Conditionals in Python

L435/L555

Dept. of Linguistics, Indiana University

Fall 2016
Interactive programs

- we know how to output something on the screen:
  `print('Hello world.')`

- input:
  `input(<prompt>)`
  - returns the input from the keyboard

Example

```python
name = input('type your name: ')
```
**If statement**

Write a program to: 1) ask a user to type his/her name, 2) check if the user is known, & 3) print a welcome statement

- We know how to do the first part:
  ```python
  known_users = [ 'Sandra', 'Markus' ]
  name = input( 'type your name:' )
  ```

- We can check whether a person is in the list of known users:
  ```python
  name in known_users
  ```

- But how do we tell python to print a welcome message if the name is known?
If statement

- syntax:

  ```python
  if <test>:
    do this
  ```

- full program:

  ```python
  known_users = ['Sandra', 'Markus']
  name = input('type your name: ')

  if name in known_users:
    print("Hello" + name)
  ```

Definition

In python, blocks are created by the use of a colon, followed by an indented section of text.

```python
if <test>:
    do something
    do another thing
    a final thing
do this regardless
```
Truth values

- a test (in the if statement) corresponds to a yes/no question and can be either true or false
- the following values count as false:
  
  False
  None
  0
  [] (empty list)
  {} (empty dict)
  '' (empty string)
  () (empty tuple)
- everything else counts as true!
Else statements

▶ In case the program needs to do something when the test is false, use the else statement
▶ E.g. if a user is not known, add him/her to the list

Example

```python
known_users = [ 'Sandra', 'Markus' ]
name = input( 'type your name:' )

if name in known_users:
    print( 'Hello ' + name + ' .' )
    print( 'It is nice to have you back.' )
else:
    known_users.append(name)
    print( 'You have been added to the list.' )
```

7 / 13
Elif

- if you want to check the next condition in the else case, there is a shortcut for else if called **elif**

**Example**

```python
known_users = [ 'Sandra', 'Markus' ]
name = input( 'type your name:' )

if name in known_users:
    print( 'Hello ' + name + ' . ' )
    print( 'It is nice to have you back. ' )
elif len(name) > 20:
    print( 'Your name is too long! ' )
else:
    known_users . append(name)
    print( 'You have been added to the list. ' )
```
Nested blocks

Example

define_users = ['Sandra', 'Markus']
name = input('Type your name: ')

if name in known_users:
    print('Hello ' + name + '!
    if name.startswith('Dr. '):
        print('Taking yourself seriously, huh?')
    else:
        print('You’re my buddy.
else:
    known_users.append(name)
    print('You have been added to the list.')
More tests

- `x == y`  x equals y
- `x < y`   x is less than y
- `x > y`   x is greater than y
- `x >= y`  x is greater than or equal to y
- `x <= y`  x is less than or equal to y
- `x != y`  x is not equal to y
- `x is y`  x is the same object as y
- `x is not y` x is not the same object as y
- `x in y`  x is a member of y
- `x not in y` x is not a member of y

▶ Caution: `=` and `==` are different:
  - `=` assigns a value
  - `==` compares values
Equality vs. identity

Having the same values is not the same thing as being the same object

```python
>>> x = y = [1, 2, 3]
>>> z = [1, 2, 3]
>>> x == y
True
>>> x == z
True
>>> x is y
True
>>> x is z
False
```
Booleans

Definition

You can combine conditions with and and or, and negate with not

Example

```python
if 5 < x < 10 and x not in y:
    print('x is between 5 and 10')
    print('and is not in the list y')
```
Short-circuit logic

Python evaluates the first part of an `and/or` condition and can short circuit

- If `x in x or y` is True, no need to evaluate both
- If `x in x and y` is False, no need to evaluate both

This means you can do things like:

```python
if line and line.startswith('%'):
```

You can also do things like:

```python
name = input('Please enter your name: ') or '<unknown>'
```