

INTRODUCTION (Exercise 1 of 20)

Here's the official solution in case you want to compare notes:

```
console.log('hello')
```

YOU DID IT!

Anything between the parentheses of `console.log()` are printed to the terminal.

So this:

```
console.log('hello')
```

prints hello to the terminal.

Currently we are printing a **string** of characters to the terminal: hello.

JAVASCRIPTING

VARIABLES (Exercise 2 of 20)

Here's the official solution in case you want to compare notes:

```
const example = 'some string'
console.log(example)
```

YOU CREATED A VARIABLE!

Nice work.

In the next challenge we will look at strings more closely.

Run `javascripting` in the console to choose the next challenge.

JAVASCRIPTING

STRINGS (Exercise 3 of 20)

Here's the official solution in case you want to compare notes:

```
console.log('this is a string')
```

SUCCESS.

You are getting used to this string stuff!

In the next challenges we will cover how to manipulate strings.

Run javascripting in the console to choose the next challenge.

You have 17 challenges left.

Type 'javascripting' to show the menu.

JAVASCRIPTING

STRING LENGTH (Exercise 4 of 20)

Here's the official solution in case you want to compare notes:

```
const example = 'example string'
console.log(example.length)
```

WIN: 14 CHARACTERS

You got it! The string example string has 14 characters.

Run javascripting in the console to choose the next challenge.

JAVASCRIPTING

REVISING STRINGS (Exercise 5 of 20)

Here's the official solution in case you want to compare notes:

Terminal

```
let pizza = 'pizza is alright'
pizza = pizza.replace('alright', 'wonderful')
console.log(pizza)
```

YES, PIZZA IS WONDERFUL.

Well done, with that `.replace()` method!

Next we will explore **numbers**.

Run javascripting in the console to choose the next challenge.

JAVASCRIPTING

NUMBERS (Exercise 6 of 20)

Here's the official solution in case you want to compare notes:

```
var example = 123456789
console.log(example)
```

YEAH! NUMBERS!

Cool, you successfully defined a variable as the number 123456789.

In the next challenge we will look at manipulating numbers.

Run javascripting in the console to choose the next challenge.

JAVASCRIPTING

ROUNDING NUMBERS (*Exercise 7 of 20*)

Here's the official solution in case you want to compare notes:

```
const roundUp = 1.5
const rounded = Math.round(roundUp)
console.log(rounded)
```

THAT NUMBER IS ROUNDED

Yep, you just rounded the number 1.5 to 2. Good job.

In the next challenge we will turn a number into a string.

Run javascripting in the console to choose the next challenge.

JAVASCRIPTING

NUMBER TO STRING (*Exercise 8 of 20*)

Here's the official solution in case you want to compare notes:

```
const n = 128
console.log(n.toString())
```

THAT NUMBER IS NOW A STRING!

Excellent. Good work converting that number into a string.

In the next challenge we will take a look at **if statements**.

Run javascripting in the console to choose the next challenge.

JAVASCRIPTING

IF STATEMENT (Exercise 9 of 20)

Here's the official solution in case you want to compare notes:

```
const fruit = 'orange'
if (fruit.length > 5) {
  console.log('The fruit name has more than five characters.')
} else {
  console.log('The fruit name has five characters or less.')
}
```

CONDITIONAL MASTER

You got it! The string orange has more than five characters.

Get ready to take on **for loops** next!

Run javascripting in the console to choose the next challenge.

JAVASCRIPTING

FOR LOOP (Exercise 10 of 20)

Here's the official solution in case you want to compare notes:

```
let total = 0
const limit = 10

for (let i = 0; i < limit; i++) {
  total += i
}

console.log(total)
```

THE TOTAL IS 45

That is a basic introduction to for loops, which are handy in a number of situations, particularly in combination with other data types like strings and arrays.

In the next challenge we'll start working with **arrays**.

Run javascripting in the console to choose the next challenge.

JAVASCRIPTING

ARRAYS (Exercise 11 of 20)

Here's the official solution in case you want to compare notes:

```
const pizzaToppings = ['tomato sauce', 'cheese', 'pepperoni']
console.log(pizzaToppings)
```

YAY, A PIZZA ARRAY!

You successfully created an array!

In the next challenge we will explore filtering arrays.

Run javascripting in the console to choose the next challenge.

