Objectives
- Code Readability
- Intro to conditional statements
- `sys` module
- Broader Issue: 4 Puzzles from Cyberspace

What Does This Program Do?
```python
import random
picknum=""
for x in xrange(3):
    numChosen = random.randint(0,9)
    picknum += str(numChosen) + "-"

numChosen = random.randint(0,9)
picknum += str(numChosen)
print "The number is", picknum
```

VA Lottery: Pick 4
- To play: you pick 4 numbers between 0 and 9
- To win: select the numbers that are selected by the magic ping-pong ball machine
- Your job: Simulate the magic ping-pong ball machines
  - Revision: display number as #-#-#-#

Changes to pick4.py
- Comments
  - Clarify what the program is doing
  - We wrote the program Wednesday
    - Already unclear on the details
- Constants
  - Give meaning to “magic numbers”
    - What were 0, 9, 3?

VA Lottery: Mega Millions
- Modify your Pick 4 to simulate Mega Millions
- To play: you pick 5 numbers between 1 and 56
  - Ignoring rule: 1 Mega Ball number between 1 and 46
- Your job: Simulate the result of the magic ping-pong ball machines, displayed as #-#-#-#
- How difficult to modify the last program?
- What could we do to make easier?

Improving Code Readability
- Comments
  - Describe blocks of code at a high level
- Output/Display
  - Descriptive, explains what program outputs
- Constants
  - Change one value (at top of program) to change value everywhere in program
  - Flexible programs
  - Gets rid of “magic numbers”
    - Give a clear name/purpose to values
What does this program do?
- How would you figure it out?
- What would you do to improve the program’s readability and usability?

Examples:
- If the PB is new (has a safety seal)
  - Then, I will take off the safety seal
- If it is raining and it is cold
  - Then, I will wear a raincoat
- If it is Saturday or it is Sunday
  - Then, I will wake up at 10 a.m.
  - Otherwise, I wake up at 7 a.m.
- If the shirt is purple or the shirt is on sale and blue
  - Then, I will buy the shirt

Sometimes, we do things only if some other condition holds (i.e. "is true")

Examples:
- Finding the absolute value of a number
  - \(|4| = 4\)
  - \(|-10| = 10\)
- To get the answer, we want to multiply the number by -1 only if it’s a negative number
- Code:
  ```python
  if x < 0:
    abs = x*-1
  ```

So far, we’ve thought of programs as a sequence of statements. Statements execute in order.
if Statements

- Change the control flow of the program

```
Fahr = input("...")

Celsius = (5/9.0)*(Fahr-32)

print "Celsius", Celsius
```

Other "things" that change control flow

- Function calls
  - "Go execute some other code and then come back with the result"

```
abs = x * -1
abs = x
```

Syntax of if statement: Simple Decision

```
if condition:
    statement1
    statement2
    ...
    statementn
```

Other "things" that change control flow

- for loops
  - Repeats a loop body a fixed number of times before going to the next statement after the for loop

```
for x in xrange(10):
    print "Hello"
    print "Goodbye"
```

Conditions

- Syntax:
  - `<expr> <relational_operator> <expr>`
- Evaluates to either True or False
- Boolean type

Relational Operators

- Syntax:
  - `<expr> <relational_operator> <expr>`

<table>
<thead>
<tr>
<th>Relational Operator</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;=</code></td>
<td>Less than or equal to?</td>
</tr>
<tr>
<td><code>&gt;=</code></td>
<td>Greater than or equal to?</td>
</tr>
<tr>
<td><code>==</code></td>
<td>Equals?</td>
</tr>
<tr>
<td><code>!=</code></td>
<td>Not equals?</td>
</tr>
</tbody>
</table>
Examples: Using Conditionals

- Determine if a number is even or odd

```python
x = input("Enter a number: ")
remainder = x%2
if remainder == 0 :
    print x, "is even"
else:
    print x, "is odd"
```

Assignment Operator and the Equality Operator

- Assignment operator: =
- Equality operator: ==

```python
x = input("Enter a number: ")
remainder = x%2
if remainder = 0 :
    print x, "is even."
```

Syntax of if statement: Two-Way Decision

```python
if condition :
    statement1
    statement2
    ...
else :
    statement1
    statement2
    ...
```

If-Else statements (absolute values)

```python
if x < 0 :
    x *= -1
    print "x=", x
else:
    abs = x
    print "abs=", abs
```

Examples: Using Conditionals

- Determine if a number is even or odd
- More efficient implementation

- Don’t need to check if remainder is 1 because if it’s not 0, it must be 1

```python
x = input("Enter a number: ")
remainder = x % 2
if remainder == 0 :
    print x, "is even"
else:
    print x, "is odd"
```

Practice: Draw the Flow Chart

```python
print "This program determines your birth year" print "given your age and current year"
age = input("Enter your age >> ")
if age > 110:
    print "Don’t be ridiculous, you can’t be that old."
else:
currentYear = input("Enter the current year >> ")
birthyear = currentYear - age
print "You were either born in", birthyear, "or",
print birthyear-1
```
sys module

- Has useful “system” functions
- Use the exit([status]) function
  - Exits the whole program
  - If status is empty, defaults to 0
  - Status of 0 means success
  - Other values are various failures
- Another example of changing control flow

Example Use of sys module

```python
import sys
print "This program determines your birth year"
print "given your age and current year"
print
age = input("Enter your age >> ")
if age > 110:
    print "Don't be ridiculous, you can't be that old."
    sys.exit(1)
# input is reasonable ...
currentYear = input("Enter the current year >> ")
birthyear = currentYear - age
print "You were either born in", birthyear, "or",
print birthyear-1
```