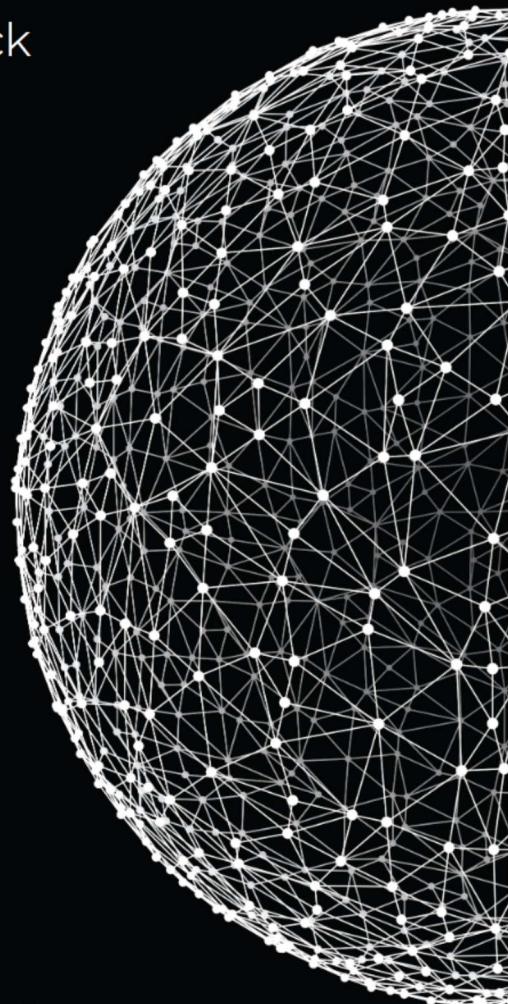


Sprint 02

Half Marathon Full Stack

November 15, 2021



ucode connect

Contents

Engage	2
Investigate	3
Act Basic: Task 00 > Hello, JavaScript!	5
Act Basic: Task 01 > What type of data?	7
Act Basic: Task 02 > Superhero name maker	9
Act Basic: Task 03 > What kind of idiom?	11
Act Basic: Task 04 > Numbers	13
Act Basic: Task 05 > Total price	15
Act Basic: Task 06 > Greeting	17
Act Basic: Task 07 > Date	18
Act Advanced: Task 08 > Place all	20
Act Advanced: Task 09 > Clone of Steve Rogers	22
Act Advanced: Task 10 > Word for word	24
Act Advanced: Task 11 > Brackets	26
Act Advanced: Task 12 > Hulk Closure	27
Act Advanced: Task 13 > Calculator	30
Share	32

ucode connect

Engage

DESCRIPTION

Welcome to Sprint02!

You now have a good foundation in HTML and CSS, and you can develop attractive and informative web pages. However, so far, you have been creating only static pages that always look and behave the same each time they are loaded into the browser.

A dynamic website contains both client-side and/or server-side scripting to generate changeable content. You can use JavaScript, or other scripting languages, to dynamically change the data of a web page.

Dynamic web pages include the following features:

- content that can be changed dynamically
- dynamic positioning of web page elements
- dynamic style

With just a few lines of JavaScript, your web page will become dynamic and functional.

So, let's get started with learning JS!

BIG IDEA

Dynamic web pages.

ESSENTIAL QUESTION

How to create an interactive web page?

CHALLENGE

Explore JavaScript basics.



Sprint 02 | Half Marathon Full Stack > 2

Investigate

GUIDING QUESTIONS

We invite you to find answers to the following questions. By researching and answering them, you will gain the knowledge necessary to complete the challenge. To find solutions, ask the students around you and search the internet. We encourage you to ask as many questions as possible. Note down your findings and discuss them with your peers.

- What are the differences between Web 1.0 and Web 2.0?
- What does DHTML mean?
- What is JavaScript usually used for?
- What is the difference between JavaScript and Java?
- Which extension do JavaScript files use?
- Is Javascript a compiled or an interpreted language?

GUIDING ACTIVITIES

Complete the following activities. Don't forget that you have limited time to overcome the challenge. Use it wisely. Distribute tasks correctly.

- Take into account: most of your questions about JS can be answered [here](#) and [here](#).
- Read about debugging inside a browser (e.g [Chrome](#)).
- Get acquainted with [JavaScript Style Guide and Coding Conventions](#) and [JavaScript Best Practices](#).
- Clone your git repository, issued on the challenge page in the LMS.
- Employ the full power of P2P by brainstorming with other students.

