Announcements

- I will be gone Wds and Fri. Prof. Gusfield and Balaji will guest lecture.
- I will be in lab hours tomorrow; no office hour Wds.
- Assignment due tomorrow night.
- Next assignment due Thurs 14th.
- Discussion sections next week will be midterm review.

**Pick a number**

```python
from random import randrange
from inputCheck import canBeInt

choice = randrange(1,11) # random integer
print("I'm thinking of a number between 1 and 10.")
inStr = input("What's your guess? ") #string
if not canBeInt(inStr):
    print("That's not a guess I recognize."")
else:
    guess = int(inStr) # integer: the user's guess
    if guess < choice:
        print("No, too small."")
    elif guess > choice:
        print("No, too big."")
    else:
        print("You got it!")
        print("My number was", choice)
        input("Press enter to exit.")
```

**Flow chart**

- Draw a diagram of what program is doing.
- Main advantage: stop and think with a piece of paper when you are stuck.

**Flow Chart**

- If today == "y":
  - If yesterday == "y":
    - Print "Doin' good!"
  - Else:
    - Print "Try harder!"
- Else:
  - Print "Try harder!"

**Nested if statements**
Boolean expressions

- Values of Boolean expressions are either True or False
- Boolean is a data type.
- We now have four data types (and four kinds of expressions):
  - integer, floating point, string and Boolean.

```
 today == "y"
```

- Have value True or False
- Examples:
  - today == "y"
  - x < 30
  - 2+2 == 4
  - 2+2 <= 5
  - "cat" != "Cat"

Boolean algebra

- Named after George Boole (1815-1864)
- Main idea: you can write down logic as mathematical formulas.
- His book: *An Investigation into the Laws of Thought*
- Computers do logical as well as numerical and string computation.

True and False

- These are the only two possible Boolean data items.
- NOT descriptions of expressions; actual data.
- Maybe better names would be Truth and Falsehood, so we think of them as things.
- We can store True and False in variables, just like other data.

and and or

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

or

- ```
  >>> True or True
  True
  >>> True or False
  True
  >>> False or True
  True
  >>> False or False
  False
  ```
- Memorize this!
Booleans simplify this program

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

Using Booleans

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

for loop

```python
for count in range(3):
    print(count)
```

- Standard way to do something a fixed number of times.
- This program prints:
  - 0
  - 1
  - 2

For loop

```python
for i in range(3):
    print("Rabbit!")
```

- Don't need to use the new variable in the block under the for statement.
- Next time: use it to give user three guesses.
- Often the new variable is called `i`