Python Basics!

scripting, logic, control

CS101 Lecture #6
Administrivia
Homework #2 is due Wed Oct. 19.
Homework #3 is due Wed Oct. 26.
Warmup Question
(No Quiz this lecture!)
s = "74.125.21.147"
i = s.find( "." )
x = s[i+1:i+3]
x = x * 2

What is the value of $x$?
A "125125"
B 250
C "1212"
D 24
Composing Functions
def pow(a, b):
    y = a ** b
    return y
We define a function with the following:
- the keyword `def`
- the name of the function
- a pair of parentheses
- a `block` of code
Example: Defining functions

def greetings():
    print("Bom dia!")
    print("Bonjour!")
    print("Ni hao!")
    print("Hello!")
    print("Shalom!")
    print("Guten tag!")
    print("Konichiwa!")
    print("As-salamu alaykum!")
A section of code grouped together.
- Begins with a `:`.
- Contents of the block are indented:
```python
def hello():
    print('hello')
```
Scope

- Variables defined inside of a block are *independent* of variables outside of the block.
- Variables inside a block *do not exist* outside of the block.
- Blocks are isolated from the rest of the code!
```python
a = 5
def fun():
    a = 3
    b = 4
    a = a + b
fun()
print(a)
```
Functions can return values with the keyword `return`.
```python
def three():
    return 3

def zero():
    return 0
print('0')
```

Does the above code face an error? Does the print statement take effect if invoking the function?
Functions can accept values as parameters (input, arguments).
These variables are declared in the function header.
Multiple parameters are separated by commas.

```python
def print_message( msg ):
    print( msg )
```
def fun(a):
    return a+2

x = fun(2) * fun(3)

What is the value of $x$?

A 6
B 8
C 24
D None of the above.
def fun(m):
    return m.title().swapcase()

x = fun( "abb") + fun( "acab" )

What is the value of $x$?

A 'AbbAcab'
B 'aBBaCAB'
C 'abbacab'
D 'ABBACAB'
def fun(a,b):
    c = ((a + ' ') * len(b)).title()

x = fun( "ab", "caa" )

What is the value of \( x \)?

A 'ab ab ab'
B 'Ab Ab Ab'
C 'AB AB AB'
D None of the above.
def fun(a,b):
    c = ((a + ' ') * len(b)).title()
    return c

x = fun( "ab", "caa" )

What is the value of x?
A 'ab ab ab '
B 'Ab Ab Ab '
C 'AB AB AB '
Boolean Logic
- bool is a type with two possible values:
  - True
  - False
- We use these to make decisions.
- Their logic is based on Boolean algebra.
- Operators:
  - and
  - or
  - not
Example: Boolean logic

\[ 0 < x \leq 10 \]

\[ (x > 0) \text{ and } (x \leq 10) \]
### Boolean operators

<table>
<thead>
<tr>
<th>and</th>
<th>True</th>
<th>False</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
</tbody>
</table>

- True when BOTH inputs are true

<table>
<thead>
<tr>
<th>or</th>
<th>True</th>
<th>False</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>False</td>
</tr>
</tbody>
</table>

- True when EITHER input is true
## Boolean operators

<table>
<thead>
<tr>
<th>not</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
</tr>
</tbody>
</table>

Inverts truth-value
def fun():
    return True and False

x = fun() and not (True or False)

What is the value of x?
A True
B False
These produce Boolean output.
- less than, <
- greater than, >
- less than or equal to, <=
- greater than or equal to, >=
- equal to, ==
- not equal to, !=
Example

\[
\begin{align*}
  a &= 5 \\
  b &= 3 \\
  x &= (a < 5) \text{ and } ((b \leq 5) \text{ or } (a \neq b))
\end{align*}
\]

What is the value of \(x\)?

A True
B False
Example

```python
a = 'URSA MAJOR'
b = 'GEMINI'

x = a < b and a[1] != b[-2]
```

What is the value of `x`?
A  True
B  False
def fun(a,b):
    return a<b
a = 3
b = 4
x = fun(b,a)

What is the value of $x$?
A True
B False
Conditional Execution
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.
Example: `if` statement

```python
ans = input( "Enter a number:" )
if float(ans) < 0:
    print( "The number was negative." )
```
We create an if statement as follows:
- the keyword if
- a logical comparison (results in bool)
- a block of code
This lets us make decisions in the program!
We can change program behavior as it executes.
Example: if statements

```python
ans = input( "Enter a number:" )
if float(ans) < 0:
    print( "The number was negative." )
if float(ans) > 0:
    print( "The number was positive." )
if float(ans) == 0:
    print( "The number was zero." )
```
Reminders
Reminders

- Homework #2 is due Wed Oct. 19.