Python Basics!

branched control, range, lists

CS101 Lecture #8
Homework #4 is due Friday Sep. 23.
Midterm #1 will be Monday Oct. 3. (evening)
Warmup Quiz
s = 'ABCDEFGH'
t = ''
i = 0
while i < 8:
    t = t + s[i+1]
    i += 2

What is the final value of t?
A "ACEG"
B "BDFH" ★
C "ABCDEF"
D "ABEF"
s = '0123456789'
t = 

i = 0
while i < 5:
    if (i%2) == 1:
        t = t + s[i-1]
    if (i%2) == 0:
        t = t + s[i+1]
i = i + 1

What is the final value of t?
A "92143"
B "103254"
C "10325" *
D "921436"
E None (loop doesn’t terminate)
z = [ 1.2, 0.6, 0.5, 0.3 ]
z = z.sort()

What is the final value of z[1]?
A 0.6
B 0.5
C None ★
D None of the above.
What are two changes this code needs to be executable?

```python
if x < 1.5:
    x = x + 1
if x == (1.5 or 2.0):
    x = x - 1
```
What are two changes this code needs to be executable?

```python
if x < 1.5:
    x = x + 1
if x == 1.5 or x == 2.0:
    x = x - 1
```
Go to office hours!
I’ll hw03 move to basis out of 70 (instead of 74).
Will be some opportunity for extra credit later.
Clunker Motors Inc. is recalling all vehicles in its Extravagant line from model years 1999-2002 as well all vehicles in its Guzzler line from model years 2004-2007.

Given variables `modelYear` and `modelName` write a statement that assigns `True` to `recalled` if the values of `modelYear` and `modelName` match the recall details and assigns `False` otherwise.
Clunker Motors Inc. is recalling all vehicles in its Extravagant line from model years 1999-2002. Given a variable `modelYear` and a string `modelName` write a statement that prints the message ”RECALL” to standard output if the values of `modelYear` and `modelName` match the recall details.
Assume that $x$ is a string variable which has been given a value. Write an expression whose value is True if and only if $x$ is alphanumerics, that is, either a letter or a decimal digit.
ans = input( "Enter a number:" )
if float(ans) < 0:
    print( "The number was negative." )
Control flow represents actual sequence of lines executed by processor.

Conditional execution lets you execute (or not) a block of code based on logical comparison.
We often need to make decisions with several options.

Branched conditional execution lets you execute one of several blocks of code.
def absolute(x):
    if x >= 0:
        return x
    else:
        return -x
We create an if/else statement as follows:

- the keyword if
- a logical comparison (results in bool)
- a block of code
- the keyword else
- a different block of code
These produce Boolean output.

- **in**  
  Is one string inside of the other?

- **not in**  
  Is one string not inside of the other?
def fun(s):
    return s.isalpha() and 'a' in s

x = fun( "sam" ) and fun( "AS" )

What is the value of x?
A True
B False
Sometimes we need to make more than one decision.

We can nest blocks.

```python
word = input( 'Enter a Scrabble word: ' )
if not word.isalpha():
    print( 'There are only letters in Scrabble!' )
else:
    if not word.isupper(): # why not 'word.islower()'
        word = word.upper()
    print( 'You entered %s.' % word )
```
Nesting

```
not alphabetical?
  True
    "There are only letters in Scrabble."
  False
    not upper-case?
      True
        .upper()
      False
        "You entered %s."
```
Exercise: Nesting

```
positive?
  +-----------------+-----------------+
  | True            | False           |
  +-----------------+-----------------+
  | even?           | even?           |
  +-----------------+-----------------+
  | True            | False           |
  +-----------------+-----------------+
    x               x + 1
  +-----------------+-----------------+
  | -x              | -x + 1          |
  +-----------------+-----------------+
```

Conditional Execution
def evenpos(x):
    if x >= 0:
        if (x%2) == 0:
            return x
        else:
            return x + 1
    else:
        if (x%2) == 0:
            return -x
        else:
            return (-x) + 1
Sometimes we need to select among many choices.
Example

```python
if day == 1:
    print("Sunday")
else:
    if day == 2:
        print("Monday")
    else:
        if day == 3:
            print("Tuesday")
        else:
            if day == 4:
                print("Wednesday")
            else:
                if day == 5:
                    print("Thursday")
                else:
                    if day == 6:
                        print("Friday")
                    else:
                        if day == 7:
                            print("Saturday")
```
if day == 1:
    print("Sunday")
elif day == 2:
    print("Monday")
elif day == 3:
    print("Tuesday")
elif day == 4:
    print("Wednesday")
elif day == 5:
    print("Thursday")
elif day == 6:
    print("Friday")
elif day == 7:
    print("Saturday")
else:
    print("That is not a valid day.")
We create an \texttt{if/elif/else} statement as follows:

- the keyword \texttt{if}
- a logical comparison (results in \texttt{bool})
- a \texttt{block} of code
- the keyword \texttt{elif}
- a logical comparison (results in \texttt{bool})
- a \texttt{block} of code
- the keyword \texttt{else}
- a different \texttt{block} of code
Iteration Redux
```python
colors = [ 'red', 'yellow', 'blue', 'jale', 'ulfire' ]
for color in colors:
    print( color )
```
A for loop requires:
- the keyword `for`
- a loop variable
- the keyword `in`
- a set of values
- a block of code

For loops iterate over iterable types one at a time.
s = 'abcdefg'
t = ''
for c in s:
    t = c + t

What is the value of t?
A  'abcdefg'
B  'gfedcba'
C  'a'
D  'g'
Write a function to sum all of the digits in a number. I.e.,

\[ 12145 \rightarrow 1 + 2 + 1 + 4 + 5 \rightarrow 13 \]
```python
def sum_digits(n):
    result = 0
    for letter in str(n):
        result += int(letter)
    return result
```
for i in range(10):
    print(i ** 2)
The `range` function returns an `iterator` containing integers.

`range` can be cast as a list.

Two arguments:
- (optional) the starting value of the range (inclusive)
- the ending value of the range (exclusive)
Reminders

- Homework #4 is due Friday Sep. 23.
- Midterm #1 will be Monday Oct. 3. (evening)