BSCODE

KEEP LEARNING

EXERCISES

Ljupco Shemov

CODE ACADEMY codeacademy.mk

CONTENTS

INTRODUCTION

I'm a believer in learning by doing. Programming exercises are a useful tool to practice and improve your coding skills. After all, practice makes one, if not perfect, at least pretty darn good. One of the best ways to learn, practice, and enhance your skills with this code editor is through basic exercises and practice projects. These will allow you to understand the language better.

Learning to program means learning how to solve problems using code. Conceptually it is not very difficult to write a program that solves a problem that you can solve yourself. The skill you need to acquire is thinking very precisely about how you solve the problem and breaking it down into steps that are so simple that a computer can execute them. I encourage you to first solve a few instances of a problem by hand and think about what you did to find the solution. For example, if the task is sorting lists, sort some short lists yourself. A reasonable method would be to find the smallest element, write it down and cross it out of the original list and repeat this process until you have sorted the whole list. Then you have to teach the computer 1) how to find the smallest element, 2) how to write it down, 3) how to cross it out, and wrap this in a loop. Then continue this task breakdown process until you're confident you know how to write the necessary program.

To make good progress in your programming task, you need to test your work as early and as thoroughly as possible. Everybody makes mistakes while programming and finding mistakes in programs consumes a very large part of a programmer's workday. Finding a problem in a small and easy piece of code is much simpler than trying to spot it in a large program. This is why you should try to test each sub-task you identified during your task breakdown by itself. Only after you're confident that each part works as you expect you can attempt to plug them together. Make sure you test the complete program as well, errors can creep in the way the different parts interact. You should try to automate your tests. The easier it is to test your program, the freer you are in experimenting with changes.

Hope, these exercises help you to improve your coding skills.

CLOCK

Building your own digital clock with JavaScript is a relatively easy small project to practice variables and simple if loops.

Again, you want to use some CSS to customize the design and make your JavaScript clock look amazing.

HTML

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Clock | Exercise</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div id="MyClockDisplay" class="clock" onload="showTime()"></div>

    <script src="main.js"></script>
</body>
</html>
```

CSS

```css
body {
  background: black;
}

.clock {
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translateX(-50%) translateY(-50%);
  color: #17D4FE;
  font-size: 60px;
  font-family: Orbitron;
  letter-spacing: 7px;
}
```

JAVASCRIPT

1. First, we need to create a function called ***showTime*** without parameters.

2. Inside the block scope of the function we will start by declaring and assigning a ***date*** variable with the value of ***new Date().***

3. After declaring the date variable we need to declare and assign three more variables:
    a. One for getting the hours named ***h*** with the value of ***date.GetHours().***
    b. One for the minutes named ***m*** with the value of ***date.GetMinutes().***
    c. And one for the seconds named ***s*** with the value of ***date.GetSeconds().***

4. Then we will declare one more variable named ***session*** with the value of ***"AM".***

5. Now we can create two if statements needed for the ***AM*** and ***PM***
    a. The first if statement needs to check if the ***h*** variable is equal to ***0***. If it is then we need to set the value of ***h*** to ***12***.
    b. The second if statement needs to check if the ***h*** variable is bigger than ***12***. If it is then we need to set the value of ***h*** to ***h – 12*** and after set the session variable value to ***"PM"***.

6. The next three statements we will create for assigning new values to the variables ***h, m, s*** using the ternary operator:
    a. First we will assign a new value to the ***h*** variable that equals
    (h < 10) ? "0" + h : h
    b. Second, we will assign a new value to the ***m*** variable that equals
    (m < 10) ? "0" + m : m
    c. Third, we will assign a new value to the ***s*** variable that equals
    (s < 10) ? "0" + s : s

7. Now we declare and assign a variable called ***time*** with the value that sets the time in a string: ***h + ":" + m + ":" + s + " " + session***

8. Then we change the ***innerText*** property of the HTML element with the id ***"MyClockDisplay"*** and assign the value of the ***time*** variable.

9. The last statement we need is ***setTimeout()*** to call the function every second
   (The first argument in the ***setTimeout*** needs to be ***showTime*** the name of the function we created and the second needs to be ***1000*** which means it will call the function every second)

10. For everything to work we need to call the function for the first time then it will call itself.

    Enjoy your time now