# Lab: Introduction to TypeScript

**Note: You should use types as much of your code as possible.**

## Calculate Rectangle Area

### Write a simple function that calculates rectangle area. You will be given two integers (whole) numbers, which will represent length and width of the rectangle. Calculate and print the area of the rectangle.

### Examples

|  |  |
| --- | --- |
| **input** | **output** |
| 5,  7 | 35 |

|  |  |
| --- | --- |
| **input** | **output** |
| 6,  8 | 48 |

## 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.** |

## Sum of Numbers N…M

Write a function that takes two integer numbers **N and M** as an input and **prints the sum** of all numbers from **N** to **M**.

* The **input** comes as **two string elements** that need to be **parsed** as numbers.
* The **output** should print the **sum**.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| **'1', '5'** | **15** |
| **'-8', '20'** | **174** |

## Day of Week

Write a function that prints a number between 1 and 7 when a **day of the week** is passed to it as a string and an **error message** if the string is **not recognized**.

* The **input** comes as a single-string argument.
* The **output** should be printed.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 'Monday' | 1 |
| 'Friday' | 5 |
| 'Invalid' | error |

## Math Operations

Write a function that takes **two** **numbers** and **a string** as an 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 |

## Even Position Element

Write a function that finds the elements at even positions in an array.

The **input** comes as an **array of string** elements.

The **output** is printed on the console. Collect all elements in a string, separated by space.

### Examples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| ['20', '30', '40', '50', '60'] | 20 40 60 | ['5', '10'] | 5 |

## Bigger Half

You are given an array of numbers. Write a JS function that **sorts** the array in **ascending order** and returns a new array, containing only the **second half** of the input. If there is an odd number of elements in the input, always take the bigger half. For example, if the input array contains 4 elements, the output should be 2, and if the input is 5 – the output is 3.

The **input** comes as an **array of number elements**.

The **output** is the **return** value of the function and should be an **array of numbers**.

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| [4, 7, 2, 5] | [5, 7] |
| [3, 19, 14, 7, 2, 19, 6] | [7, 14, 19, 19] |

## Cats

Write a function that receives **array** of strings in the following format **'{cat name} {age}'**.

Create a **Cat** **class** that receives in the **constructor** the **name** and the **age** parsed from the input.

It should also have a method named **"meow"** that will print **"{cat name}, age {age} says Meow"** on the console.

For each of the strings provided, you must **create a cat object** and invoke the **.meow ()** method**.**

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| **['Mellow 2', 'Tom 5']** | **Mellow, age 2 says Meow**  **Tom, age 5 says Meow** |
| **['Candy 1', 'Poppy 3', 'Nyx 2']** | **Candy, age 1 says Meow**  **Poppy, age 3 says Meow**  **Nyx, age 2 says Meow** |

## Employees

You're tasked to create a list of employees and their personal numbers.

You will receive an array of strings. Each string is an employee **name** and to assign them a personal number you have to find the **length of the name** (whitespace included).

At the end print all the list employees in the following format:

**"Name: {employeeName} -- Personal Number: {personalNum}"**

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| [  'Silas Butler',  'Adnaan Buckley',  'Juan Peterson',  'Brendan Villarreal'  ] | Name: Silas Butler -- Personal Number: 12  Name: Adnaan Buckley -- Personal Number: 14  Name: Juan Peterson -- Personal Number: 13  Name: Brendan Villarreal -- Personal Number: 18 |
| [  'Samuel Jackson',  'Will Smith',  'Bruce Willis',  'Tom Holland'  ] | Name: Samuel Jackson -- Personal Number: 14  Name: Will Smith -- Personal Number: 10  Name: Bruce Willis -- Personal Number: 12  Name: Tom Holland -- Personal Number: 11 |

## Aggregate Elements

Write a program that performs different operations on an array of elements. Implement the following operations:

* **Sum(ai)** - calculates the sum of all elements from the input array
* **Sum(1/ai)** - calculates the sum of the inverse values (1/ai) of all elements from the array
* **Concat(ai)** - concatenates the string representations of all elements from the array

The **input** comes as an array of number elements.

The **output** should be printed on the console on a new line for each of the operations.

### Examples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| [1, 2, 3] | 6  1.8333333333333333  123 | [2, 4, 8, 16] | 30  0.9375  24816 |