

Solve each of the practice exercises below. Each problem includes three CodeSkulptor links: one for a template that you should use as a starting point for your solution, one to our solution to the exercise, and one to a tool that automatically checks your solution.

1. Write a Python function `miles_to_feet` that takes a parameter `miles` and returns the number of feet in `miles` miles. [Miles to feet template](#) --- [Miles to feet solution](#) --- [Miles to feet \(Checker\)](#)
2. Write a Python function `total_seconds` that takes three parameters `hours`, `minutes` and `seconds` and returns the total number of seconds for `hours` hours, `minutes` minutes and `seconds` seconds. [Hours to seconds template](#) --- [Hours to seconds solution](#) --- [Hours to seconds \(Checker\)](#)
3. Write a Python function `rectangle_perimeter` that takes two parameters `width` and `height` corresponding to the lengths of the sides of a rectangle and returns the perimeter of the rectangle in inches. [Perimeter of rectangle template](#) --- [Perimeter of rectangle solution](#) --- [Perimeter of rectangle \(Checker\)](#)
4. Write a Python function `rectangle_area` that takes two parameters `width` and `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](#) --- [Area of rectangle solution](#) --- [Area of rectangle \(Checker\)](#)
5. Write a Python function `circle_circumference` that takes a single parameter `radius` corresponding to the radius of a circle in inches and returns the the circumference of a circle with radius `radius` in inches. Do not use  $\pi = 3.14$ , instead use the `math` module to supply a higher-precision approximation to  $\pi$ . [Circumference of circle template](#) --- [Circumference of circle solution](#) --- [Circumference of circle \(Checker\)](#)
6. Write a Python function `circle_area` that takes a single parameter `radius` corresponding to the radius of a circle in inches and returns the the area of a circle with radius `radius` in square inches. Do not use  $\pi = 3.14$ , instead use the `math` module to supply a higher-precision approximation to  $\pi$ . [Area of circle template](#) --- [Area of circle solution](#) --- [Area of circle \(Checker\)](#)
7. Write a Python function `future_value` that takes three parameters `present_value`, `annual_rate` and `years` and returns the future value of `present_value` dollars invested at `annual_rate` percent interest, compounded annually for `years` years. [Future value template](#) --- [Future value solution](#) --- [Future value \(Checker\)](#)
8. Write a Python function `name_tag` that takes as input the parameters `first_name` and `last_name` (strings) and returns a string of the form "My name is % %." where the percents are the strings `first_name` and `last_name`. Reference the test cases in the provided template for an exact description of the format of the returned string. [Name tag template](#) --- [Name tag solution](#) --- [Name tag \(Checker\)](#)
9. Write a Python function `name_and_age` that takes as input the parameters `name` (a string) and `age` (a number) and returns a string of the form "% is % years old." where the percents are the string forms of `name` and `age`. Reference the test cases in the provided template for an exact description of the format of the returned string. [Name and age template](#) --- [Name and age solution](#) --- [Name and age \(Checker\)](#)
10. Write a Python function `point_distance` that takes as the parameters `x0`, `y0`, `x1` and `y1`, and

returns the distance between the points  $(x_0, y_0)$  and  $(x_1, y_1)$  . [Point distance template](#) --- [Point distance solution](#) --- [Point distance \(Checker\)](#)

11. **Challenge:** Write a Python function `triangle_area` that takes the parameters `x0`, `y0`, `x1`, `y1`, `x2`, and `y2`, and returns the area of the triangle with vertices  $(x_0, y_0)$ ,  $(x_1, y_1)$  and  $(x_2, y_2)$  . (Hint: use the function `point_distance` as a helper function and apply [Heron's formula](#).) [Triangle area template](#) --- [Triangle area solution](#) --- [Triangle area \(Checker\)](#)
12. **Challenge:** Write a Python function `print_digits` that takes an integer `number` in the range  $[0, 100)$ , i.e., at least 0, but less than 100. It prints the message `"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 `//` and remainder `%` to find the two digits. Note that this function should print the desired message, rather than returning it as a string. [Print digits template](#) --- [Print digits solution](#) --- [Print digits \(Checker\)](#)
13. **Challenge:** [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 `powerball` that takes no arguments and prints the message `"Today's numbers are %, %, %, %, and %. The Powerball number is %."`. The first five numbers should be random integers in the range  $[1, 60)$ , 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  $[1, 36)$ , i.e., at least 1 but less than 36. Use the `random` module and the function `random.randrange` to generate the appropriate random numbers. Note that this function should print the desired message, rather than returning it as a string. [Powerball template](#) --- [Powerball solution](#) --- [Powerball \(Checker\)](#)