ANALYSIS

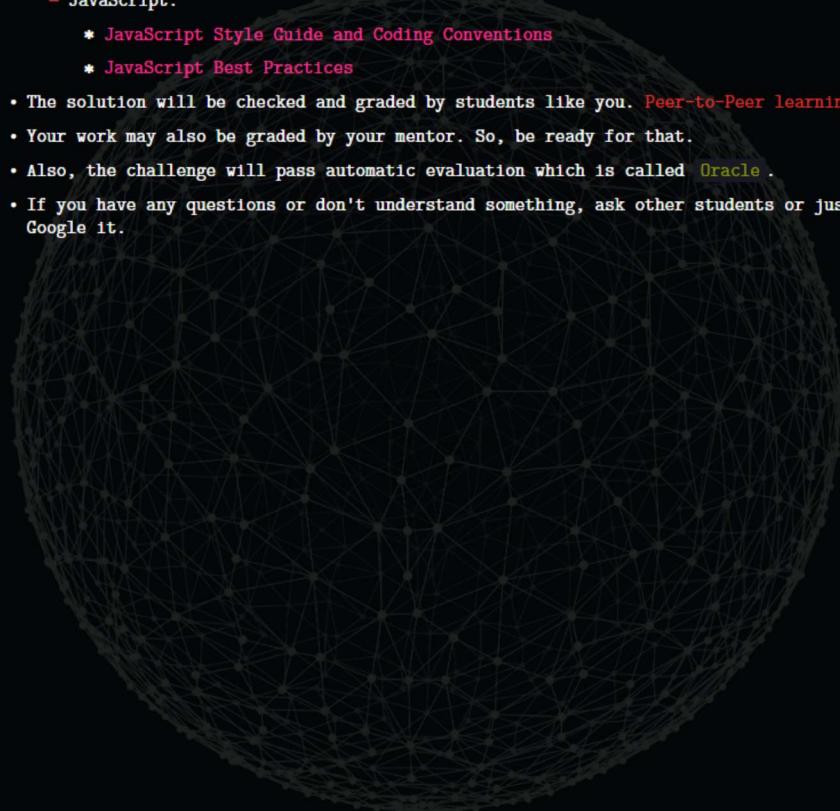
Analyze your findings. What conclusions have you made after completing guiding questions and activities? In addition to your thoughts and conclusions, here are some more analysis results.

- Be attentive to all statements of the story. Examine the given examples carefully. They may contain details that are not mentioned in the task.
- All tasks are divided into [Act Basic](#) and [Act Advanced](#). This means that the complexity of the tasks increases gradually. Try to complete all tasks to get maximum points and more knowledge.
- Analyze all information you have collected during the preparation stages.
- Perform only those tasks that are given in this document.
- Submit only the specified files in the required directory and nothing else. Garbage shall not pass.
- Pay attention to what is allowed. Use of forbidden stuff is considered a cheat and your challenge will be failed.
- The web page in the browser must open through [index.html](#).
- The scripts must be written outside the HTML file - in a separate JS file (script.js).



Sprint 02 | Half Marathon Full Stack > 3

- You can always use the `Console` panel to test and catching errors.
- Complete tasks according to the rules specified in the following style guides:
 - HTML and CSS: [Google HTML/CSS Style Guide](#). As per section 3.1.7 Optional Tags, it doesn't apply. Do not omit optional tags, such as `<head>` or `<body>`
 - JavaScript:
 - * [JavaScript Style Guide and Coding Conventions](#)
 - * [JavaScript Best Practices](#)
- The solution will be checked and graded by students like you. [Peer-to-Peer learning](#).
- Your work may also be graded by your mentor. So, be ready for that.
- Also, the challenge will pass automatic evaluation which is called [Oracle](#).
- If you have any questions or don't understand something, ask other students or just Google it.



ucode connect

Sprint 02 | Half Marathon Full Stack > 4

Act Basic: Task 00

NAME

Hello, JavaScript!

DIRECTORY

t00_hello_javascript/

SUBMIT

index.html, js/script.js

ALLOWED FUNCTIONS

alert()

DESCRIPTION

Create a web page that runs two JS scripts. One of the scripts must be written inside the HTML file, and the other - outside, as a separate JS file.

