أكاديمية البرمجة
Coding Academy
powered by Simplon

# JavaScript Exercises – Basics

## Data Types and Operators

Duration: 1.5 hours

1. Write the expressions in JavaScript syntax for the following accounting equations (analyze the operation from the input and output) – use google if necessary:

- Cash flow ratio
  **Sample Input:** cash: 1000, current liabilities: 500
  **Sample Output:** 2
- Net income
  **Sample Input:** revenues: 1000, expenses: 500
  **Sample Output:** 500
- Total assets
  **Sample Input:** liabilities: 1000, equity: 500
  **Sample Output:** 1500
- Net income (using profit margin and sales)
  **Sample Input:** profit: 1000, sales: 500
  **Sample Output:** 500000
- Average
  **Sample Input:** numbers: 7, 9, 2
  **Sample Output:** 6
- Discount
  **Sample Input:** price: 150, discount: 30%
  **Sample Output:** 105
- Age limit (older than 18 and less than 30)
  **Sample Input:** 20
  **Sample Output:** true
- Exponential
  **Sample Input:** number 1: 2, number 2: 3
  **Sample Output:** 8
- Remainder
  **Sample Input:** number 1: 10, number 2: 4
  **Sample Output:** 2

2. Find the output for these expressions and justify the output according to JavaScript interpretation:

- `typeof(100)`
- `typeof(73.9)`
- `typeof(NaN)`
- `typeof("Water")`
- `typeof(false)`
- `typeof(9 != 11)`
- `"Orang" + "e"`
- `"Orange" - "s"`
- `"4" + "8"`
- `"4" - "8"`
- `"name" + 3`
- `"name" - 3`
- `82 * "word"`
- `1 + "hello"`
- `"hello" + 1`
- `1 + true`
- `"hello" + true`
- `typeof (Infinity)`
- `1 == '1'`
- `1 === '1'`

---

## Strings

Duration: 15 minutes

1. Write the code to make this string “Welcome to Orange” outputted in the following forms using string methods:
   - **Output:** WELCOME TO ORANGE
   - **Output:** TO
   - **Output:** Hello from Orange
   - **Output:** welcome to orange
   - **Output:** 17
   - **Output:** Welcome to “Orange”
   - **Output:** Welcome to Orange Jordan
2. Write the code that replaces all occurrences of the first letter of a string with '*'

**Sample Input:** cactus
**Sample Output:** ca*tus

---

## Arrays

Duration: 15 minutes

1. Write the code to get this array outputted in the following forms:

   [“Coding”, “Academy”, “By”, “Orange”]

   - **Output:** [“Coding”, “Academy”, “By”, “Orange”, “Jordan”]
   - **Output:** [“Coding”, “Academy”]
   - **Output:** [“Welcome”, “To”, “Coding”, “Academy”, “By”, “Orange”]
   - **Output:** [“Academy”, “By”, “Orange”]
   - **Output:** “Coding Academy By Orange”
   - **Output:** [“Coding”, “Academy”, “By”, “Orange”]
   - **Output:** [“Coding”, “Orange”]

2. Using these two arrays write the JavaScript code that performs these operations on them:
   var fruit = ["banana", "apple", "orange", "watermelon"];
   var vegetables = ["carrot", "tomato", "pepper", "lettuce"];
   a. Remove the last item from the vegetable array.
   b. Remove the first item from the fruit array.
   c. Find the index of "orange."
   d. Add that number to the end of the fruit array.
   e. Find the length of the vegetable array.
   f. Add that number to the end of the vegetable array.
   g. Put the two arrays together into one array. Fruit first. Call the new Array "food".
   h. Remove 2 elements from your new array starting at index 4.
   i. Reverse your array.
   j. Turn the array into a string.

