# Lab: JavaScript Basics

You can check your solutions in <https://judge.softuni.org/Contests/5400/JavaScript-Basics-Lab>

## Multiply the Number by 2

Write a function that receives a **number** and **prints** as result that **number** **multiplied** by **two**.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 | 4 |
| 5 | 10 |
| 20 | 40 |

### Hints

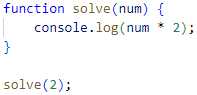
Create a function called solve (or some other name). As parameters, it will receive a number num.



Print the result inside the function.



If you want to test your code locally, you need to call the function.



## Student Information

You will be given **3 parameters** – student name (**string**), age (**number**) and average grade (**number**). Your task is to **print** all the info about the student in the following format:

"Name: {student name}, Age: {student age}, Grade: {student grade}"

**Note**: The grade should be formatted to the **second decimal** point.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 'John', 15, 5.54678 | Name: John, Age: 15, Grade: 5.55 |
| 'Steve', 16, 2.1426 | Name: Steve, Age: 16, Grade: 2.14 |
| 'Marry', 12, 6.00 | Name: Marry, Age: 12, Grade: 6.00 |

### Hint

Use toFixed() method to format the grade.

First, receive the input



Then, print the output



## Excellent Grade

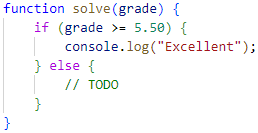
Write a function that receives a single **number** and checks if the grade is **excellent** or **not**.   
If it is, print "**Excellent**", otherwise print "**Not excellent**".

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5.50 | Excellent |
| 4.35 | Not excellent |

### Hints

Check if the number given is greater or equal to 5.50 and print the corresponding result.



## Month Printer

Write a program, that takes an **integer** as a parameter and **prints** the corresponding **month**. If the number **is more than 12** or **less than 1,** print "**Error!**"

### Input

You will receive a **single number**.

### Output

If the number is within the boundaries, **print** the corresponding month, otherwise print "**Error!**"

### Examples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| 2 | February |  | 13 | Error! |

## Math Operations

Write a JS function that takes **two** **numbers** and **a string** as input.

The **string** may be one of the following: '**+**', '**-**', '**\***', '**/**', '**%**', '**\*\***'.

**Print** on the console the result of the mathematical **operation** between **both numbers** and the **operator** you receive as a string.

The **input** comes as **two numbers** and **a string argument**, passed to your function.

The **output** should be printed on the console.

### Examples

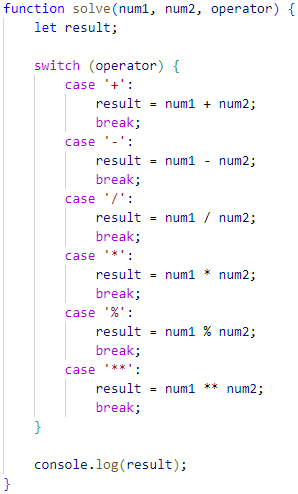
|  |  |
| --- | --- |
| **Input** | **Output** |
| 5, 6, '+' | 11 |
| 3, 5.5, '\*' | 16.5 |

### Hints

Write a function that receives **three** arguments.

Declare a variable named result that will keep your mathematical result.

Write down the switch command that will take the string from your input and depending on it, perform the mathematical logic between the two numbers.



Print the result on the console.



## Largest Number

Write a function that takes **three number arguments** as input and finds the **largest** of them.

**Print** the following text on the console: "**The largest number is {number}."**.

The **input** comes as **three number arguments** passed to your function.

The **output** should be printed to the console.

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5, -3, 16 | The largest number is 16. |
| -3, -5, -22.5 | The largest number is -3. |

### Hints

Write a function that receives three number arguments.

Declare a variable named result that will keep the result.

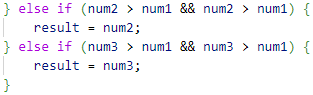


Make several checks to find out the largest of the three numbers.

Start with **num1**.



Do the same for the others.



Print the result on the console.



