

Learn to Code

CodeWars

Student Workbook 6

Version 5.1 Y

Table of Contents

Module 1 Algorithms and CodeWars.....	1-1
Section 1–1 Algorithms	1-2
CodeWars Katas: Practice Practice Practice.....	1-4
Signing up for CodeWars.....	1-5
CodeWars Coding Solutions	1-7
Getting Started.....	1-8
Solving a Kata	1-10
Debugging Your Kata.....	1-12
CodeWars Clans	1-13
Exercises	1-16
Keep it Going!.....	1-18
Exercises	1-19
Module 2 Python.....	2-1
Section 2–1 Trying a New Language: Python	2-2
Learning a Second Language.....	2-3
Python.....	2-4
CodeWars and Python	2-6
Trying Python.....	2-8
Exercises	2-9
Module 3 Growing Your Skills: More JavaScript Katas.....	3-1
Section 3–1 Growing Your Skills.....	3-2
Getting Back to JavaScript.....	3-3
Exercises	3-4
Growing Your Skills	3-5
Exercises	3-6
Module 4 Java.....	4-1
Section 4–1 Trying a New Language: Java	4-2
Java.....	4-3
Exercises	4-6
Module 5 Growing Your Skills: More JavaScript - Part II	5-1
Section 5–1 Continuing the Complexity!	5-2
Growing Your Skills: Continued.....	5-3
Exercises	5-4
Section 5–2 Common Algorithms and Mob Coding	5-6
Common Algorithms.....	5-7
Mathematical Algorithms	5-8
Sorting Algorithms	5-9
Traveling Salesman Problem.....	5-10
Section 5–3 Mob Coding	5-11
Mob Coding	5-12
Module 6 Weekly Workshop: CodeWars Style!.....	6-1
Clan Challenge	6-3
Individual Challenges	6-4

Module 1

Algorithms and CodeWars

Section 1–1

Algorithms

Algorithms

- **What is an Algorithm?**
 - At its simplest, it is a set of instructions that solves a problem

- **Formally, an algorithm is defined as:**

"a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer."

- **In this workbook, we will practice different common computational patterns we find in computer programming**
- **Some of these algorithms are named by their type of solution, or by the way they solve a type of problem**
 - For example:
 - * Sort algorithms
 - * Search algorithms
 - * String matching and parsing

CodeWars Katas: Practice Practice Practice...

- The website CodeWars is an excellent source for programming exercises, or *katas*
- A code kata is a programming exercise that focuses on problem solving and learning new coding skills
 - Essentially, katas are puzzles
- For each code kata, you try various solutions, learn from your mistakes, and then develop even better solutions
- CodeWars is a great source for katas and is also gamified
 - Gamified means that, like a game, you gain points for completing katas, participating in discussions, etc.
- Senior developers use CodeWars to help them learn a new language or to help them learn new patterns
 - However, CodeWars also has a lot of beginner content to help practice the fundamentals
- Since you can see other people's solutions after you've completed a kata (or forfeited), you can:
 - reinforce the skills you've already learned
 - learn new approaches from seeing how others solve the same problems

Signing up for CodeWars

- Create an account on **CodeWars . com**, if you haven't already
 - When you create your account, answer "Frontend Web" to the question "Which specialized programming tasks are you most interested in?"

Which specialized programming tracks are you most interested in?
Select the engineering roles that you want to actively improve on.

Frontend Web Backend Web Mobile
 Desktop Data Science Game
 Devops & Cloud Computing Cybersecurity Academic
 Web3 Other

What best describes your current programming skill level?
Select the level that fits your coding ability based on the programming language you know the best.

Learning to Program Advanced Beginner Competent Proficient Master

Choose the languages you wish to train on:
Based on your selections above, we have underlined languages which are popular choices.

Agda	BF	C	CFML	Clojure	COBOL	CoffeeScript	CommonLisp	Coq	C++	Crystal
C#	D	Dart	Elixir	Elm	Erlang	Factor	Forth	F	F#	Go
Groovy	Haskell	Haxe	Idris	Java	JavaScript	Julia	Kotlin	<u>λ</u> . Calculus	L ₃ N	Lua

Are you interested in joining a bootcamp to assist your learning?
Codewars offers a self-guided training experience. If you wish to augment your training with more hands-on mentorship and instruction, we can recommend you to some of our favorite bootcamp programs.