---

أكاديمية البرمجة
Coding Academy
powered by Simplon

# Functions

Duration: 1.5 hours

1. Write a function that makes each word in a string an element of an array.

    **Sample Input:** “Orange Jordan”
    **Sample Output:** [“Orange”, “Jordan”]

2. Write a function that converts the first seven digits of a mobile phone number to a hidden form.

    **Sample Input:** 0776807777
    **Sample Output:** *******777

3. Write a function that converts an email address to a hidden form.

    **Sample Input:** orange_academy@orange.jo
    **Sample Output:** orange…@orange.com

4. Write a function that turns the first letter of every word in a string to the upper case.

    **Sample Input:** “coding academy by orange”
    **Sample Output:** “Coding academy by orange”

5. Write a function that turns the first letter of every word in a string to the upper case.

    **Sample Input:** “coding academy by orange”
    **Sample Output:** “Coding Academy By Orange”

6. Write a function that flips a number.

    **Sample Input:** 92485
    **Sample Output:** 58429

7. Write a function that swaps the value of 2 variables in at least three ways.

    **Sample Input:** a=3, b=4
    **Sample Output:** a=4, b=3

8. Write a function that removes an indexed character from a string.

**Sample Input: (**“Orange”, 3)
**Sample Output:** Orage

9. Write a function that merges two strings after removing the first character of each one.

**Sample Input:** (“lora”, “inge”)
**Sample Output:** orange

10. Write a function that ensures a specific character appears in a string's first or last position.

**Sample Input:** (“o”, “orange”)
**Sample Output:** true

**Sample Input:** (“z”, “orange”)
**Sample Output:** false

11. Write a function that takes a string and returns every word in the string as an array element.

**Sample Input:** “Coding Academy by Orange”
**Sample Output:** [“Coding”, “Academy”, “By”, “Orange”]

12. Write a function that takes a string and reorders the characters alphabetically.

**Sample Input:** “Orange”
**Sample Output:** “aegnor”

---

# Conditionals

Duration: 1 hour

1. If someone is applying to the academy’s website, create the conditional to cover these situations:

- If the candidate is older than 30, print out: “You are not eligible. You may join other programs.”
- If the candidate is between 18 and 30, print out: “You are eligible. Start your application.”
- If the candidate is younger than 18, print out: “You may join the kids' program.”
- If the candidate is older than 60, print out: “You may join the seniors’ program.”
- Calculate the age using the year of birth dynamically.

2. Write a function that takes a string and switches the letters' case from upper to lower and vice versa.

**Sample Input:** “OrAnGe”
**Sample Output:** “oRaNgE”

3. Write a function that takes a string, capitalizes each word's first letter, and removes all spaces (Camel Case).

**Sample Input:** “Coding Academy by Orange”
**Sample Output:** “CodingAcademybyOrange”

4. Write a function that removes a specific element in an array.

**Sample Input: (**[“Coding”, “Academy”, “By”, “Orange”], “By”)
**Sample Output:** [“Coding”, “Academy”, “Orange”]

**5.** Write a function that checks if a number is odd or even.
6. Write a function that checks whether an input variable is a number.
7. Write a function that finds the largest of two numbers.
8. Write a function that checks if a triangle is equilateral, scalene, or isosceles.
9. Write a function that finds if a number is present in a given range.
10. Write a function that checks whether a year is a leap year.

---

# Loops

Duration: 2 hours

1. Write a script to print out all even numbers between 1 and 50, twice.
Use 1 for loop and 1 while loop.

2. Try solving the previous exercise using a single for loop.

3. Resolve Exercise 1 so that one loop prints out even numbers and the second prints out the odd ones.

4. Fizz-Buzz! Write a script that does the following:

- Counts from 1 to 100 and prints out something for each number:
  - If the number is divisible by 3, print "Fizz"
  - If the number is divisible by 5, print "Buzz"
  - If the number is divisible by both 3 and 5, print "FizzBuzz"
  - If the number does not meet any of the above conditions, just print the number

**Output:**

```
1, 2, Fizz, 4, Buzz, Fizz, 7, 8, Fizz, Buzz, 11, Fizz, 13, 14, Fizz Buzz, 16, 17, Fizz, 19, Buzz, Fizz, 22, 23, Fizz, Buzz, 26, Fizz, 28, 29, Fizz Buzz, 31, 32, Fizz, 34, Buzz, Fizz, ...
```

5. Rewrite the script with a function so that each iteration will trigger a function call.

**Sample Input:** fizzBuzz(1)
**Sample Output:** 1

**Sample Input:** fizzBuzz(15)
**Sample Output:** FizzBuzz

6. Create a recursive solution. A FizzBuzz function calls itself to continue the series.

7. Write a function that takes a number and converts it to certain banknotes.

**Sample Input:** (57, [25, 10, 5, 1])
**Sample Output:** 25, 25, 5, 1, 1

8. Write a function that counts the existence of a specific character within a string, despite the case.

**Sample Input:** (“Coding Academy by Orange”, “o”)
**Sample Output:** 2

**9.** Write for loops that accomplish the following tasks:

a. Print the numbers 0 - 20, one number per line.
b. Print only the ODD values from 3 - 29, one number per line.
c. Print the EVEN numbers 12 down to -14 in descending order, one number per line.
d. Print the numbers 50 down to 20 in descending order, but only if the numbers are multiples of 3.

10. Initialize two variables to hold the string 'CodingAcademy' and the array [7, 500, 'KH404', 'black', 36], then construct for loops to accomplish the following tasks:

e. Print each element of the array to a new line.
f. Print each string character in reverse order to a new line.

11. Construct a for loop that sorts the array [7, 23, 18, 9, -13, 38, -10, 12, 0, 124] into two new arrays:
Define an empty evens array to hold the even numbers and an odds array for the odd numbers.
In the loop, determine if each number is even or odd, then put that number into evens or odds, as appropriate.

12. Write the code to create meals using elements from these arrays, ensuring the rules below.

Protein options:
['chicken', 'pork', 'tofu', 'beef', 'fish', 'beans']
Grain options:
['rice', 'pasta', 'corn', 'potato', 'quinoa', 'crackers']
Vegetable options:

['peas', 'green beans', 'kale', 'edamame', 'broccoli', 'asparagus']
Beverage options:
['juice', 'milk', 'water', 'soy milk', 'soda', 'tea']
Dessert options
['apple', 'banana', 'more kale', 'ice cream', 'chocolate', 'kiwi']

- The number of meals is determined as an input.
- The meals must include one item from each category.
- No two meals should be identical.

---