Conditionals in Python

L555

Dept. of Linguistics, Indiana University
Fall 2010
Interactive Programs

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

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

Example

```python
name = raw_input('type your name: ')
```
If Statement

Let’s assume you want to write a program that: 1) asks a user to type his/her name, 2) checks if it is a known user, and 3) prints a welcome statement.

- we know how to do the first part:

```python
known_users = [ 'Sandra', 'Markus' ]
name = raw_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 = raw_input('type your name: ')
  
  if name in known_users:
      print 'Hello ' + name
  ```
Blocks and Indenting

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 = raw_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 .'
```
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 = raw_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

```python
known_users = ['Sandra', 'Markus']
name = raw_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.'
```

▶ Note: If you want to use an apostrophe in a string, you have to escape it with \\!"
Nested Blocks

Example

```python
def check_user(name):
    known_users = [ 'Sandra', 'Markus' ]
    name = raw_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.'

check_user(input())
```

- Note: If you want to use an apostrophe in a string, you have to escape it with ’\’!
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
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
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'
```