

Common Issues using WHILE

The Condition

Two common errors that occur when people start to use the `while` statement in Python for the first time is that they either forget to add the colon (`:`) at the end of the statement or they capitalise the “w” in “while”.

WRONG CODE	REASON
<code>while (x > y)</code>	Missing the colon (<code>:</code>) at the end
<code>While (x > y):</code>	Capitalised the “W” in “while”

So, in practice, the `while` statement should look as follows:

```
while (x > y):
```

Loop Count

Sometimes people are trying to count from 1 to 5 in the loop and have the condition on the loop that causes it to loop either one too few times or one too many times. It's a good idea to take the loop out of a bigger program and into a simple loop like the one on Page 37. So assuming that we have a loop counter X that starts at 1, and is incremented in each iteration of the loop, either of the following statements work:

```
while (X < 6):
```

```
while (X <= 5):
```

Other Issues

If we declare a variable as lowercase (“x”) initially, then the computer won't recognise it if you change it to uppercase (“X”). Also, another issue is around indentation, make sure the body of the loop is indented from the “while”.

WRONG CODE	REASON
<pre>x = int(input()) y = int(input()) while (X > y):</pre>	<p>“x” needs to be the same case:</p> <pre>x = int(input()) y = int(input()) while (x > y):</pre>
<pre>while (x > y): print("X is", x) x = x + 1 # EndWhile;</pre>	<p>Need to indent the body:</p> <pre>while (x > y): print("X is", x) x = x + 1 # EndWhile;</pre>

Reflections

It takes a bit of practice to get used of the precision required for programming, so don't get discouraged by making small errors at the start of your journey; you are learning a key skill, and it's worth taking the time, and being patient along the way.

#PythonMonday © Damian Gordon