Boolean Data Type & Comparison Operators

The next set of notewbooks discusses conditional execution. In the first notebook we will introduce the boolean (bool) data type. The bool data type allows only two values:

1. True
2. False

Note both values are case sensitive and should be written exactly as shown above. We also take a look at the comparison operators that can be used to compare arithmetic expressions with each other or string expressions with each other.

In [1]:

```
print(type(False))
print(type(True))
```

```
<class 'bool'>
<class 'bool'>
```

The following operators are called relational or comparison operators.

>   greater than

>=   greater than or equal to

<   less than

<=   less than or equal to

==   equal to -- note the two equal signs

!=   not equal to

These operators can be used to compare arithmetic or string expressions with each other.

In [2]:

```
'''
Comparing numeric values, numeric variables and / or numeric expressions
'''

x = 18
y = 20
print('x > y: {}'.format(x > y))
print('y > 15: {}'.format(y > 15))
print('x * y > 350: {}'.format(x*y > 350))
```

```
x > y: False
y > 15: True
x * y > 350: True
```

In [3]:

```
'''
Comparing string values and / or string variables.
'''
str1 = 'hello'
str2 = 'Hello'
str3 = 'hell'
str4 = 'hello'
str5 = 'k'
print('str1 == str2: {}'.format(str1 == str2)) #Comparisons are case sensitive as is demonstrated
by this example
print('str1 == str3: {}'.format(str1 == str3))
print('str1 == str4: {}'.format(str1 == str4))

print('str1 > str2: {}'.format(str1 > str2))
print('str1 > str3: {}'.format(str1 > str3))
print('str1 > str4: {}'.format(str1 > str4))
print('str1 > str5: {}'.format(str1 > str5))
```

```
str1 == str2: False
str1 == str3: False
str1 == str4: True
str1 > str2: True
str1 > str3: True
str1 > str4: False
str1 > str5: False
```