Booleans and Logic

True, False, If, Else, Oh My!

Types of data

Data types

Three types of data we already know about:

“Hi!”	string
27	integer
15.238	float

Python can tell us about types using the `type()` function:

```
>>> type(“Hi!”)
<type ‘str’>
```

Can you get Python to output `int` and `float` types?

Data type: Booleans

Booleans

A boolean value can be: `True` or `False`

Is 1 equal to 1?

```
>>> 1 == 1
True
```

Is 15 less than 5?

```
>>> 15 < 5
False
```

Booleans

What happens when we type Boolean values in the interpreter?

```
>>> True
>>> False
```

When the words ‘True’ and ‘False’ begin with capital letters, Python knows to treat them like Booleans and not strings or integers.

```
>>> true
>>> false
>>> type(True)
>>> type(“True”)
```

Booleans: Comparisons

`and`

Both sides of the expression must be correct to be True.

```
>>> True and True
>>> False and False
>>> 1 == 1 and 2 == 2
>>> True and False
```

Booleans: Comparisons

or

One side of the expression must be correct to be True.

```
>>> True or True
>>> True or False
>>> 1 == 1 or 2 != 2
>>> False or False
```

Booleans: Reverse

not

True becomes False

False becomes True

```
>>> not 1 == 1
>>> not True
```

Boolean operators

Boolean operators:

==	Equal to
!=	Not equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

Boolean operators

Comparison practice:

```
>>> 5 < 4 + 3

>>> 12 + 1 >= 12

>>> 16 * 2 == 32

>>> 16 != 16

>>> 5 >= 6
```

Boolean operators

Comparison practice:

```
>>> 5 < 4 + 3
True
>>> 12 + 1 >= 12
True
>>> 16 * 2 == 32
True
>>> 16 != 16
False
>>> 5 >= 6
False
```

Booleans: Practice

Try some of these expressions in your interpreter:

```
>>> True and True
>>> False and True
>>> 1 == 1 and 2 == 1
>>>"test" == "test"
>>> 1 == 1 or 2 != 1
>>> True and 1 == 1
>>> False and 0 != 0
>>> True or 1 == 1
>>>"test" == "testing"
>>> 1 != 0 and 2 == 1
```

Logic

`if` Statements

`if` Statements

Making decisions:

> "**If** you're not busy, let's eat lunch now."
> "**If** the trash is full, go empty it."

If a condition is met, perform the action that follows:

```
>>> name = "Jennifer"
>>> if name == "Jennifer":
...     print "Hi Jennifer!"

Hi Jennifer!
```

`if` Statements

Adding more choices:

"**If** you're not busy, let's eat lunch now.
Or **else** we can eat in an hour."

"**If** there's mint ice cream, I'll have a scoop.
Or **else** I'll take butter pecan."

The else clause:

```
>>> if name == "Jennifer":
...     print "Hi Jennifer!"
... else:
...     print “Impostor!”
```

`if` Statements

Including many options:

"**If** you're not busy, let's eat lunch now.
Or **else if** Bob is free I will eat with Bob.
Or **else if** Judy's around we'll grab a bite.
Or **else** we can eat in an hour."

The elif clause:

```
>>> if name == "Jennifer":
...     print "Hi Jennifer!"
... elif name == “Margaret”:
...     print “Hi Margaret!”
... else:
...     print “Who are you?!?”
```

`if` Statements: Practice

Write an if statement that prints "Yay!" if the variable called color is equal to "yellow".

Add an elif clause and an else clause to print two different messages under other circumstances.

`if` Statements: Practice

Write an if statement that prints "Yay!" if the variable called color is equal to "yellow".

Add an elif clause and an else clause to print two different messages under other circumstances.

```
>>> color = "blue"
>>> if color == "yellow":
...     print "Yay!"
... elif color == "purple":
...     print "Try again"
... else:
...     print "We want yellow!"
```