Today's "Menu"

(week 4)

- Announcements
- Decision structures (if-/elif/else)
- Booleans
- A few internet/networking 'tricks'
- How to 'break up' a program using functions
- How to write a program?
- To write a program!
- Questions/Concerns?

Announcements

- Exam starts this week – Wednesday through Tuesday
- Windows calc has binary (hex/octal) mode
- Question: Best communication mode? Which?
  - E-mail?
  - Class web page?
  - myFranklin announcements?
- "Turing Test of Intelligence" and "smart birds"

Misc

NOTE

```
IDLE 2.6.1
>>> 'hello' * 5
'hellohellohellohellohello'

>>> '77' * 3
'777777'
>>>
```

However

```
>>> 77 * 3
231
```

PAPERS: Unless you are expert in the field of AI, your "reasoned" thoughts/opinions will benefit from references to as a basis for your statements/conclusions.

Functions

- How to break up a program into functions?
- Every 20 lines?

Functions

- Distinct task (encapsulate one task - abstraction)
- Repetitive tasks (code re-use)
- *Test*: Name of method should reflect method.
  - get_user_input_and_cacluates_total_plus_tax_display_bill()? Hmpf!

  Or?
  - get_user_Input()
  - calculate_Tax()
  - calculate_Total()
  - display_Bill()

Global vs.Local Variables

- Local variables are "local" to the body of a function.
- Local variables exist only within the block where they are defined, they are not accessible elsewhere.
- Global variables are visible everywhere.
- Global variables should be AVOIDED!!
- See section 3.4-3.6 of text and sample program: `loc_glob_vars.py`

if-Statement (1)

- Allows us to break out of simple sequential execution
- We can make decisions based on user input
- Essential for error handling and algorithmic control

if-Statement (2)

if <Boolean expr is true>:
execute "body" of if-statement
could one or more statements

also

```
if <Boolean expr>:
   ....
else:
   ....
```

```
if <Boolean expr>:
   ....
elif <Boolean expr>:
      ....
elif <Boolean expr>:
   ....
else: # note, no Boolean expr here
   ....
```

Quick Example

```
.
.

number = ???
                    ← always have a blank line before
if number >= 0:
   print 'absolute value of number is ', number
else:
   print 'absolute value of number is ', -number
                    ← always have a blank line after for readability!
.
.
.
```


Quick Example

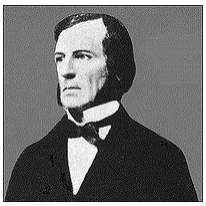
```
from random import randint

op1 = randint(0, 3)

if op1 == 0:
   print ' + ',
elif op1 == 1:
   print ' - ',
elif op1 == 2:
   print ' * ',
else:
   print ' / ',
```

http://cs.franklin.edu/~esmail/COMP_480/SampleProgs/rand_Expression1.py

Boolean Logic
It's all about Truth



George Boole (2 November 1815 – 8 December 1864) was an English mathematician and philosopher.

As the inventor of Boolean logic, which is the basis of modern digital computer logic, Boole is regarded in hindsight as one of the founders of the field of computer science.

http://en.wikipedia.org/wiki/George_Boole

Boolean AND

- True – if *all* parts are true

if smart *and* handsome: # want both

....

Truth Table:

AND	True	False
True		
False		

Boolean OR

- True – if *at least one* part is true

if smart *or* handsome: # will settle for one ☺

….

Truth Table:

OR	True	False
True		
False		

Compound and Relational Expressions

- Boolean expressions always evaluate to TRUE or FALSE
- Can consist of one or many expressions
- Relational expression use relational operators: >, <, <=, >=, !, ==
- Assignment != Boolean comparison
- Relational expressions yield TRUE or FALSE too