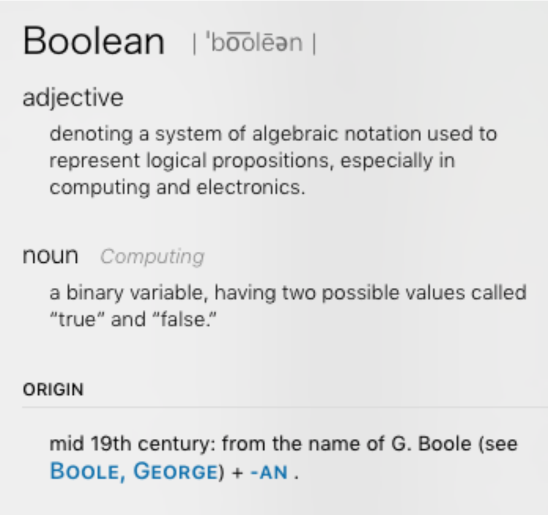
2.2a Boolean Variables and Comparisons

Boolean variables are either True or False.

We've seen these get returned when we did **comparisons**.



Boolean | ˈbo͞olēən |

adjective

denoting a system of algebraic notation used to represent logical propositions, especially in computing and electronics.

noun *Computing*

a binary variable, having two possible values called “true” and “false.”

ORIGIN

mid 19th century: from the name of G. Boole (see BOOLE, GEORGE) + -AN .

Operator	Meaning
==	equal to
!=	not equal to
<	less than
>	greater than
<=	Less than or equal to
>=	Greater than or equal to

Practice

1. What is the output from each of these?

a)
```
5 < 4
```

b)
```
x=6
x>=6
```

c)
```
z=10
z!=9
```

“If” statements only follow the instructions of the following code when the boolean returned is true. “Else” can be added on to run if the previous boolean had been false.

Note: The following two code blocks are equivalent.

```
x = 42
if x % 2 == 0:
  print('even')
else:
  print('odd')
```

```
x = 42
if x % 2 == 0: print('even')
else: print('odd')
```

2.2 Lists, Tuples, Sets, Dictionaries

A **list** is a collection which is ordered and changeable.

- Allows duplicate members.

```
thislist = ["apple", "banana", "cherry"]
print(thislist)
```

```
print(thislist[1])
```

```
thislist[1] = "blackcurrant"
print(thislist)
```

```
for x in thislist:
  print(x)
```

```
print(len(thislist))
```

```
thislist.append("orange")
print(thislist)
```

```
thislist.insert(1, "orange")
print(thislist)
```

```
thislist.remove("orange")
print(thislist)
```

Tuples are like lists, but cannot be changed.

```
a=(1,2,3)
```

Sets are an unordered collection of unique elements

```
x = set([3, 1, 2, 1])
```

Dictionaries are unordered, changeable and indexed.

```
rain_percent = { 1980: '17%', 1981: '15%', 1982: '10%'}
print(rain_percent)
print(rain_percent[1980])
```