Practice Exercises for Functions

Practice Exercises for Functions

Solve each of the practice exercises below. Each problem includes two CodeSkulptor3 links: one for a template that you should use as a starting point for your solution and one to our solution to the exercise.

1. Write a Python function \color{red}{\verb|miles\_to\_feet|}miles\_to\_feet that takes a parameter \color{red}{\verb|miles|}miles and returns the number of feet in \color{red}{\verb|miles|}miles miles. [Miles to feet template](http://py3.codeskulptor.org/#exercises3_fn_miles_to_feet_template.py) --- [Miles to feet solution](http://py3.codeskulptor.org/#exercises3_fn_miles_to_feet_solution.py)
2. Write a Python function \color{red}{\verb|total\_seconds|}total\_seconds that takes three parameters \color{red}{\verb|hours|}hours, \color{red}{\verb|minutes|}minutes and \color{red}{\verb|seconds|}seconds and returns the total number of seconds for \color{red}{\verb|hours|}hours hours, \color{red}{\verb|minutes|}minutes minutes and \color{red}{\verb|seconds|}seconds seconds. [Hours to seconds template](http://py3.codeskulptor.org/#exercises3_fn_hours_to_seconds_template.py) --- [Hours to seconds solution](http://py3.codeskulptor.org/#exercises3_fn_hours_to_seconds_solution.py)
3. Write a Python function \color{red}{\verb|rectangle\_perimeter|}rectangle\_perimeter that takes two parameters \color{red}{\verb|width|}width and \color{red}{\verb|height|}height corresponding to the lengths of the sides of a rectangle and returns the perimeter of the rectangle in inches. [Perimeter of rectangle template](http://py3.codeskulptor.org/#exercises3_fn_perimeter_of_rectangle_template.py) --- [Perimeter of rectangle solution](http://py3.codeskulptor.org/#exercises3_fn_perimeter_of_rectangle_solution.py)
4. Write a Python function \color{red}{\verb|rectangle\_area|}rectangle\_area that takes two parameters \color{red}{\verb|width|}width and \color{red}{\verb|height|}height corresponding to the lengths of the sides of a rectangle and returns the area of the rectangle in square inches. [Area of rectangle template](http://py3.codeskulptor.org/#exercises3_fn_area_of_rectangle_template.py) --- [Area of rectangle solution](http://py3.codeskulptor.org/#exercises3_fn_area_of_rectangle_solution.py)
5. Write a Python function \color{red}{\verb|circle\_circumference|}circle\_circumference that takes a single parameter \color{red}{\verb|radius|}radius corresponding to the radius of a circle in inches and returns the the circumference of a circle with radius \color{red}{\verb|radius|}radius in inches. Do not use \pi = 3.14*π*=3.14, instead use the [\color{red}{\verb|math|}math module](https://docs.python.org/3/library/math.html) to supply a higher-precision approximation to \pi*π*. [Circumference of circle template](http://py3.codeskulptor.org/#exercises3_fn_circumference_of_circle_template.py) --- [Circumference of circle solution](http://py3.codeskulptor.org/#exercises3_fn_circumference_of_circle_solution.py)
6. Write a Python function \color{red}{\verb|circle\_area|}circle\_area that takes a single parameter \color{red}{\verb|radius|}radius corresponding to the radius of a circle in inches and returns the the area of a circle with radius \color{red}{\verb|radius|}radius in square inches. Do not use \pi = 3.14*π*=3.14, instead use the [\color{red}{\verb|math|}math module](https://docs.python.org/3/library/math.html) to supply a higher-precision approximation to \pi*π*. [Area of circle template](http://py3.codeskulptor.org/#exercises3_fn_area_of_circle_template.py) --- [Area of circle solution](http://py3.codeskulptor.org/#exercises3_fn_area_of_circle_solution.py)
7. Write a Python function \color{red}{\verb|future\_value|}future\_value that takes three parameters \color{red}{\verb|present\_value|}present\_value, \color{red}{\verb|annual\_rate|}annual\_rate and \color{red}{\verb|years|}years and returns the future value of \color{red}{\verb|present\_value|}present\_value dollars invested at \color{red}{\verb|annual\_rate|}annual\_rate percent interest, compounded annually for \color{red}{\verb|years|}years years. [Future value template](http://py3.codeskulptor.org/#exercises3_fn_future_value_template.py) --- [Future value solution](http://py3.codeskulptor.org/#exercises3_fn_future_value_solution.py)
8. Write a Python function \color{red}{\verb|name\_tag|}name\_tag that takes as input the parameters \color{red}{\verb|first\_name|}first\_name and \color{red}{\verb|last\_name|}last\_name (strings) and returns a string of the form \color{red}{\verb|"My name is % %."|}"My name is % %." where the percents are the strings \color{red}{\verb|first\_name|}first\_name and \color{red}{\verb|last\_name|}last\_name. Reference the test cases in the provided template for an exact description of the format of the returned string. [Name tag template](http://py3.codeskulptor.org/#exercises3_fn_nametag_template.py) --- [Name tag solution](http://py3.codeskulptor.org/#exercises3_fn_nametag_solution.py)
