CIS 122

Booleans Continued

Conditional logic recap

```
def abs(x):
    if x < 0:
        return -x
    elif x > 0:
        return x
    else:
        return 0

>>> abs(-42)
42

>>> abs(0)
0
```

Print vs Return

```
def even(x):
    if x % 2 == 0:
        return True
    else:
        return False
```

```
def even(x):
    if x % 2 == 0:
        print True
    else:
        print  False
```

Print vs Return

```
def even(x):
    if x % 2 == 0:
        return True
    else:
        return False
```

```
def even(x):
    if x % 2 == 0:
        print True
    else:
        print  False
```

```
if even(6):
    print "6 is even"
else:
    print "6 is odd"
```

Print vs Return

- Functions which print values are useful only to the user
- Functions which return values can be used as building blocks in other functions
- When should you print?
    - When you want to convey information
    - Interacting with the user
    - Useful for debugging code
- When should you return?
    - When you want to use your function in a larger context
    - Most of the time

A Conditional Shortcut

```
def even(x):
    if x % 2 == 0:        → Evaluates to True or False
        return True
    else:
        return False
```

A Conditional Shortcut

```
def even(x):
    if x % 2 == 0:
        return True
    else:
        return False
```

```
def even(x):
    return (x % 2 == 0)
```

Logical Connectives

- What can we do with booleans?
  - Combine them
- Logical Connectives
  - and
  - or
  - not

Logical Connectives - and

- When is **a and b** true?
  - When both a and b are true

```
>>> True and True
True

>>> True and False
False

>>> False and False
False
```

a	**b**	**a and b**
True	True	True
True	False	False
False	True	False
False	False	False

Logical Connectives - or

- When is **a or b** true?
    - When a is true or b is true (or both)

```
>>> True or True
True

>>> True or False
True

>>> False or False
False
```

a	**b**	**a and b**
True	True	True
True	False	True
False	True	True
False	False	False

Logical Connectives - or

- When is **not a** true?
    - When a is false

```
>>> not True
False

>>> not False
True
```

a	**not a**
True	False
False	True

Logical Connectives Quiz

- 1 < 2 and 2 < 3
- 10 > 100 or 'a' == 'a'
- (not not True)
- (True and False) or (not 7 != 8)
- (5 <= 5) and (not 'red' == 'blue') and ('a' >= 0 or 'a' <= 0)
- BONUS: What does this code do?
    - x = (x == False)   (assume x is a defined boolean var)

What's so great about booleans?

- What can we use as a condition?

- Boolean values
  - False
  - True

- Expressions that evaluate to booleans
  - 2 < 1
  - True or False

- Values that could be interpreted as booleans
  - 0
  - Any number other than 0

What's so great about booleans?

- "Empty" values are interpreted as False
    - 0
    - 0.0
    - "" (the empty string)

- Everything else is interpreted as True
    - -7
    - 0.1
    - " " (the space character)