Python 2.7: Boolean Logic

Lecture notes of Alexander Wood
awood@citytech.cuny.edu

New York City College of Technology

Boolean expressions

A **boolean expression** is an expression which is either true or false. The values `True` and `False` are special values of type `bool`.

```
>>> type(True)
<class 'bool'>
>>> type(False)
<class 'bool'>
>>>
```

Relational operators

There are a variety of relational operators, which tell us whether a relationship is True or False.

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

Relational operators

```
>>> 6 > 7
False
>>> 6 < 7
True
>>> 6 >= 7
False
>>> 6 <= 7
True
>>> 6 == 7
False
>>> 6 != 7
True
```

Relational Operators

Note that == is used to test equality. The single equal sign, =, is used to assign a value to a variable.

```
>>> x = 7
>>> x == 7
True
>>> x == 8
False
>>> x = 8
>>> x == 8
True
```

Logical Operators

The logical operators are `and`, `or`, and `not`.

- `x and y` is true if BOTH `x` and `y` are true
- `x or y` is true if EITHER `x` or `y` is true
- `not x` negates (ie, switches) the truth value of a boolean expression

Logical Operators

Remember to use parenthesis to avoid confusion about precedence!

```
x > 5 and x < 6
(x > 5) and (x < 6)
```

Logical Operators

```
>>> x = 5
>>> (x > 5) or (x < 10)
True
>>> (x > 5) and (x < 10)
False
>>> x == 5
True
>>> not (x == 5)
False
```

What is True?

What if we type the following:

```
bool("Hotline Bling")
```

Does this evaluate to `True`, or `False`? In other words, we are asking ourselves the big question: What does Python consider to be True?

What is False?

To determine what is `True`, we must first determine what is `False`. The following are all considered false:

```
bool:          False
null           None
integer:       0
float:         0.0
string:        ''
list:          []
tuple:         ()
dictionary:    {}
set:           set()
```

Note that we have NOT covered lists, tuples, dictionaries, or sets yet in this course.

What is True?

Everything which is not `False` is considered `True` in Python. This is counter-intuitive and takes some getting used to, but ends up being quite useful.

```
>>> bool("Hotline Bling")
True
>>> bool("")
False
>>> bool(" ")
True
>>> bool(3.14)
True
>>> bool(0)
False
>>> bool(0.00)
False
```

Testing for empty data structures

A common application of this 'truthiness' definition in Python is checking for empty data structures. See the example below, which checks for an empty string.

```
# this program determines whether a string is empty.

def main():
    some_string = ''

    if some_string:
        print 'there is something in the string!'

    else:
        print 'there is nothing here!'

main()
```

References

- http://www.openbookproject.net/thinkcs/python/english2e/ch04.html