting.")
```

We can determine if number n is prime by trying to divide all of the numbers from 2 up to (but not including) n into it. If any of them divide in evenly then n is not prime; if none of them do it is prime.

Here is an easy loop for this:

```
n = eval(input("Enter n: " ))
isPrime = True
for i in range(2,n):
    if n%i == 0:
        isPrime = False
if isPrime:
    print( "%d is prime." % n)
else:
    print( "%d is not prime." %n )
```

Note that the for-loop makes this program do a lot of useless checking. For example if n is 100 it divides 99 numbers into n , although it finds out at the start that 2 divides evenly into n . We can prevent this with a while loop:

```
isPrime = True
```

```
i = 2
```

```
while i < n and isPrime:
```

```
    if n%i == 0:
```

```
        isPrime = False
```

```
    i = i+1
```

Now use this to write a program that has a variable Max and prints all of the primes from 2 to Max.

Clicker Question:

Think about that beer song from Lab 2. A typical verse is

90 bottles of beer on the wall

90 bottles of beer!

Take one down, pass it around

89 bottles of beer on the wall.

Will you do this with a `WHILE`-loop or a `FOR`-loop?

- A) `WHILE` B) `FOR` C) Either D) Neither