The script that is inside the HTML file

- shows a message 'Hello JavaScript from inside!'
- contains a 1-row comment with a description of the `alert` function

Keep in mind that the [Google HTML/CSS Style Guide](#) advises against the practice of mixing HTML and CSS or JS in one document. Implement it here as an exercise in order to know how it can be done, but avoid doing it in the future.

The script that is outside the HTML file, in the JS file

- contains a 2-row comment
- has a variable with the value 'Hello JavaScript from outside!'
- shows a message with the value of the variable

SYNOPSIS

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <title>Hello, JavaScript!</title>
    <meta name="description" content="t00. Hello, JavaScript!">
</head>

<body>
    <h1>Hello, JavaScript!</h1>
    <script>
```

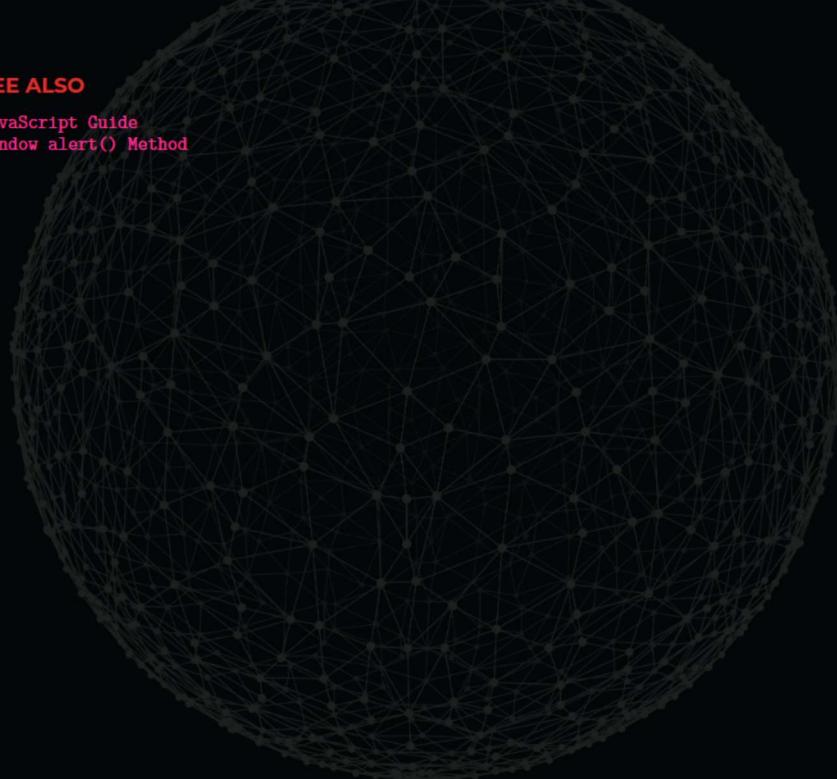


Sprint 02 | Half Marathon Full Stack > 5

```
// your code here  
</script>  
  
<script /*your code here*/></script>  
</body>  
  
</html>
```

SEE ALSO

[JavaScript Guide](#)
[Window.alert\(\) Method](#)



ucode connect

Sprint 02 | Half Marathon Full Stack > 6

Act Basic: Task 01

NAME

What type of data?

DIRECTORY

t01_what_type_of_data/

SUBMIT

index.html, js/script.js

ALLOWED FUNCTIONS

alert()

DESCRIPTION

Create a JS file that will be included into the HTML page written in the SYNOPSIS.
The script must:

- create variables for the following data types in JavaScript and assign the appropriate values:
 - Number
 - BigInt
 - String
 - Boolean
 - Null
 - Undefined
 - Object
 - Symbol
 - Function
- display, at once, all the variable names and their data types in the following format: `variable_name is data_type\n with alert() method`

Note: in this task `typeof` will help you. One of the outputs may surprise you, because there is a known error in the JS language. Don't be scared and read SEE ALSO.

SYNOPSIS

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <title>What type of data?</title>
```

