While Loops

Complete Python Bootcamp

While loops will continue to execute a block of code **while** some condition remains True.

For example, **while** my pool is not full, keep filling my pool with water.

Or **while** my dogs are still hungry, keep feeding my dogs.

Complete Python Bootcamp

- Syntax of a while loop

```
while some_boolean_condition:
    #do something
```

Complete Python Bootcamp

- You can combine with an else if you want

```
while some_boolean_condition:
    #do something
else:
    #do something different
```

Let's explore these concepts!

File Edit View Insert Cell Kernel Widgets Help Python 3

In [1]:
```
x = 0

while x < 5:
    print(f'The current value of x is {x}')
    x = x + 1
```

```
The current value of x is 0
The current value of x is 1
The current value of x is 2
The current value of x is 3
The current value of x is 4
```

In [ ]:

File Edit View Insert Cell Kernel Widgets Help Python 3

In [3]:
```
x = 0

while x < 5:
    print(f'The current value of x is {x}')

    x += 1
else:
    print("X IS NOT LESS THAN 5")
```

```
The current value of x is 0
The current value of x is 1
The current value of x is 2
The current value of x is 3
The current value of x is 4
X IS NOT LESS THAN 5
```

```
x += 1
```

break, continue, pass ¶

We can use break, continue, and pass statements in our loops to add additional functionality for various cases. The three statements are defined by:

```
break: Breaks out of the current closest enclosing loop.
continue: Goes to the top of the closest enclosing loop.
pass: Does nothing at all.
```

In [ ]:

break, continue, pass

We can use break, continue, and pass statements in our loops to add additional functionality for various cases. The three statements are defined by:

```
break: Breaks out of the current closest enclosing loop.
continue: Goes to the top of the closest enclosing loop.
pass: Does nothing at all.
```

In [6]:
```
x = [1,2,3]

for item in x:
    # comment
```

```
  File "<ipython-input-6-b7255e8c2a99>", line 4
    # comment
                ^
SyntaxError: unexpected EOF while parsing
```

In [ ]:

We can use break, continue, and pass statements in our loops to add additional functionality for various cases. The three statements are defined by:

```
break: Breaks out of the current closest enclosing loop.
continue: Goes to the top of the closest enclosing loop.
pass: Does nothing at all.
```

In [7]:
```
x = [1,2,3]

for item in x:
    # comment
    pass

print('end of my script')
```

```
end of my script
```

In [ ]:

We can use break, continue, and pass statements in our loops to add additional functionality for various cases. The three statements are defined by:

```
break: Breaks out of the current closest enclosing loop.
continue: Goes to the top of the closest enclosing loop.
pass: Does nothing at all.
```

In [10]:
```
mystring = 'Sammy'
```

In [11]:
```
for letter in mystring:
    print(letter)
```

```
S
a
m
m
y
```

In [ ]:

We can use break, continue, and pass statements in our loops to add additional functionality for various cases. The three statements are defined by:

```
break: Breaks out of the current closest enclosing loop.
continue: Goes to the top of the closest enclosing loop.
pass: Does nothing at all.
```

In [10]:
```
mystring = 'Sammy'
```

In [12]:
```
for letter in mystring:
    if letter == 'a':
        continue
    print(letter)
```

```
S
m
m
y
```

In [ ]:

File Edit View Insert Cell Kernel Widgets Help Python 3

p.
continue: Goes to the top of the closest enclosing loo
p.
pass: Does nothing at all.

In [10]:
```
mystring = 'Sammy'
```

In [13]:
```
for letter in mystring:
    if letter == 'a':
        break
    print(letter)
```

```
S
```

```
S
```

In [14]:
```
x = 0

while x < 5:
    print(x)
    x += 1
```

```
0
1
2
3
4
```

In [ ]:

File Edit View Insert Cell Kernel Widgets Help Python 3

```
S
```

In [15]:

```
x = 0

while x < 5:

    if x == 2:
        break
    print(x)
    x += 1
```

```
0
1
```

In [ ]: