Functions -Exercises

[1. Smallest of Three Numbers 1](#_Toc145489475)

[2. Add and Subtract 2](#_Toc145489476)

[3. Characters in Range 2](#_Toc145489477)

[4. Odd and Even Sum 3](#_Toc145489478)

[5. Palindrome Numbers 3](#_Toc145489479)

[6. Password Validator 3](#_Toc145489480)

[7. Shortest and Longest Word 4](#_Toc145489481)

[8. Perfect Number 4](#_Toc145489482)

[9. Progress Bar 5](#_Toc145489483)

[10. Factorial Division 5](#_Toc145489484)

# Smallest of Three Numbers

Write a JS function which receive **three integer** numbers to print the **smallest** of the three integer numbers. Use appropriate name for the function.

**Examples**

|  |  |
| --- | --- |
| Input | Output |
| 2,  5,  3 | 2 |
| 600,  342,  123 | 123 |
| 25,  21,  4 | 4 |

# Add and Subtract

You will receive 3 **integers.** Write a JS function **sum** to get the sum of the first two integers and **subtract** function that subtracts the third integer from the result from the Sum function.

**Examples**

|  |  |
| --- | --- |
| Input | Output |
| 23,  6,  10 | 19 |
| 1,  17,  30 | -12 |
| 42,  58,  100 | 0 |

# Characters in Range

Write a JS function that receives **two characters** and prints on a single line all the characters in between them according to the **ASCII** code.

**Examples**

|  |  |
| --- | --- |
| Input | Output |
| 'a',  'd' | b c |
| '#',  ':' | $ % & ' ( ) \* + , - . / 0 1 2 3 4 5 6 7 8 9 |
| 'C',  '#' | $ % & ' ( ) \* + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @ A B |

# Odd and Even Sum

You will receive a **single number.**You have to write a JS function, that returns the **sum** of **all even** and **all odds** digits from that number.

**Examples**

|  |  |
| --- | --- |
| Input | Output |
| 1000435 | Odd sum = 9, Even sum = 4 |
| 3495892137259234 | Odd sum = 54, Even sum = 22 |

# Palindrome Numbers

A palindrome is a number which reads the same **backward as forward**, such as 323 or 1001. Write a JS function which receives an **array of positive integer** and checks if each integer is a palindrome or not.

Write isPalindrome function.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Input | Output |  | Input | Output |
| [123,323,421,121] | false  true  false  true |  | [32,2,232,1010] | false  true  true  false |

# Password Validator

Write a JS function that checks if a given password is valid. Password rules are:

* **6 – 10 characters (inclusive)**
* **Consists only of letters and digits**
* **Have at least 2 digits**

If a password is valid print "Password is valid". If it is not valid, for every unfulfilled rule print a message:

* **"Password must be between 6 and 10 characters"**
* **"Password must consist only of letters and digits"**
* **"Password must have at least 2 digits"**

Write a function for each rule.

|  |  |
| --- | --- |
| Input | Output |
| 'pass' | Password must be between 6 and 10 characters  Password must have at least 2 digits |
| 'APass123' | Password is valid |
| 'Pa$s$s' | Password must consist only of letters and digits  Password must have at least 2 digits |

# Shortest and Longest Word

You will receive a **single string.** This string will be a sentence. Your task here is to create JS function to find:

The **longest** and the **shortest** word in that sentence. If two words have **equal length** take the first occurrence.

**Examples**

|  |  |
| --- | --- |
| Input | Output |
| 'Hello how are you today? I hope you are fine ' | Longest -> Hello Shortest -> I |
| 'Lorem Ipsum is dummy text of the typesetting industry.' | Longest -> typesetting Shortest -> is |

# Perfect Number

Write a JS function that receive a **number** and return if this number is perfect or not.

A perfect number is a positive integer that is equal to the **sum of its proper positive divisors**. That is the sum of its positive divisors excluding the number itself (also known as its **aliquot sum**).

Equivalently, a perfect number is a number that is **half the sum** of all of its positive divisors (including itself) => 6 is a perfect number, because it is the sum of 1 + 2 + 3 (all of which are divided without remainder).

**Examples**

|  |  |
| --- | --- |
| Input | Output |
| 6 | Perfect number!  1 + 2 + 3 |
| 28 | Perfect number!  1 + 2 + 4 + 7 + 14 |
| 1236498 | It’s not so perfect. |

# Progress Bar

You will receive a **single number** between 0 and 100 which is divided with 10 without residue (0, 10, 20, 30...).

Your task is to create a JS function that visualize a **loading bar** depending on that number you have received in the input. See examples for more clarity.

**Examples**

|  |  |
| --- | --- |
| Input | Output |
| 30 | 30% [%%%.......]  Loading... |
| 50 | 50% [%%%%%.....]  Loading... |
| 100 | [%%%%%%%%%%]  Complete! |

# Factorial Division

Write a JS Function that reaceives **two** integer numbers. Calculate [factorial](https://en.wikipedia.org/wiki/Factorial) of each number. Divide the first result by the second and print the division formatted to the **second decimal** point.

**Examples**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Input | Output |  | Input | Output |
| 5  2 | 60.00 |  | 6  2 | 360.00 |

**Hints**

Try to use [recursion](https://en.wikipedia.org/wiki/Recursion_(computer_science))