Logical operations in Python

Numeric comparisons

There are eight comparison operations in Python. Comparisons can be chained arbitrarily; for example, `x < y <= z` is equivalent to `x < y and y <= z`.

Here is a summary of the Python comparison operators:

- `<`: strictly less than
- `<=`: less than or equal
- `>`: strictly greater than
- `>=`: greater than or equal
- `==`: equal
- `!=`: not equal

All of these operations return a boolean values (`True` or `False`):

```
5 > 2
```

```
2 != 3
```

```
3.0 <= 3.0
```

```
x = -1.0
0 <= x <= 1
```

```
42 == 42
```

Note: floating point numbers should never be tested for equality with `==`. Floating point numbers are approximations of real numbers, thus they should be tested for approximate equality:

```
.1 == .21 - .11
```

Why?

```
.21 - .11
```

A good way to do this is use `math.isclose` for Python versions 3.5 and above:

```
import math
math.isclose(.1,.21-.11)
```

Boolean operations

The Python boolean operations are:

- `x or y`: if `x` is false, then `y`, else `x`
- `x and y`: if `x` is false, then `x`, else `y`
- `not x`: if `x` is false, then `True`, else `False`

```
print(True or False)
print(False or False)
```

```
print(True and False)
print(True and True)
```

```
print(not True)
print(not False)
```

These logical operators are typically used to combine a comparisons:

```
a = 2
b = 3
print(a == 2 and b == 3)
print(a == 2 or b == 4)
```

```
x = 5
# test if x is not in range (0,10)
print(x <= 0 or x >= 10)
```