In [ ]:

```
Booleans represent one of two values: True or False.

Boolean Values

In programming you often need to know if an expression is True or False.

You can evaluate any expression in Python, and get one of two answers, True or False.

When you compare two values, the expression is evaluated and Python returns the Boolean ans
```

In [1]:

```
print(10 > 9)
print(10 == 9)
print(10 < 9)
```

```
True
False
False
```

In [2]:

```
a = 200
b = 33
if b > a:
  print("b is greater than a")
else:
  print("b is not greater than a")
```

```
True
b is not greater than a
```

In [3]:

```
print(bool("Hello"))
print(bool(15))
```

```
True
True
```

In [4]:

```
x = "Hello"
y = 15

print(bool(x))
print(bool(y))
```

```
True
True
```

In [5]:

```
print(bool("abc"))
print(bool(123))
print(bool(["apple", "cherry", "banana"]))
```

```
True
True
True
```

In [6]:

```
print(bool(False))
print(bool(None))
print(bool(0))
print(bool(""))
print(bool(()))
print(bool([]))
print(bool({}))
```

```
False
False
False
False
False
False
False
```

In [7]:

```
class myclass():
  def __len__(self):
    return 0

myobj = myclass()
print(bool(myobj))
```

```
False
```

In [8]:

```
def myFunction() :
  return True

print(myFunction())
```

```
True
```

In [1]:

```
def myFunction() :
  return True

if myFunction():
  print("YES!")
else:
  print("NO!")
```

```
NO!
```

In [10]:

```
x = 200
print(isinstance(x, int))
```

```
True
```