I'm interested, help me find a bootcamp Not Interested I'm already enrolled in a course or bootcamp

Do you want to receive a bi-weekly newsletter with featured training content?

Yes No

- Once you get your account established, your home page will show:

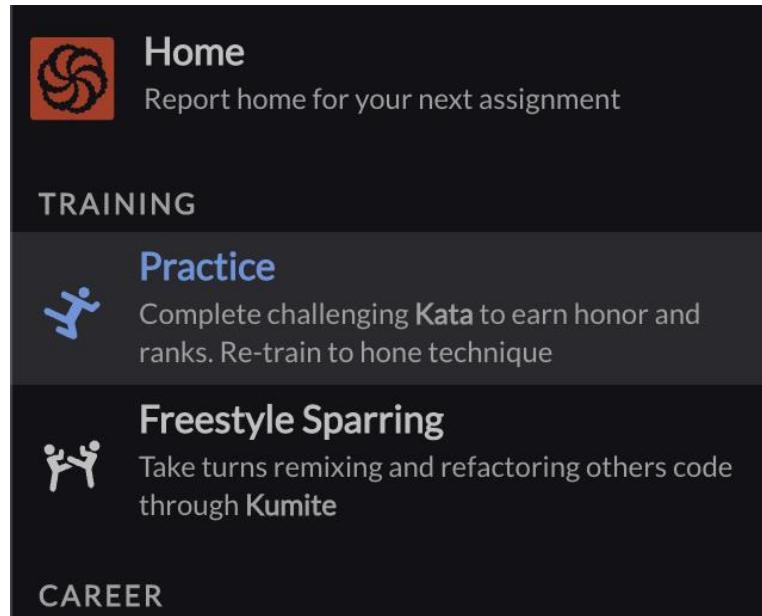
The screenshot shows the homepage of the CodeWars website. At the top, there's a navigation bar with icons for search, user profile, notifications, and account status (8 kyu, 2 notifications). Below the header, the user profile for "Skyline76!" is displayed, showing a placeholder profile picture, a rank of 8 kyu, and the name "Skyline76!". Below the profile, the user's stats are listed: Name: Unknown, Clan: Unknown, Member Since: Jun 2022, Last Seen: Jun 2022, Profiles: 0, Following: 0, Followers: 0, and Allies: 0. A button labeled "View Profile Badges" is present. To the right of the stats, there's a sidebar with an advertisement for "GoJS" and a link to "Ads via Carbon". Below the stats, there's a section titled "Progress" with icons for progress bars and arrows. Under "Progress", the user's rank is listed as 8 kyu, Honor: 2, and Total Completed Kata: 1. To the right of "Progress", there are sections for "Languages" (Total Languages Trained: 1, Highest Trained: JavaScript (8 kyu), Most Recent: JavaScript) and "Up Votes" (Up Votes: 1, Upgrade To Red). The main menu below the stats includes tabs for Stats, Kata, Solutions, Translations, Collections, Kumite, Social, and Discourse.

