Conditional Logic

Performing different actions depending on whether or not some expression is True or False

```
In [1]: a = 9
        b = 3
```

Conditional Operators

```
In [2]: a > b # a greater than b

Out[2]: True
```

```
In [3]: a < b # a less than b

Out[3]: False
```

```
In [4]: a >= b # a greater than or equal to b

Out[4]: True
```

```
In [5]: a <= b # a less than or equal to b

Out[5]: False
```

```
In [6]: a != b # a not equal to b

Out[6]: True
```

```
In [7]: a == b # a equal to b

Out[7]: False
```

Conditional operators can be used to compare strings as well

```
In [8]: "a" < "f"

Out[8]: True
```

```
In [9]: "money" == "power"

Out[9]: False
```

Logical Operators

The and Keyword: All statements must be True for the whole statement to be True

```
In [10]: True and True
Out[10]: True
```

```
In [11]: True and False
Out[11]: False
```

```
In [12]: False and True
Out[12]: False
```

```
In [13]: False and False
Out[13]: False
```

The or Keyword: Only one statement needs to be True for the whole statement to be True

```
In [14]: True or True
Out[14]: True
```

```
In [15]: True or False
Out[15]: True
```

```
In [16]: False or True
Out[16]: True
```

```
In [17]: False or False
Out[17]: False
```

The not Keyword reverses the truth

```
In [18]: not True
Out[18]: False
```

```
In [19]: not False
Out[19]: True
```

The if Statement

if statements run a section of code if the specified condition is met

In [20]:

```
grade = 88

if grade >= 60:
    print("You passed")

print("Thank you for attending")
```

```
You passed
Thank you for attending
```

The else Keyword runs a sectin of code if the specified condition in the if statement is False

In [21]:

```
grade = 88

if grade >= 60:
    print("You passed")
else:
    print("You failed")

print("Thank you for attending")
```

```
You passed
Thank you for attending
```

The elif keyword is used to add addtional conditions after an if statement

In [22]:

```
grade = 88

if grade >= 90:
    print("You score a top rank")
elif grade >= 60:
    print("You passed")

print("Thank you for attending")
```

```
You passed
Thank you for attending
```

Ending loops and loop iterations

The break keyword tells Python to end a loop, generally is a condition is met

In [ ]:

```

```

In [23]:

```
for i in range(0, 4):
    if i == 2:
        break
    print(i)
```

```
0
1
```

The continue keyword ends the current loops iteration

In [24]:

```
for i in range(0, 4):
    if i == 2:
        continue
    print(i)
```

```
0
1
3
```