JAVASCRIPTING

ARRAY FILTERING (Exercise 12 of 20)

Here's the official solution in case you want to compare notes:

```
const numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

const filtered = numbers.filter(function (number) {
  return (number % 2) === 0
})

console.log(filtered)
```

FILTERED!

Good job filtering that array.

In the next challenge we will work on an example of accessing array values.

Run javascripting in the console to choose the next challenge.

JAVASCRIPTING

ACCESSING ARRAY VALUES (Exercise 13 of 20)

Here's the official solution in case you want to compare notes:

```
const food = ['apple', 'pizza', 'pear']  
console.log(food[1])
```

SECOND ELEMENT OF ARRAY PRINTED!

Good job accessing that element of array.

In the next challenge we will work on an example of looping through arrays.

Run javascripting in the console to choose the next challenge.

JAVASCRIPTING

LOOPING THROUGH ARRAYS (Exercise 14 of 20)

Here's the official solution in case you want to compare notes:

```
const pets = ['cat', 'dog', 'rat']  
for (let i = 0; i < pets.length; i++) {  
  pets[i] = pets[i] + 's'  
}  
console.log(pets)
```

SUCCESS! LOTS OF PETS!

Now all the items in that pets array are plural!

In the next challenge we will move from arrays to working with **objects**.

Run javascripting in the console to choose the next challenge.

JAVASCRIPTING

OBJECTS (Exercise 15 of 20)

Here's the official solution in case you want to compare notes:

```
const pizza = {  
  toppings: ['cheese', 'sauce', 'pepperoni'],  
  crust: 'deep dish',  
  serves: 2  
}  
  
console.log(pizza)
```

PIZZA OBJECT IS A GO.

You successfully created an object!

In the next challenge we will focus on accessing object properties.

Run javascripting in the console to choose the next challenge.

JAVASCRIPTING

OBJECT PROPERTIES (Exercise 16 of 20)

Here's the official solution in case you want to compare notes:

```
const food = {  
  types: 'only pizza'  
}  
  
console.log(food.types)
```

CORRECT. PIZZA IS THE ONLY FOOD.

Good job accessing that property.

The next challenge is all about **object keys**.

Run javascripting in the console to choose the next challenge.

JAVASCRIPTING

OBJECT KEYS (*Exercise 17 of 20*)

Here's the official solution in case you want to compare notes:

```
const car = {  
  make: 'Toyota',  
  model: 'Camry',  
  year: 2020  
}  
const keys = Object.keys(car)  
  
console.log(keys)
```

CORRECT.

Good job using the `Object.keys()` prototype method. Remember to use it when you need to list the keys of an object.

The next challenge is all about **functions**.

Run javascripting in the console to choose the next challenge.

JAVASCRIPTING

FUNCTIONS (*Exercise 18 of 20*)

Here's the official solution in case you want to compare notes:

```
function eat (food) {  
  return food + ' tasted really good.'  
}  
  
console.log(eat('bananas'))
```

WOOO BANANAS

You did it! You created a function that takes input, processes that input, and provides output.

Run javascripting in the console to choose the next challenge.

JAVASCRIPTING

FUNCTION ARGUMENTS (*Exercise 19 of 20*)

Here's the official solution in case you want to compare notes:

```
function math (a, b, c) {  
  return (b * c) + a  
}  
  
console.log(math(53, 61, 67))
```

YOU'RE IN CONTROL OF YOUR ARGUMENTS!

Well done completing the exercise.

Run javascripting in the console to choose the next challenge.

SCOPE (*Exercise 20 of 20*)

Here's the official solution in case you want to compare notes:

```
/* eslint-disable no-unused-vars */  
  
const a = 1; const b = 2; const c = 3;  
  
(function firstFunction () {  
  const b = 5; const c = 6;  
  
  (function secondFunction () {  
    const b = 8  
    console.log('a: ' + a + ', b: ' + b + ', c: ' + c);  
    (function thirdFunction () {  
      const a = 7; const c = 9;  
  
      (function fourthFunction () {  
        const a = 1; const c = 8  
      })()  
    })()  
  })()  
})()
```

#EXCELLENT!

You got it! The second function has the scope we were looking for.

Now move on to a more challenging Javascript workshopper **Functional Javascript**:

```
npm install -g functional-javascript-workshop
```