9. Write a Python function \color{red}{\verb|name\_and\_age|}name\_and\_age that takes as input the parameters \color{red}{\verb|name|}name (a string) and \color{red}{\verb|age|}age (a number) and returns a string of the form \color{red}{\verb|"% is % years old."|}"% is % years old." where the percents are the string forms of \color{red}{\verb|name|}name and \color{red}{\verb|age|}age. Reference the test cases in the provided template for an exact description of the format of the returned string. [Name and age template](http://py3.codeskulptor.org/#exercises3_fn_nameage_template.py) --- [Name and age solution](http://py3.codeskulptor.org/#exercises3_fn_nameage_solution.py)
10. Write a Python function \color{red}{\verb|point\_distance|}point\_distance that takes as input the parameters \color{red}{\verb|x\_0|}x\_0, \color{red}{\verb|y\_0|}y\_0, \color{red}{\verb|x\_1|}x\_1 and \color{red}{\verb|y\_1|}y\_1, and returns the distance between the points (x\_0,y\_0)(*x*0​,*y*0​) and (x\_1, y\_1)(*x*1​,*y*1​). [Point distance template](http://py3.codeskulptor.org/#exercises3_fn_point_distance_template.py) --- [Point distance solution](http://py3.codeskulptor.org/#exercises3_fn_point_distance_solution.py)
11. **Challenge:** Write a Python function \color{red}{\verb|triangle\_area|}triangle\_area that takes the parameters \color{red}{\verb|x\_0|}x\_0, \color{red}{\verb|y\_0|}y\_0, \color{red}{\verb|x\_1|}x\_1,\color{red}{\verb|y\_1|}y\_1, \color{red}{\verb|x\_2|}x\_2, and \color{red}{\verb|y\_2|}y\_2 (all numbers), and returns the area of the triangle with vertices (x\_0,y\_0)(*x*0​,*y*0​), (x\_1, y\_1)(*x*1​,*y*1​) and (x\_2, y\_2)(*x*2​,*y*2​). (Hint: use the function \color{red}{\verb|point\_distance|}point\_distance as a helper function and apply [Heron's formula](http://en.wikipedia.org/wiki/Heron's_formula).) [Triangle area template](http://py3.codeskulptor.org/#exercises3_fn_area_of_triangle_template.py) --- [Triangle area solution](http://py3.codeskulptor.org/#exercises3_fn_area_of_triangle_solution.py)
12. **Challenge:** Write a Python function \color{red}{\verb|print\_digits|}print\_digits that takes an integer \color{red}{\verb|number|}number in the range [0,100)[0,100), (i.e., at least 0, but less than 100) and prints the message \color{red}{\verb|"The tens digit is %, and the ones digit is %."|}"The tens digit is %, and the ones digit is %.", where the percent signs should be replaced with the appropriate values. (Hint: Use the arithmetic operators for integer division \color{red}{\verb|//|}// and remainder \color{red}{\verb|%|}% to find the two digits. Note that this function should print the desired message, rather than returning it as a string. [Print digits template](http://py3.codeskulptor.org/#exercises3_fn_print_digits_template.py) --- [Print digits solution](http://py3.codeskulptor.org/#exercises3_fn_print_digits_solution.py)
13. **Challenge:**[Powerball](http://en.wikipedia.org/wiki/Powerball) is lottery game in which 6 numbers are drawn at random. Players can purchase a lottery ticket with a specific number combination and, if the number on the ticket matches the numbers generated in a random drawing, the player wins a massive jackpot. Write a Python function \color{red}{\verb|powerball|}powerball that takes no arguments and prints the message \color{red}{\verb|"Today's numbers are %, %, %, %, and %. The Powerball number is %."|}"Today’s numbers are %, %, %, %, and %. The Powerball number is %.". The first five numbers should be random integers in the range [1,60)[1,60), i.e., at least 1, but less than 60. In reality, these five numbers must all be distinct, but for this problem, we will allow duplicates. The Powerball number is a random integer in the range [1,36)[1,36), i.e., at least 1 but less than 36. Note that this function should print the desired message, rather than returning it as a string. [Powerball template](http://py3.codeskulptor.org/#exercises3_fn_powerball_template.py) --- [Powerball solution](http://py3.codeskulptor.org/#exercises3_fn_powerball_solution.py)

For the last problem, you will need to use the [CodeSkulptor3 Docs](http://py3.codeskulptor.org/docs.html) to learn how to import the \color{red}{\verb|random|}random module and then use the function \color{red}{\verb|random.randrange()|}random.randrange() to generate the appropriate random numbers. We will cover how import and use modules in more detail during week 4.