

### Instrucciones:

Como requisito realizar este laboratorio será necesario que tengas en tu computadora instalado lo siguiente:

- Un editor de texto
- Un navegador web o Node.js instalado

Para esta ocasión pondrás a prueba tus conocimientos adquiridos de JavaScript hasta el momento.

¡Mucha suerte!, cualquier duda que tengas al respecto me puedes enviar un mensaje por Slack.

### I. ¿Es una cadena vacía? (Dificultad: ☆)

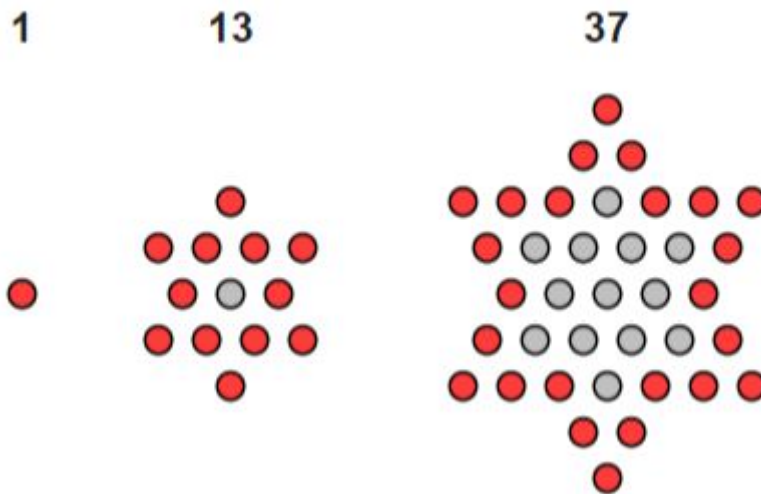
Escribe una función que indique con un valor booleano si dada una cadena de texto está vacía.

```
function isEmpty(text) {
  // ...
}

isEmpty("hola mundo"); // false
isEmpty(" "); // false
isEmpty(""); // true
```

### II. El n-ésimo número estrella (Dificultad: ☆)

Un número estrella se refiere a la cantidad de puntos que se necesitan para rodear una estrella de 6 picos con N niveles. Por ejemplo los primeros 3 números estrella son **1**, **13** y **37**:



Escribe una función que calcule el n-ésimo número estrella.

```
function starNumber(level) {
  // ...
}

starNumber(1); // 1
starNumber(2); // 13
starNumber(3); // 37
```

Hint: Busca en Google!!!

### III. Mínimo y máximo de un arreglo (Dificultad: ☆☆)

Escribe una función que dado un arreglo de números de longitud N indique con un objeto cuál es el número más pequeño y el más grande dentro del arreglo.

```
function minMax(array) {
  // ...
}

minMax([1,2,3]); // { min: 1, max: 3 }
minMax([5]); // { min: 5, max: 5 }
minMax([]); // { min: NaN, max: NaN }
```

#### IV. Combinación más grande v2 (Dificultad: ☆☆)

Escribe una función que dado un número de dos dígitos (entre 10 y 99) indique si dicho número forma la combinación más grande entre sus dos dígitos.

Por ejemplo si el número es **43** la función debería regresar **verdadero** dado que es el número más grande que se forma con el **3** y **4**; si el número es **27** la función debería regresar **falso** dado que **72** es la combinación más grande que se puede formar con el **2** y **7**.

```
function largestSwap(number) {
  // ...
}

largestSwap(10); // true
largestSwap(34); // false
largestSwap(43); // true
largestSwap(27); // false
largestSwap(11); // true
```

**Hint:** toString + charAt + parseInt = WIN

#### V. ¿Cuáles vocales tiene el texto? (Dificultad: ☆☆☆)

Escribe una función que dado un texto indique cuáles son las vocales que contiene (sin repetir y no es necesario que estén ordenadas).

```
function getVowels(text) {
  // ...
}

getVowels("Hola mundo"); // ["a", "o", "u"]
getVowels(""); // []
getVowels("aaaaaaaaah!!!"); // ["a"]
getVowels("Murcielago"); // ["a", "e", "i", "o", "u"]
```

**Hint:** Puedes ver a las cadenas como un arreglo de caracteres.

## VI. ¿Es un subconjunto? (Dificultad: ☆☆☆)

Un **conjunto** es una colección de elementos que no se repiten y que no tienen un orden en particular.

Un **subconjunto** es un conjunto que contiene elementos del conjunto original y todos los elementos del subconjunto DEBEN PERTENECER al conjunto original.

El **conjunto vacío** es subconjunto de cualquier conjunto.

Escribe una función que dado dos arreglos indique si uno es subconjunto del otro.

```
function isSubset(set, subset) {
  // ...
}

isSubset([1,2,3,4,5], [3,4]); // true
isSubset([5,4,3,2,1], []); // true
isSubset([8,4,3,7,10], [10]); // true
isSubset([1,2,3,4,5], [1,6]); // false
isSubset([6,4,7,3], [8,9]); // false
```