CodeWars Coding Solutions

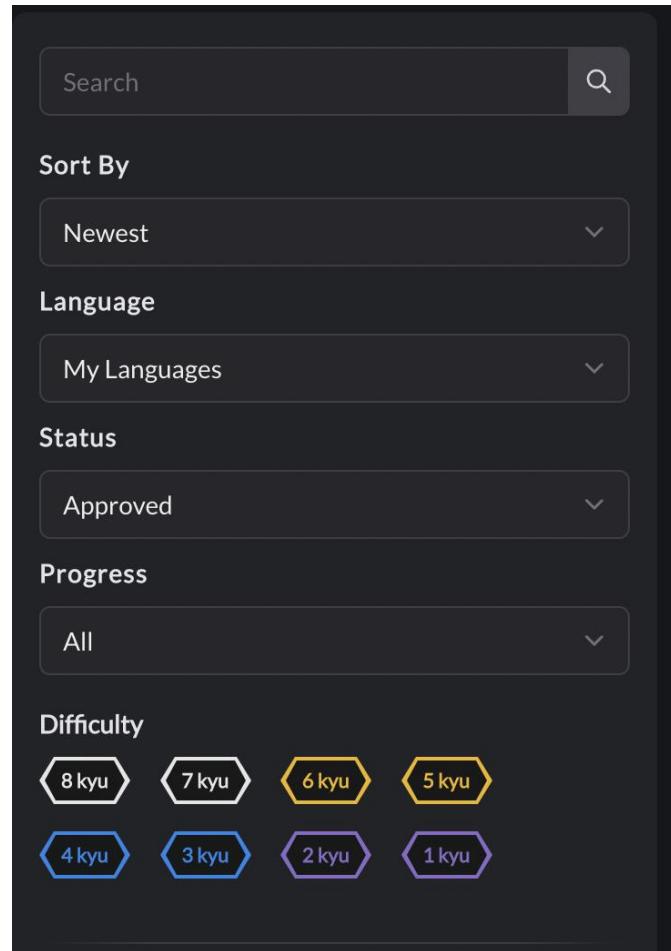
- You'll see a lot of **unreadable code** on CodeWars
 - Many developers simply use CodeWars as a game
 - Writing solutions in a readable way should be a fun challenge yet many people will not write readable, production-ready code
- Some developers on CodeWars are obsessed with the idea of "code golf"
 - Code golf is where you try to solve a problem in as few characters as possible
 - * This will add another "gaming" quality to the exercises -
 - * Although, this might be the opposite approach to what you would do on-the-job!
- While you are still struggling with syntax and learning algorithms, we suggest you avoid "code golf"
 - It will end up teaching you JavaScript features and tricks you would likely be discouraged from using on-the-job
- Shoot for some level of comprehensibility in your kata code!

Getting Started

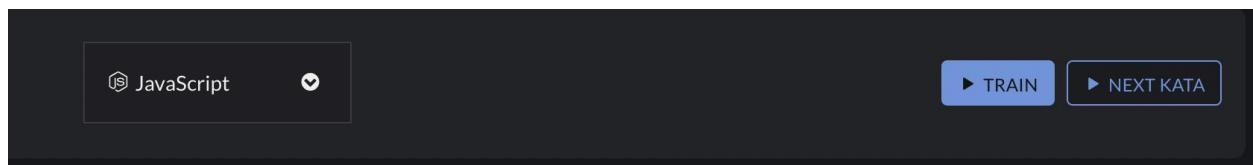
- To search for a new CodeWars problem, simply click on the 'Practice' menu in the sidebar



- The default practice screen will list all the available katas
- You can also search for Katas to solve and two filtering options worth noting are: 'Sort By' and 'Difficulty'
 - The 'Sort by' allows you to change how the Katas are displayed
 - * For example, sorting by easiest will display all available Katas by their difficulty
 - The 'Difficulty' allows you to filter by how difficult a problem is categorized as.
 - * An 8 KYU difficulty is considered their easiest problems to solve



- After a CodeWars kata is selected, you select 'Train' in the top right-hand corner to try and solve it
 - Along with the ability to attempt to solve the Kata, a selection of the language used is available in a dropdown.



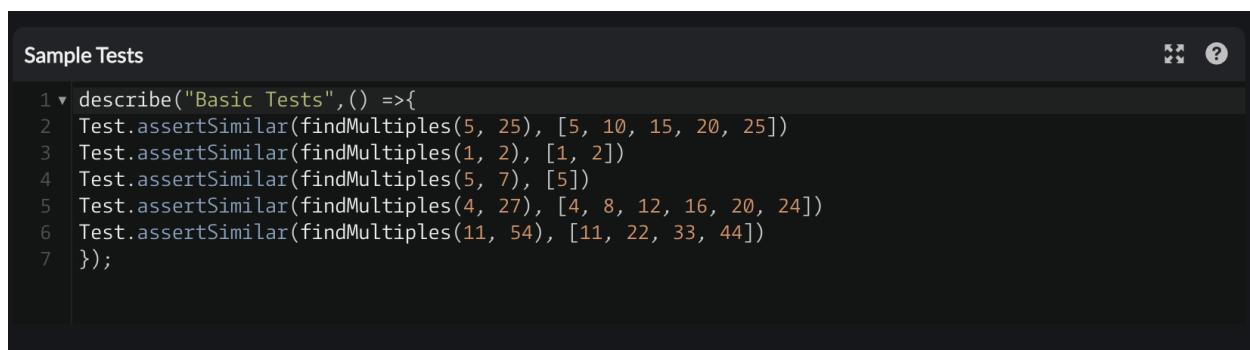
Solving a Kata

- In CodeWars, you code a solution for the kata by filling in a skeletal function

