1st Exercise

Complete the code to make the program print "Mary is 20 years old" to standard output:



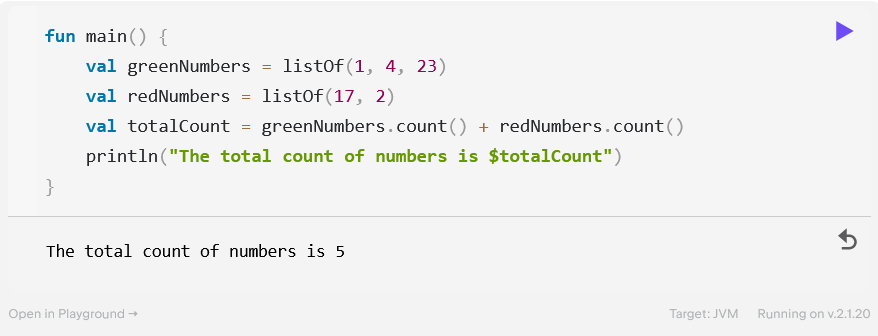
2nd Exercise

Explicitly declare the correct type for each variable:



3rd Exercise

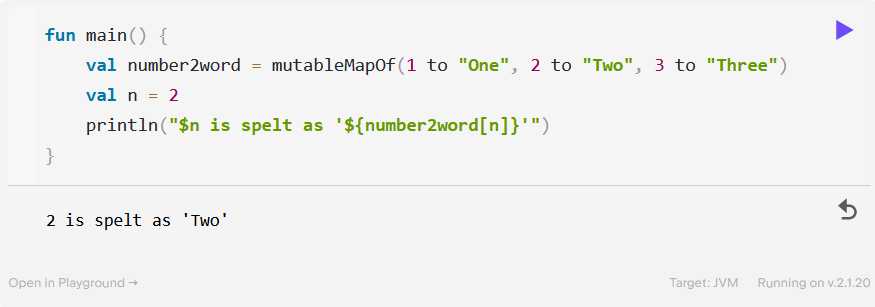
You have a list of “green” numbers and a list of “red” numbers. Complete the code to print how many numbers there are in total.



You have a set of protocols supported by your server. A user requests to use a particular protocol. Complete the program to check whether the requested protocol is supported or not (isSupported must be a Boolean value).



Define a map that relates integer numbers from 1 to 3 to their corresponding spelling. Use this map to spell the given number.



4th Exercise

Create a simple game where you win if throwing two dice results in the same number. Use if to print You win :) if the dice match or You lose :( otherwise.

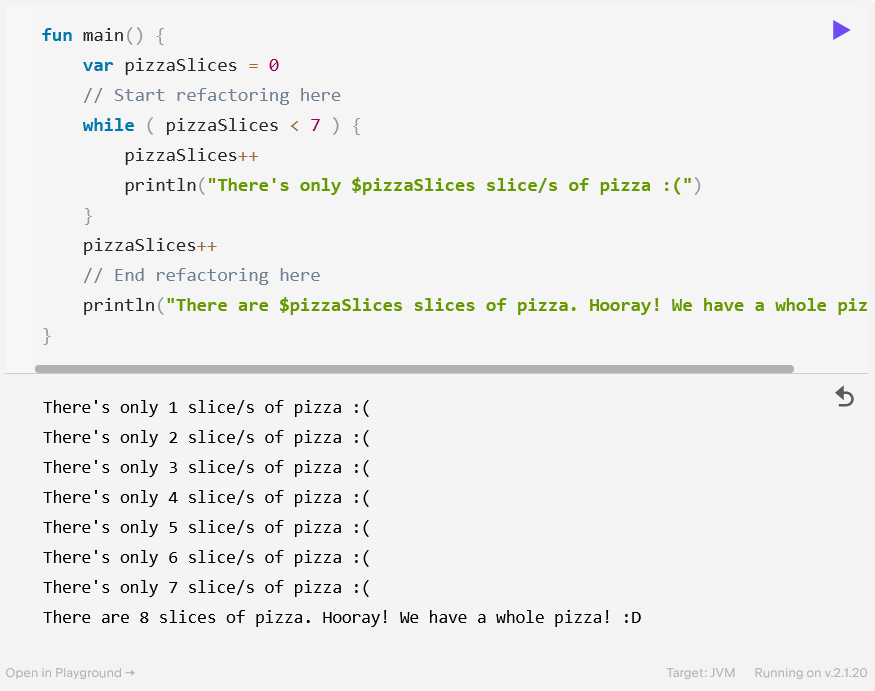


Using a when expression, update the following program so that it prints the corresponding actions when you input the names of game console buttons.



You have a program that counts pizza slices until there’s a whole pizza with 8 slices. Refactor this program in two ways:

* Use a while loop.



* Use a do-while loop.



Write a program that simulates the [Fizz buzz](https://en.wikipedia.org/wiki/Fizz_buzz) game. Your task is to print numbers from 1 to 100 incrementally, replacing any number divisible by three with the word "fizz", and any number divisible by five with the word "buzz". Any number divisible by both 3 and 5 must be replaced with the word "fizzbuzz".