## Theatre Promotions

A theatre **is doing a ticket sale**, but they need a program **to** calculate the price of a **single** ticket. If the given age does **not** fit one of the categories**,** you should print "**Error!**". You can see the prices i**n** the table below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Day / Age** | **0 <= age <= 18** | **18 < age <= 64** | **64 < age <= 122** |
| **Weekday** | 12$ | 18$ | 12$ |
| **Weekend** | 15$ | 20$ | 15$ |
| **Holiday** | 5$ | 12$ | 10$ |

### Input

The input comes in **two parameters**. The **first** one will be the **type of day (string)**. The **second** is the **age** of the person (number).

### Output

**Print** the price of the ticket according to the table, or "**Error!**" if the age is not in the table.

### Constraints

* The age will be in the interval **[-1000…1000]**.
* The type of day will **always be** **valid**.

### Examples

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |  |
| 'Weekday', 42 | 18$ |  | 'Holiday', -12 | Error! | 'Holiday', 15 | 5$ |  |

## Circle Area

Write a function that takes **a single argument** as input. **Check the type** of input argument. If it is a **number**, assume it is the radius of a circle and **calculate the circle area**. Print the **area** **rounded** to **two decimal places**.

If the argument type is **NOT** a **number**, **print** the following text on the console:   
"We can not calculate the circle area, because we received a {type of argument}."

The **input** comes as a **single argument** passed to your function.

The **output** should be printed on the console.

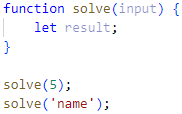
### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 | 78.54 |
| 'name' | We can not calculate the circle area, because we receive a string. |

### Hints

Write a function that receives a single argument.

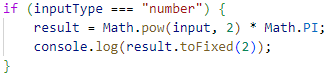
Declare a variable named result that will keep your result.



Check the type of the input argument with the typeof operator.



If the type is equal to number, calculate the circle area and print it on the console rounded to two decimal places. To do this, use the toFixed() method.  
The Math.pow() function returns the base to the exponent power, that is, the base exponent. You can find more information about the area [here](https://en.wikipedia.org/wiki/Circle):



If the type is **NOT** a number, print the following text on the console:

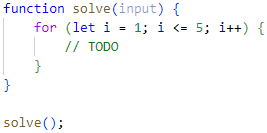


## Numbers from 1 to 5

Write a function that **prints** all the **numbers** from **1** **to** **5** (inclusive) each on a separate line.

### Hints

Create a **for** loop starting from **1** and continuing until **5** and print the number.



## Numbers from M to N

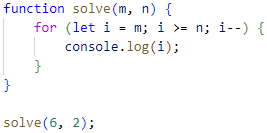
Write a function that receives a number **M** and a number **N** (M will always be bigger than N). **Print** all numbers from **M to N**.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 6, 2 | 6 5  4  3  2 |
| 4, 1 | 4  3  2  1 |

### Hints

Use **for** or **while** loop and print the numbers.



## Sum First and Last Array Elements

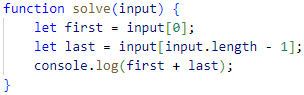
Write a function that receives an **array of numbers** and prints the sum of the **first** and **last** element in that array.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| [20, 30, 40] | 60 |
| [10, 17, 22, 33] | 43 |
| [11, 58, 69] | 80 |

### Hints

You can access the **last element** in an array by subtracting **1** from **its length**:



## Reverse an Array of Numbers

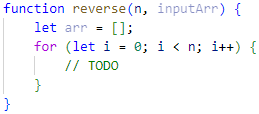
Write a program, which receives a number n and an **array** of elements. Your task is to **create** a new array with n numbers from the original array, **reverse** it and **print** its elements on a single line, space-separated.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3, [10, 20, 30, 40, 50] | 30 20 10 |
| 4, [-1, 20, 99, 5] | 5 99 20 -1 |
| 2, [66, 43, 75, 89, 47] | 43 66 |

### Hints

Use push() to add elements inside the new array.



Print the output.