Example

```
function findMultiples(integer, limit) {  
    // your write your code here  
}
```

- When you are ready, CodeWars has an infrastructure in place to call your function
 - Essentially, it will test your code by calling it several times with different inputs and confirming the returned output is correct
 - These tests define how your solution attempt is initially validated
- The 'Sample Tests' are listed underneath where the problem attempt is made

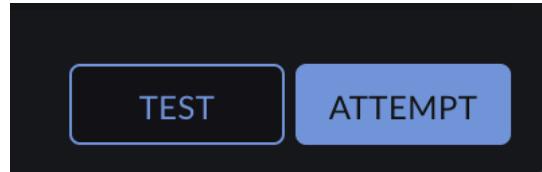


A screenshot of a terminal window titled "Sample Tests". The window contains a block of JavaScript code. The code defines a function named "findMultiples" and uses the Mocha testing framework's "assertSimilar" method to check its behavior against various inputs. The code is as follows:

```
1 describe("Basic Tests", () =>{  
2     Test.assertSimilar(findMultiples(5, 25), [5, 10, 15, 20, 25])  
3     Test.assertSimilar(findMultiples(1, 2), [1, 2])  
4     Test.assertSimilar(findMultiples(5, 7), [5])  
5     Test.assertSimilar(findMultiples(4, 27), [4, 8, 12, 16, 20, 24])  
6     Test.assertSimilar(findMultiples(11, 54), [11, 22, 33, 44])  
7 });
```

- The first parameter in `Test.assertSimilar()` calls your function and the second parameter defines what the expected return value is

- NOTE: The mechanics of `Test.assertSimilar()` and other testing functions is beyond the scope of this class -- but we are sure you can understand the process at a high level
- When you hit the 'TEST' button, it runs those tests



- However, there are more tests that you DO NOT see!
- When you hit the 'ATTEMPT' button, it runs the sample tests, and it also runs additional tests to confirm your solution is valid

Debugging Your Kata

- Using `console.log()` in JavaScript may not behave as expected
 - Unlike in a script, where the output is available in the browsers' development console, `console.log()` output is available after running 'TEST'
- Within the output screen of each of the attempted tests, the log inserted will be displayed
 - This can be very useful when working out a solution to a problem.

```
Solution
1 ▾ function findMultiples(integer, limit) {
2   console.log("where does this appear?")
3 }
4
```

```
Time: 661ms  Passed: 0  Failed: 5  Exit Code: 1

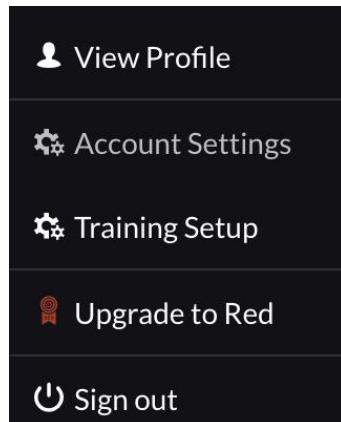
Test Results:

Basic Tests
  ▾ Log
    where does this appear?

  ! Expected: '[5, 10, 15, 20, 25]', instead got:
    'undefined'
```

CodeWars Clans

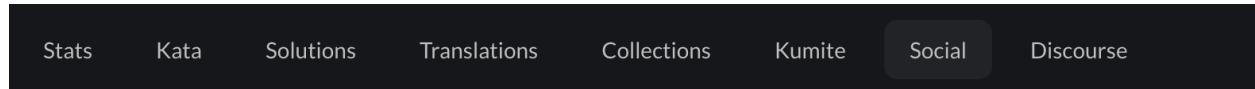
- Do you enjoy a healthy dose of competition? Want to solve problems with your friends?
- CodeWars has a feature that will allow you and others to band together as allies to show your progress
- To add your clan: navigate to the top left-hand dropdown for your user, and select 'Account Settings'



- Within this page, all you'll need to do is input the clan you want to join

A screenshot of a mobile application's interface for joining a clan. At the top is a text input field with the placeholder "Clan (Company, School or Organization)". Below the input field is a large, empty rectangular area with a light gray border. Underneath this area, the text "If set, you will be allied with users with the same clan ([docs](#))" is displayed. At the bottom of the screen is a dark gray bar containing a checked checkbox and the text "Remove the previous clan members from allies when changing clans".

