Logical Operators

Logical Operators

Logical operators act on booleans and return booleans. The logical operators are `and`, `or` and `not`.

Logical `and`

This acts on 2 booleans. It returns `True` if both booleans are `True` and `False` otherwise. For example:

[1]:
```
print('True and True is', True and True)
print('True and False is', True and False)
print('False and True is', False and True)
print('False and False is', False and False)
```

```
True and True is True
True and False is False
False and True is False
False and False is False
```

Logical `or`

This operator acts on 2 booleans. It returns `True` if at least one of the booleans is `True` and `False` if both booleans are `False`. For example:

[1]:
```
print('True or True is', True or True)
print('True or False is', True or False)
print('False or True is', False or True)
print('False or False is', False or False)
```

```
True or True is True
True or False is True
False or True is True
False or False is False
```

Logical `not`

This operator acts on a single boolean. It returns the opposite of the boolean:

[2]:
```
print('not True is', not True)
print('not False is', not False)
```

```
not True is False
not False is True
```

Combining Logical Operations

Although logical operations only act on up to 2 booleans at a time, just like arithmetic operators they can be combined in a single statement. For example:

[2]:
```
print('True and False or True is ', True and False or True)
print('True or True and False is', True or True and False)
print(not True or True and False)
```

```
True and False or True is  True
True or True and False is True
False
```

Although it isn't important for the cases above, if you need to ensure a specific order for the operations you can use brackets to group them.