ucode connect

Sprint 02 | Half Marathon Full Stack > 7

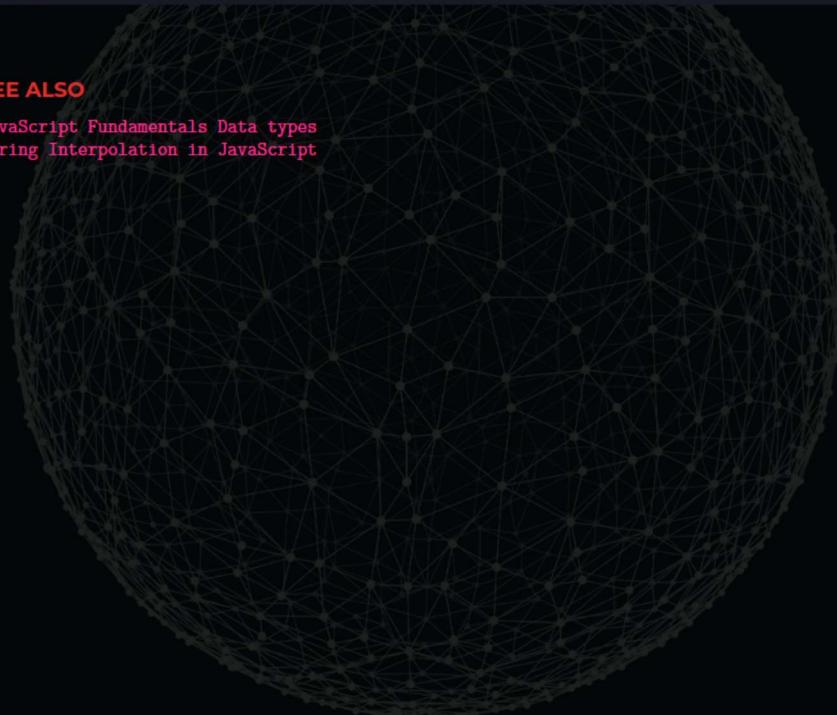
```
<meta name="description" content="t01. What type of data?">
</head>

<body>
  <h1>What type of data?</h1>
  <script src="js/script.js"></script>
</body>

</html>
```

SEE ALSO

[JavaScript Fundamentals Data types](#)
[String Interpolation in JavaScript](#)



ucode connect

Sprint 02 | Half Marathon Full Stack > 8

Act Basic: Task 02

NAME

Superhero name maker

DIRECTORY

t02_superhero_name_maker/

SUBMIT

index.html, js/script.js

ALLOWED FUNCTIONS

alert(), prompt(), RegExp.*

DESCRIPTION

Create a JS file that will be included into the HTML page written in the SYNOPSIS.

Make a script that generates superhero names based on input.

The script must:

- prompts the user to enter input three times:
 1. to enter an animal name: What animal is the superhero most similar to?
 - Input requirements: length <= 20, only one word that contains only letters
 2. to enter gender: Is the superhero male or female? Leave blank if unknown or other.
 - Input requirements: accepts only male, female, or blank (not case sensitive)
 3. to enter age: How old is the superhero?
 - Input requirements: length <= 5, only digits, cannot start with a zero
- checks input for validity using regular expression (also known as regex)
- if the input is not valid, displays an error message using alert and stops executing
- generates a description for the superhero depending on the entered gender and age:
 - boy if male + younger than 18
 - man if male + at least 18
 - girl if female + younger than 18
 - woman if female + at least 18
 - kid if gender was left blank + younger than 18
 - hero if gender was left blank + at least 18
- displays The superhero name is: [enteredAnimal]-[description]!

So, for example, if the user entered: "bat", "Male", "25", the message will be:

The superhero name is: bat-man!



Sprint 02 | Half Marathon Full Stack > 9

SYNOPSIS

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <title>Superhero name maker</title>
  <meta name="description" content="t02. Superhero name maker">
</head>

<body>
  <h1>Superhero name maker</h1>

  <script src="js/script.js"></script>
</body>

</html>
```

SEE ALSO

Making decisions in your code – conditionals
Regular expressions



Sprint 02 | Half Marathon Full Stack > 10

Act Basic: Task 03

NAME

What kind of idiom?

DIRECTORY

t03_what_kind_of_idiom/

SUBMIT

index.html, js/script.js

ALLOWED FUNCTIONS

alert(), prompt(), Number.*

DESCRIPTION

Create a JS file that will be included into the HTML page written in the SYNOPSIS.
The script must:

- call a `prompt()` method and take a number from 1 to 10 as an input value
- check that the input value is a number, and exactly from 1 to 10. If the input value will be not 1-10 - the `prompt()` method must ask for a number again
- show an idiom with `alert()` method

The idiom must depend on the input value in the following way:

- 1 - Back to square 1
- 2 - Goody 2-shoes
- 3 or 6 - Two's company, three's a crowd
- 4 or 9 - Counting sheep
- 5 - Take five
- 7 - Seventh heaven
- 8 - Behind the eight-ball
- 10 - Cheaper by the dozen

Note: You must use a `switch` statement for implementation. The conditional operator `if` is FORBIDDEN for this task.

ucode connect

Sprint 02 | Half Marathon Full Stack > 11

SYNOPSIS

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <title>What kind of idiom?</title>
  <meta name="description" content="t03. What kind of idiom?">
</head>

<body>
  <h1>What kind of idiom?</h1>

  <script src="js/script.js"></script>
</body>

</html>
```

