

|  |
| --- |
|  |

|  |  |
| --- | --- |
|  | ***Review*** |

Answer the following questions, if you’re not sure or even don’t remember, revisit our videos, refer to our book or ask your instructors or your TAs:

Remember that there isn’t only one right answer, only good ones :)

* Why should we use functions at all?

Functions can help us work faster and simplify our code. Whenever we have a repetitive task, we can considerably speed up our workflow by using a function each time we do that task.

One of the key aspects that make functions great is **reusability**. We generally use functions to speed up repetitive tasks, so a function is no good if it's not reusable for the various instances of a certain kind of task.

* How to define/declare a function?

Def <Name> (Parameters):

Statements

* How to call/use a function?

Name\_of\_Function()

* What is return, why and how do we use it?

The [return](https://docs.python.org/3/reference/simple_stmts.html#return) statement returns with a value from a function. return without an expression argument returns None. Falling off the end of a function also returns None.

* Do we have to use return in **every** function?

No

* What are function arguments/parameters, why and how we use it?

Input variables are called parameters, and the various values that parameters take are called arguments

* How to use function from a different file other than our currently working file?

Import name\_of\_function

Note: Function **arguments** are sometimes **also** called function **parameters**

Reference:

* [How to think like a computer scientist](https://drive.google.com/open?id=1S1f_krK63QLamUvte2CmYeMoPylgwW7B), chapter 4
* [Python cheat sheet](https://www.dropbox.com/s/qgws0sammkb6fi6/1.%20beginners_python_cheat_sheet_pcc_all.pdf?dl=0)

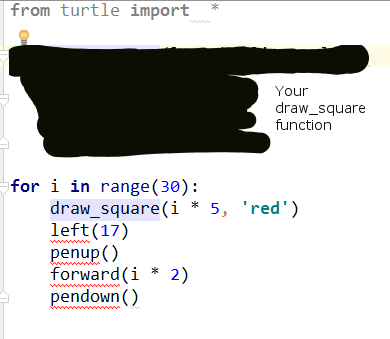
|  |  |
| --- | --- |
| *http://www.bestappsforkids.com/wp-content/uploads/2012/04/save-turtle.png* | ***Turtle exercise*** |

1. Write a function that prints out “Hello world” 3 times (note: no arguments, no return)
2. Write a function that takes **2 numbers as arguments** and print out sum of them (note: has arguments, no return)
3. Write a Python function that **draws a square**, named draw\_square, takes **2 arguments**: length and color, where length is the length of its side and color is the color of its bound (line color)
4. Now, another programmer named ‘T.Anh’ will use your code in exercise 1. He writes as follows:

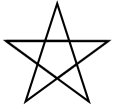
**for** i **in** range(30):  
 **draw\_square**(i \* 5, **'red'**)  
 left(17)  
 penup()  
 forward(i \* 2)  
 pendown()

Copy this code into your editor, run the whole program and see what it draws:

*Note: If your code does not run, try not to modify* T.Anh*’s code, modify your function instead*



1. Write a Python function that draws a star, named draw\_star, take 3 parameters: x, y, and length. Where x, y are the location of the star, length is the length of its side

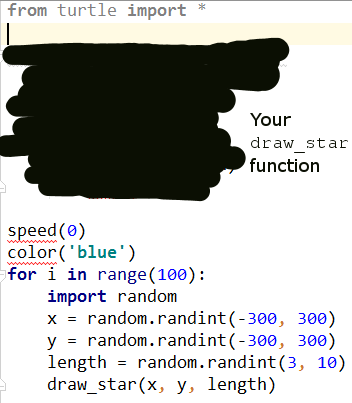


*Hint: Turn 144 degree at each point, Google ‘python 3 turtle go to position’*

1. Again, your function will be used by other programmers, they write as follows:

speed(0)  
color(**'blue'**)  
**for** i **in** range(100):  
 **import** random  
 x = random.randint(-300, 300)  
 y = random.randint(-300, 300)  
 length = random.randint(3, 10)  
 draw\_star(x, y, length)

Copy this code into your editor, run the whole program and see what it draws:



|  |  |
| --- | --- |
| http://images.8tracks.com/cover/i/001/358/131/7357.original-3735.jpg?rect=0,29,289,289&q=98&fm=jpg&fit=max&w=100&h=100 | ***Serious exercise*** |

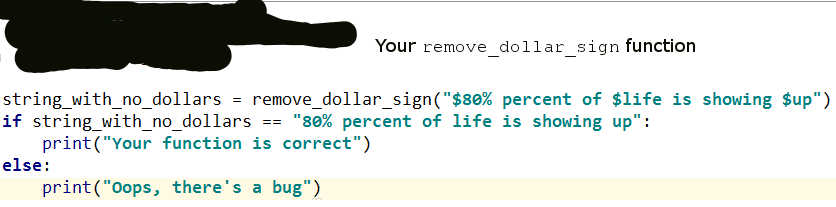
1. Write a function that removes the dollar sign (“$”) in a string, named remove\_dollar\_sign, takes 1 arguments: s, where s is the input string, **returns** the new string with no dollar sign in it

*Hint: Google “Python string replace remove”*

1. Now, another programmer named Hiep will use your code in exercise 3. He writes as follows:

string\_with\_no\_dollars = remove\_dollar\_sign(**"$80% percent of $life is to show $up"**)  
**if** string\_with\_no\_dollars == **"80% percent of life is to show up"**:  
 print(**"Your function is correct"**)  
**else**:  
 print(**"Oops, there's a bug"**)

Copy this code into your editor, run the whole program and see what it prints out:



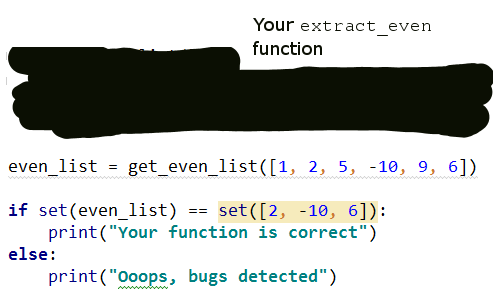
If it prints out **“Your function is correct”**, we’re good

If it prints out **“Oops, there’s a bug”**, you might want to come back and check your function

1. Write a function that extracts the even items in a given integer list, named get\_even\_list, takes 1 parameter: l, where l is the given integer list ([1, 4, 5, -1, 10] for example), returns a new list contains only even numbers ([4, 10] if the given list is [1,4,5,-1,10])
2. Let’s take your function to the test. The tester will write his/her test code as follows:

even\_list = get\_even\_list([1, 2, 5, 9, -10, 6])  
  
**if** set(even\_list) == set([2, -10, 6]):  
 print(**"Your function is correct"**)  
**else**:  
 print(**"Ooops, bugs detected"**)

Copy this code into your editor, run the whole program and see what it prints out:



If it prints out **“Your function is correct”**, we’re good

If it prints out **“Oops, bugs detected”**, you might want to come back and check your function

*Note:* ***set*** *is an unordered data structure, meaning set of (1, 2,3) equals set of (3, 1, 2)*

1. Write a function named is\_inside that checks if a point is inside a rectangle, takes 2 parameters, the first is a list with 2 elements respectively represents x and y coordinates of the given point, the second is a list with 4 elements respectively represents x, y coordinates and width height of the given rectangle

For example:

is\_inside([100, 120], [140, 60, ])

should return False

**140**

**60**

**width = 100**

**height = 200**

X

Y

**120**

**100**

and

is\_inside([200, 120], [140, 60, 100, 200])

should return True

**140**

**60**

**width = 100**

**height = 200**

X

Y

**120**

**200**