- Want to see the allies in your clan? Navigate over to the social tab in your profile!



Section 1–2

Practicing the Essentials

Exercises

EXERCISE 1

Begin by trying two different exercises. In the first, you will simply code a Hello World function. In the second, you will be given code that you have to debug!

- [Function 1 - Hello World](#)
- [Grasshopper - Debug sayHello](#)

EXERCISE 2

This next set of katas starts us off slow and will let you realize how much JavaScript you do know!

- [Grasshopper - Function syntax debugging](#)
- [Jenny's secret message](#)
- [Grasshopper - Combine strings](#)
- [Grasshopper - If/else syntax debug](#)
- [Grasshopper - Variable Assignment Debug](#)

EXERCISE 3

Let's keep this up with some more katas!

- [Grasshopper - Messi Goals](#)
- [Grasshopper - Order of operations](#)
- [Keep Hydrated!](#)

EXERCISE 4

If you finish early, search CodeWars for similar katas. The more you code, the better you will get!

Keep it Going!

- **How were those first katas you completed?**
 - Easy?
 - Hard?
- **If you said hard, it is just because it is a new process**
- **This week, you will complete many, many katas**
 - Each will require you to solve a coding problem, or perhaps to find a coding error
- **In this next set of problems, you will continue to do so**
 - Most of these next problems work with numbers

Exercises

EXERCISE 1

We've gathered a set of simple katas below. Work on them one at a time. You will eventually get into a good pattern of reading the problem, thinking, and then coding the solution.

- Is it even?
- Return Negative
- Opposite number
- Basic Mathematical Operations
- Multiply
- Function 2 - squaring an argument
- Function 3 - multiplying two numbers
- Is it a number?
- Even or Odd?
- Return the day

EXERCISE 2

If you have time, keep finding and solving katas!

Module 2

Python

Section 2–1

Trying a New Language: Python

Learning a Second Language

- **Maybe it was scary when you first joined this academy and started learning JavaScript?**
 - We went slow while you learned the syntax of JavaScript and, more importantly, the process of translating a problem statement into code
 - However, many languages are VERY similar
- **In fact, most programmers know several languages**
 - The more jobs you have, and the longer you are in this profession, the more languages you know
 - You quickly learn that the process of translating a problem statement into code is the same regardless of the language
- **Take a few minutes to look at this comparison of popular languages**
 - The link below shows the comparison statements (ifs, etc)
[https://en.wikipedia.org/wiki/Comparison_of_programming_languages_\(basic_instructions\)#Conditional_statements](https://en.wikipedia.org/wiki/Comparison_of_programming_languages_(basic_instructions)#Conditional_statements)
 - You can compare other aspects of the language by clicking on the other language options on the left of the page
- **So, what language should we take a peek at to start with? Let's try Python!**

Python

- **Python is a general-purpose programming language**
 - It can solve a multitude of problems
- **Python is often used to:**
 - build websites
 - automate tasks
 - perform data analysis and machine learning
- **The official documentation for Python can be found at:**
<https://www.python.org/>
- **But a great place to go find syntax and example code can also be found at:** <https://www.w3schools.com/python/>
- **Python may look a little different, but don't panic! It's got the same elements you are used to, including:**
 - variables and operators
 - functions
 - ifs
 - loops, arrays

- Python functions are defined using the `def` keyword

```
def greet():
    return "hello world!"
```

- Variables look familiar

```
name = "Pursalane"
age = 11
print(name)
print(age)
```

