Python for Dummies

Quiz navigation

1 2 3 4 5 6

Finish attempt ...

Start a new preview

Navigation

Home
- My home
- Site pages
- My profile
- Courses
  - py4dum
    - Participants
    - Reports
    - General
      - News forum
      - Strings practice quiz
      - **Examples of different forms of question**
        - Info
        - Results

Settings

Quiz administration
- Edit settings
- Group overrides
- User overrides
- Edit quiz
- **Preview**
- Locally assigned roles
- Permissions
- Check permissions
- Logs
- Backup
- Restore
- Question bank

Course administration

Switch role to...

My profile settings

Site administration

Question 1

Correct

Mark 1.00 out of 1.00

Edit question

Write a function *double(n)* that returns twice n.

For example:

```
double(1) → 2
double(-7) → -14
```

Answer:

```
def double(n):
    return 2 * n
```

Check

Test	Expected	Got	
`double(1)`	2	2	✓
`double(-7)`	-14	-14	✓
`double(11)`	22	22	✓

Passed all tests! ✓

Correct
Marks for this submission: 1.00/1.00.

Question 2

Correct

Mark 1.00 out of 1.00

Edit question

Write a program (one line!) that prints "Hello world!"

For example:

```
→ Hello world!
```

Answer:

Search

```
print "Hello world!"
```

Check

Expected	Got	
Hello world!	Hello world!	✓

Passed all tests! ✓

Correct
Marks for this submission: 1.00/1.00.

Question 3

Correct

Mark 1.00 out of 1.00

Edit question

Write a function *sayHello(firstName, lastName)* that prints a welcoming message of the form "Hello <firstName> <lastName>".

For example:

```
sayHello('Angus', 'McGurkinshaw') → Hello Angus McGurkinshaw
```

Answer:

```
def sayHello(first, last):
   print "Hello " + first + ' ' + last
```

Check

Test	Expected	Got	
`sayHello('Angus', 'McGurkinshaw')`	`Hello Angus McGurkinshaw`	`Hello Angus McGurkinshaw`	✓
`sayHello("Fred", "Bloggs")`	`Hello Fred Bloggs`	`Hello Fred Bloggs`	✓

Passed all tests! ✓

Correct
Marks for this submission: 1.00/1.00.

Question 4

Correct

Mark 1.00 out of 1.00

Edit question

Write a class *Name* as follows. The constructor takes two strings, first name and last name, as parameters, e.g. *person = Name('Angus', 'McGurkinshaw')*. It has a method *fullName()*, which returns the name in the form *Andrew McGurkinshaw*, and two "setter" methods *setFirstName(fname)* and *setLastName(lname)* that respectively set the first and last names.

For example:

Shell Input	Output

`obj = Name('Angus', 'McGurkinshaw')`
`print "Full name is", obj.fullName()`
`obj.setFirstName('Angela')`
`print "Now full name is", obj.fullName()`
`obj.setLastName('Ash')`
`print "Name is now", obj.fullName()`
	`Full name is Angus McGurkinshaw`
	`Now full name is Angela McGurkinshaw`
	`Name is now Angela Ash`

Answer:

```
class Name (object):
   def __init__(self, first, last):
      self.first = first
      self.last = last

   def setFirstName(self, first):
      self.first = first

   def setLastName(self, last):
      self.last = last

   def fullName(self):
      return self.first + ' ' + self.last
```

Check

Test	Expected	Got	

`obj = Name('Angus', 'McGurkinshaw')`
`print "Full name is", obj.fullName()`
`obj.setFirstName('Angela')`
`print "Now full name is", obj.fullName()`
	`Full name is Angus McGurkinshaw`
	`Now full name is Angela McGurkinshaw`
	`Name is now Angela Ash`
		`Full name is Angus McGurkinshaw`
		`Now full name is Angela McGurkinshaw`
		`Name is now Angela Ash`
			✓

`obj.setLastName('Ash')` `print "Name is now", obj.fullName()`			

`bod = Name('Fred', 'Bloggs')`
`print "Bod:", bod.fullName()`
`bod.setFirstName('Black')`
`bod.setLastName('Pete')`
`print "Bod is now", bod.fullName()`
	`Bod: Fred Bloggs`
	`Bod is now Black Pete`
		`Bod: Fred Bloggs`
		`Bod is now Black Pete`
			✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 5

Correct

Mark 1.00 out of 1.00

Edit question

Write a program that reads from stdin until the end of file is reaches, and prints the number of lines read to standard output

For example:

Standard Input	Output

`This is line 1`
`This is line 2`
`This is line 3 and the last line`
	`Number of lines: 3`

Answer:

```
count = 0
try:
   while True:
      line = raw_input()
      count += 1
except EOFError:
   pass

print "Number of lines:", count
```

Check

Stdin	Expected	Got	

`This is line 1`
`This is line 2`
`This is line 3 and the last line`
	`Number of lines: 3`	`Number of lines: 3`	✓

`This is line 1`	`Number of lines: 1`	`Number of lines: 1`	✓

`A`
`B`
`C`
	`Number of lines: 6`	`Number of lines: 6`	✓

D E F			

Passed all tests! ✓

Correct
Marks for this submission: 1.00/1.00.

Question 6

Not complete

Mark 0.33 out of 1.00

Edit question

Write a function *readnum(prompt)* that repeatedly issues the given prompt and reads a line of input until it receives a line that represents a valid integer. It then returns the integer value of that line.

For example:

Standard Input	Shell Input	Total output
Nope Not this one either 10 20	n = readnum("Gimme: ") print "Number is", n	Gimme: Gimme: Gimme: Number is 10

Answer:

```
def readnum(prompt):
   ok = False
   while not ok:
      try:
         num = int(raw_input(prompt))
         ok = True
      except ValueError:
         pass
   return num
```

Check

Test	Stdin	Expected	Got	
n = readnum("Gimme: ") print "Number is", n	Nope Not this one either 10 20	Gimme: Gimme: Gimme: Number is 10	Gimme: Gimme: Gimme: Number is 10	✓
ans = readnum("Prompt:") print "Ans:", ans	-1100	Prompt:Ans: -1100	Prompt:Ans: -1100	✓
n = readnum("X") print "n =", n	1 raw cat 2 dead dogs 3 blind mice 1000001	XXXXn = 1000001	XXXXn = 1000001	✓

Passed all tests!

Correct
Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.33/1.00**. This submission attracted a penalty of 0.33. Total penalties so far: 1.00.

Next