ECS 10
9/30

Announcements

Assignment due Tuesday night.

Nested if statements

```python
if today == "y":
    if yesterday == "y":
        print "Doin' good!"
    else:
        print "Try harder!"
else:
    print "Try harder!"
```

Re-assigning variable

```python
x = 5
x = x+1
```

5

Re-assigning variable

```python
x = 5
x = x+1
```

5+1

Re-assigning variable

```python
x = 5
x = x+1
```

6
Getting random numbers

```
from random import randrange
choice = randrange(0, 20)
```
- `randrange(0, 20)` produces a random integer value >= 0 and < 20.
- We import it from the random module, which was installed with Python.

Modules
- A module is a collection of additions to the language that handle specialized data or problems.
- There are many modules that come with the Python installation, and many, many others that are available over the Web.
- To use a module, you need to import it, at the top of your program.
- Imports should be the first lines of the program.

Integers vs strings
- 5
- 5 is an integer.
- ‘5’ is a string.
- Fiona is a dog.
- ‘Fiona’ is a string.

Converting between data types
- `x = int("2")`
- int() converts the string “2” to the integer 2
- Can take a string or a float as input.
- The data value it produces is an integer.

<, >, <=, >=
- More Boolean operators.
- Only work with numbers.

Three possible outcomes
```
guess = int(inStr)
if guess < choice:
    print("No, too small.")
elif guess > choice:
    print("No, too big.")
else:
    print("You got it!")
```
Checking if we can convert

```python
from inputCheck import canBeInt
...
if canBeInt(inStr):
    celsius = int(inStr)
```

- `canBeInt()` is a function that produces a Boolean value (True or False).
- If the `inStr` can safely be converted to an integer, then `canBeInt(inStr)` produces the value True.
- Otherwise, DON'T try to convert `inStr` to a number; it will crash the program!

Not!

- The `not` command changes True to False and False to True

```python
if not canBeInt(guess):
    print("That is not a valid guess.")
```

Flow Chart

```
canBeInt? T F
  convert to
  int msg msg
  guess < choice T F msg
  guess > choice T F msg
```

and and or

- `and` and `or` are operators on Boolean data
  ```
  >>> 5+6
  11
  >>> (2<3) and (1<3)
  True
  >>> (2<3) or (1<3)
  True
  ```
- The value of a Boolean operation is Boolean (True or False)

or

```python
>>> True or True
True
>>> True or False
True
>>> False or True
True
>>> False or False
False
```

Memorize this!