- Operators are the same

```
x = 11
y = 22
z = (x + y) / x
print(z)
```

- If statements are readable, but the syntax is different

- Indentation is required in Python

```
guess = 33
targetValue = 55
if guess < targetValue:
    print("Your guess is too low!")
elif guess == targetValue:
    print("You guessed the right value!")
else:
    print("Your guess is too high!")
```

CodeWars and Python

- **CodeWars gives you the ability to solve a single problem in a multitude of languages**
 - Using the previously solved kata 'Keep Hydrated', you can see a comparison of how JavaScript and Python structure their functions and test functions
- **CodeWars provides the starter:**

JavaScript

```
Solution
1 | function litres(time) {
2 |   return 0;
3 | }
```

Python

```
Solution
1 | def litres(time):
2 |   return 0
```

- **Take note of the difference in syntax**
 - While both languages have in effect, the same functionality, the way they go about it can be very different

- Some observable differences are found in how a language handles semicolons, curly braces, and indentation
- Also note the differences in the code required to run the tests

JavaScript

```

1 const chai = require("chai");
2 const assert = chai.assert;
3 chai.config.truncateThreshold=0;
4
5 ▼ describe('Fixed tests', () => {
6 ▼   it('Tests', () => {
7     assert.strictEqual(litres(2), 1, 'should return 1 litre');
8     assert.strictEqual(litres(1.4), 0, 'should return 0 litres');
9     assert.strictEqual(litres(12.3), 6, 'should return 6 litres');
10    assert.strictEqual(litres(0.82), 0, 'should return 0 litres');
11    assert.strictEqual(litres(11.8), 5, 'should return 5 litres');
12    assert.strictEqual(litres(1787), 893, 'should return 893 litres');
13    assert.strictEqual(litres(0), 0, 'should return 0 litres');
14  });
15});
```

Python

```

1 import codewars_test as test
2 from solution import litres
3
4 @test.describe('Fixed tests')
5 def basic_tests():
6     @test.it('Basic Test Cases')
7     def basic_test_cases():
8         test.assert_equals(litres(2), 1, 'should return 1 litre')
9         test.assert_equals(litres(1.4), 0, 'should return 0 litres')
10        test.assert_equals(litres(12.3), 6, 'should return 6 litres')
11        test.assert_equals(litres(0.82), 0, 'should return 0 litres')
12        test.assert_equals(litres(11.8), 5, 'should return 5 litres')
13        test.assert_equals(litres(1787), 893, 'should return 893 litres')
14        test.assert_equals(litres(0), 0, 'should return 0 litres')
```

Trying Python

- Take a few minutes to view this website:
<https://www.w3schools.com/python/default.asp>
- Specifically, look at:
 - Python variables
 - Python comments
 - Python If...Else
 - Python functions
 - Python examples (scroll towards the bottom of the page)
- Once you've got your bearings, try solving the following exercises in Python!
 - You may be surprised to find out you can code in a whole different language than JavaScript!

Exercises

EXERCISE 1

As you saw earlier, you can complete katas in several different languages. In this exercise set, try to do a few simple katas in Python

- Function 1 - hello world
- Function 3 - multiplying two numbers
- Quarter of the year
- Convert boolean values to strings 'Yes' or 'No'

EXERCISE 2

If you have time, find some more simple katas and solve them in Python! Sure, you have to look up a lot of syntax on Google... but do you know what you are looking for? Do you recognize it when you find it?

Module 3

**Growing Your Skills:
More JavaScript Katas**

Section 3–1

Growing Your Skills

Getting Back to JavaScript

- Most people take longer than a few hours to conquer a programming language
- But how did it feel? Were you able to read the code? How many katas did you get working in Python?
- Now it is time to get back to JavaScript where you feel more comfortable
- But let's kick up the difficulty a little bit

Exercises

EXERCISE 1

We've gathered a set of simple katas below. In JavaScript, you could solve them quickly. But here's the rub -- we are asking you to do it in Python!

- Is it even?
- Return Negative
- Opposite number
- Basic Mathematical Operations
- Multiply
- Function 2 - squaring an argument
- Function 3 - multiplying two numbers
- Is it a number?
- Even or Odd?
- Return the day

EXERCISE 3

Let's take the complexity up a bit. Can you solve these?