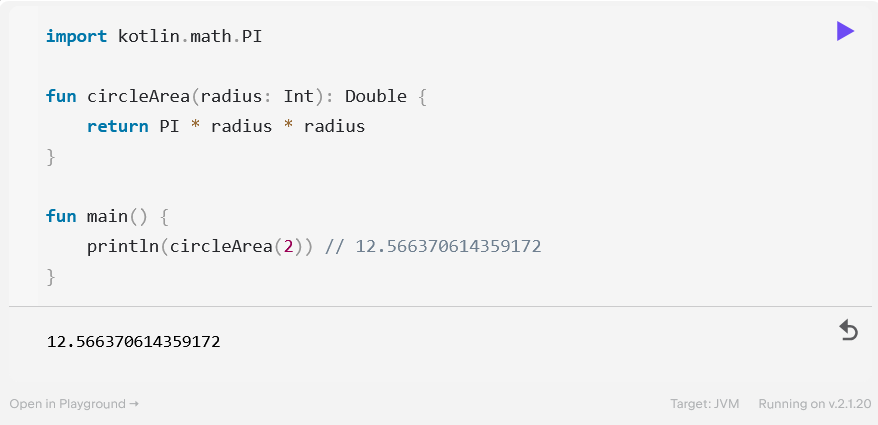


You have a list of words. Use for and if to print only the words that start with the letter l.

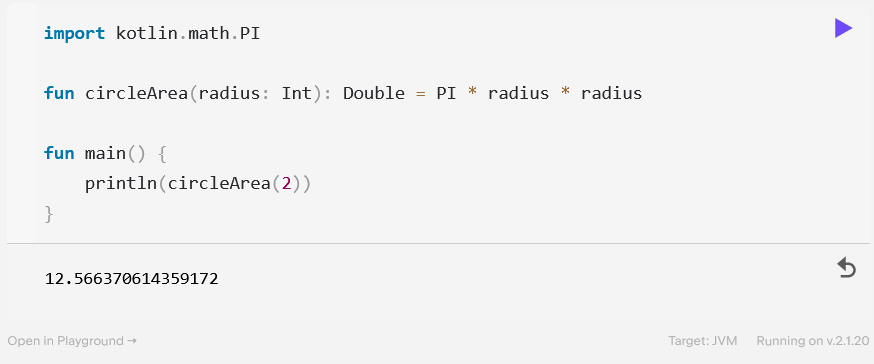


5th Exercise

Write a function called circleArea that takes the radius of a circle in integer format as a parameter and outputs the area of that circle.



Rewrite the circleArea function from the previous exercise as a single-expression function.



You have a function that translates a time interval given in hours, minutes, and seconds into seconds. In most cases, you need to pass only one or two function parameters while the rest are equal to 0. Improve the function and the code that calls it by using default parameter values and named arguments so that the code is easier to read.

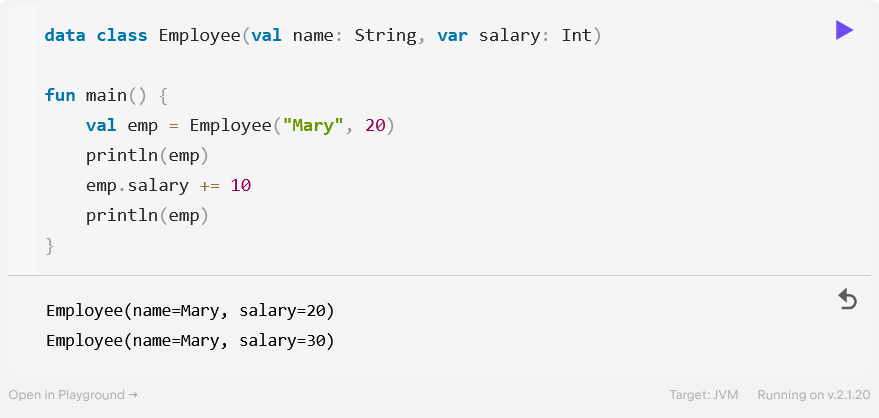


You have a list of actions supported by a web service, a common prefix for all requests, and an ID of a particular resource. To request an action title over the resource with ID: 5, you need to create the following URL: https://example.com/book-info/5/title. Use a lambda expression to create a list of URLs from the list of actions.



6th Exercise

Define a data class Employee with two properties: one for a name, and another for a salary. Make sure that the property for salary is mutable, otherwise you won’t get a salary boost at the end of the year! The main function demonstrates how you can use this data class.



Declare the additional data classes that are needed for this code to compile.



To test your code, you need a generator that can create random employees. Define a RandomEmployeeGenerator class with a fixed list of potential names (inside the class body). Configure the class with a minimum and maximum salary (inside the class header). In the class body, define the generateEmployee() function. Once again, the main function demonstrates how you can use this class.



7th Exercise

You have the employeeById function that gives you access to a database of employees of a company. Unfortunately, this function returns a value of the Employee? type, so the result can be null. Your goal is to write a function that returns the salary of an employee when their id is provided, or 0 if the employee is missing from the database.



Write a function that takes an Int value and an action (a function with type () -> Unit) which then repeats the action the given number of times. Then use this function to print “Hello” 5 times.