SEE ALSO

[JavaScript Switch Statement](#)
[Window prompt\(\) Method](#)
[JavaScript Number isFinite\(\) Method](#)



Sprint 02 | Half Marathon Full Stack > 12

Act Basic: Task 04

NAME

Numbers

DIRECTORY

t04_numbers/

SUBMIT

index.html, js/script.js

ALLOWED FUNCTIONS

prompt(), console.log(), String.*

DESCRIPTION

Create a JS file that will be included into the HTML page written in the **SYNOPSIS**.
The script must:

- call `prompt()` and take the numbers for the beginning and end of a range
- contain a function that:
 - takes two number variables (inclusive range), and prints suitable descriptions for all numbers in the range to the **Console** panel. Descriptions:
 - * 'number' is even
 - * 'number' is a multiple of 3
 - * 'number' is a multiple of 10
 - has default range of `1 - 100`

Look at the **EXAMPLE** of how the result may look like.

SYNOPSIS`checkDivision(beginRange, endRange)`

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <title>Numbers</title>
  <meta name="description" content="t04. Numbers">
</head>

<body>
  <h1>Numbers</h1>
```

ucode connect**Sprint 02 | Half Marathon Full Stack > 13**

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

</html>
```

EXAMPLE

```
1 -
2 is even
3 is a multiple of 3
4 is even
5 -
...
60 is even, a multiple of 3, a multiple of 10
```

ucode connect**Sprint 02 | Half Marathon Full Stack > 14**

Act Basic: Task 05

NAME

Total price

DIRECTORY

t05_total_price/

SUBMIT

index.html, js/script.js

ALLOWED FUNCTIONS

Number.*

DESCRIPTION

Create a JS file that will be included into the HTML page written in the **SYNOPSIS**. Imagine that you're shopping online, and every time you add something to your cart, this function is called. The script must contain a function that:

- takes three parameters:
 - Number of items
 - The price per item
 - The current total of the price
- returns the total order sum

Display and track the result in the **Console** panel.

You can test your function using the **test.js** file written in the **EXAMPLE**. It is appropriate to use **default parameter** in this task. Add more test cases of your own.

SYNOPSIS

```
total(addCount, addPrice, currentTotal) : number
```

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <title>Total price</title>
  <meta name="description" content="t05. Total price">
</head>

<body>
  <h1>Total price</h1>
```

ucode connect**Sprint 02 | Half Marathon Full Stack > 15**

```
<script src="js/script.js"></script>
<!-- <script src="js/test.js"></script> uncomment this line when testing -->
</body>

</html>
```

EXAMPLE

```
let sum = total(1, 0.1);
sum = total(1, 0.2, sum);
sum = total(1, 0.78, sum);
console.log(sum); // will return 1.08
```

SEE ALSO

[HTML DOM console.log\(\) Method](#)
[Number.prototype.toFixed\(\)](#)



Sprint 02 | Half Marathon Full Stack > 16

Act Basic: Task 06

NAME

Greeting

DIRECTORY

t06_greeting/

SUBMIT

index.html, js/script.js

ALLOWED FUNCTIONS

alert(), prompt(), console.log(), isNaN(), String.*

DESCRIPTION

Create a JS file that will be included into the HTML page written in the **SYNOPSIS**.
The script must:

- prompt the user to enter their first name and last name
- check if the input is valid
- capitalize the first letter of the first and last name if it is not
- use `alert()` and the `Console` panel to greet the user using their full name
- display `Wrong input!` both to the `Console` panel and using `alert()` if a line contains a digit or other incorrect input

SYNOPSIS

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <title>Greeting</title>
  <meta name="description" content="t06. Greeting">
</head>

<body>
  <h1>Greeting</h1>

  <script src="js/script.js"></script>
</body>

</html>
```

ucode connect

Sprint 02 | Half Marathon Full Stack > 17