- Determine the date by the day number

EXERCISE 2

Search for a more slightly complicated katas and solve them

Growing Your Skills

- In this section, we will continue solving katas, but their difficulty will increase
- These katas that require problem solving with skills that include:
 - string manipulation
 - loops
 - arrays
- But mostly what makes these katas stand out is that the algorithms to solve them become a little more complex

Exercises

EXERCISE 1

The following warm-up katas will start today's series. Take some time to read them, think about them, and then code them

- [Quarter of the year](#)
- [Convert boolean values to strings 'Yes' or 'No'](#)

EXERCISE 2

To solve these katas, you will need to rely upon your string and array skills. If you need a refresher, you can find a refresher on [string](#) and [array](#) methods at MDN!

- [Thinkful - String Drills: Repeater](#)
- [MakeUpperCase](#)
- [Remove First and Last Character](#)
- [Sentence Smash](#)
- [Reversed Strings](#)
- [A Needle in the Haystack](#)

EXERCISE 3

These katas emphasize looping skills.

- [Unfinished Loop - Bug Fixing #1](#)
- [Grasshopper - Summation](#)
- [Counting valleys](#)

EXERCISE 4

If you finish early, search for and solve other interesting and more difficult katas.

Module 4

Java

Section 4–1

Trying a New Language: Java

Java

- Java is a very popular, mature, high-level object-oriented programming (OOP) language
- Java is an open-source language
- Java's strength is its platform independence
 - Programs can run on many different types of computers as long as the computer has a Java Runtime Environment (JRE) installed.
- Java can be used for many things, including:
 - building websites
 - building Android applications
 - creating back-end applications
- A few very popular applications written in Java include:
 - Netflix
 - Spotify
 - LinkedIn
 - Uber
- The official documentation for Java can be found at:
<https://docs.oracle.com/en/java/>

- But a great place to go find syntax and example code can also be found at: <https://www.w3schools.com/java/>
- Java may look a little different, but don't panic! It's got the same elements you are used to, including:
 - variables and operators
 - functions
 - ifs
 - loops, arrays
- However, because Java is an object-oriented language, all the code you write is organized into classes
 - Classes are a way of organizing code around the types of data it works with
- Classes usually have nouns as names and represent types of things (ex: Employee, Satellite, Song)
 - The classes then define the data attributes that are required by the type (ex: employee name, job title, pay)
 - Classes also contain the functions (or methods) that operate upon that data

```
public class HelloWorld {  
    public static String greet() {  
        return "hello world!";  
    }  
}
```

- **Variables look familiar, but they must be typed**
 - Typed means you decide ahead of time what type of data will be placed in the variables? Strings? ints (whole numbers)? doubles (numbers with a decimal point)?

```
String name = "Pursalane";
int age = 11;
System.out.print(name);
System.out.print(age);
```

- **Operators are the same**

```
double price = 102.33;
double tax = 8.77;
double total = price + tax;
System.out.print(total);
```

- **If statements look very similar to JavaScript**

```
int guess = 33;
int targetValue = 55;
if (guess < targetValue) {
    System.out.print("Your guess is too low!");
}
else if (guess == targetValue) {
    System.out.print("You guessed the right value!");
}
else {
    System.out.print("Your guess is too high!");
}
```

- However, string comparison is different

```
String choice = "YES";

if (choice.equals("YES")) {
    System.out.print("Your action will be performed!");
}
```

Exercises

EXERCISE 1

As you saw earlier, you can complete katas in several different languages. In this exercise set, try to do a few simple katas in Java

- Function 1 - hello world
- Function 3 - multiplying two numbers
- Quarter of the year
- Convert boolean values to strings 'Yes' or 'No'

EXERCISE 2

If you have time, find some more simple katas and solve them in Java! Again, you have to look up a lot of syntax on Google... but do you know what you are looking for? Do you recognize it when you find it?

Module 5

**Growing Your Skills:
More JavaScript - Part II**

Section 5–1

Continuing the Complexity!

Growing Your Skills: Continued

- Learning to program is a lot like:
 - learning algebra
 - learning the guitar
 - becoming a professional athlete
 - becoming a sculptor
- Can you guess why? They all take practice, practice, practice...
- And that's what you are going to do all week!

Exercises

EXERCISE 1

We've gathered more katas. They are certainly getting a little more complex!

- [Calculate average](#)
- [Reversed Sequence](#)
- [Sum of positive](#)
- [No oddities here](#)
- [Shortest Word](#)
- [Remove duplicate words](#)
- [Complete The Pattern #1](#)

EXERCISE 2

Okay, these aren't a different category from above -- we just broke the list in half so it wouldn't be so intimidating! Keep coding!

- [If you can't sleep, just count sheep!!](#)
- [Who likes it](#)
- [Disemvoweled trolls](#)
- [Card Counting](#)
- [List filtering](#)

EXERCISE 3

If you get this far, don't stop! Find some more difficult katas! Keep coding!

Section 5–2

Common Algorithms and Mob Coding

Common Algorithms

- Throughout this workbook, you have worked through many different algorithms
- You can classify these algorithms based on what they require
 - By no means are we listing every type of algorithm in the following pages
- After reading through the various algorithm types, you will realize that you've already worked on many of them

Mathematical Algorithms

- Mathematical algorithms can be grouped into sub-categories
 - Counting / Summation: Algorithms whose purpose is counting
 - * How many Katas have you completed so far?
 - * How many sales were completed for a given day?
 - * How many new subscribers have we had this week?
 - Min/Max: Algorithms that determine a minimum or maximum value within a set of numbers
 - * What is the most expensive item sold today?
 - * What is the lowest age in the conference?
 - * What student has the best average test score?

Sorting Algorithms

- **Sorting algorithms sort a set of data**
 - However, there are several sub-categories of sort algorithms because of the different sort techniques
- **Each sort type has its own use case**
 - For instance, the size of the data and the required speed of the sort can determine the most appropriate sort
- **Sorts include:**
 - **Bubble Sort:**
 - * This type of sorting Algorithm attempts to order a set of data by comparing the adjacent piece of data, putting the largest value in the correct order.
 - **Insertion Sort:**
 - * Taking longer than a bubble sort, this type of Algorithm requires setting a sorted and unsorted set of data. Keeping track and putting in place the sorted data from the unsorted stack.
 - **Selection Sort:**
 - * The minimum first approach. This type of sorting Algorithm will move through the set of data and move the lowest value to the beginning.
- **There are more sorting algorithms than the few listed here**
 - Learn more at:
<https://www.geeksforgeeks.org/sorting-algorithms/>

Traveling Salesman Problem

- The Traveling Salesman problem differs from the mathematic and sorting algorithms
- In this type of problem, you have a set of cities or locations and must find the fastest route to reach all destinations
 - This problem is coined after the door-to-door salesman era where salesmen would have various neighborhoods and towns they must cover.
 - * To do this most efficiently, determining the fastest route to all locations is crucial
- Read more about TSP and how to implement a solution at [GeeksforGeeks](#) and [Google Developers](#)

Section 5–3

Mob Coding

Mob Coding

- Within the development world, programming is often an individual activity
- However, sometimes we must work together as a team to collaborate on a solution
- "Mob coding" is the term used when a group of programmers come together, sharing one screen, and attempt to collaboratively work through a problem
 - Discussing the design of an algorithm is often more effective when mob coding!

Module 6

**Weekly Workshop:
CodeWars Style!**

Section 6–1

Challenge Yourself and
Your Cohort Members

Clan Challenge

- **Form a clan of 2-3 people for this week's Coding Challenge!**
 - Complete the following mob coding challenges *together*
- **In your breakout room:**
 - talk about the kata
 - discuss possible solutions
 - whoever is typing should share their screen and your class should practice team coding
 - * This doesn't mean one or more individuals can't do research on the side to help the team figure out a solution
- **Your team challenge contains a few katas:**
 - [Alphabetic Addition](#)
 - [Credit Card Mask](#)
 - [JavaScript filter - 2](#)

Individual Challenges

- When clan competes the katas, you can revert back to individual work
- Complete these additional katas:
 - Halving Sum
 - Power of Two
 - Sort by Binary Ones
 - Validate Credit Card Number
 - Halving Sum
- Finished with all the Katas? Ready for more of a challenge? *See if you can reach level 6 or even 5 KYU!*