

---

## EXAMPLE 4 - JAVASCRIPT

---

This example has to be tested in the Automated Test Environment introduced in class. Check the ATE-settings.pdf document for more information.

Fill the **practica1.js** document with the code of the functions corresponding to each exercise (*exercise1* in *misterio1*, and so on)

```
function misterio1(edad){  
  return true;  
}
```

**The results have to be correct (check spaces or capital letters) as they will be tested in an automatic process. It is highly recommended to study the tests to better understand the exercises.**

### Exercises:

1. The function receives one parameter, and it has to verify that it is a number and that it is between 18 and 120. If the "age" is between 18 and 120 it will return the string of characters "Puedes conducir" Otherwise it will return: " Introduce un numero válido"
2. The function receives one parameter, that should be a number from 1 to 12 (or a string-number). It will return the name of the month: "Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio", "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre". If the parameter is not correct, it will return " No es un mes"
3. The function receives three parameters: Operation, number1 and number2. Operation must be "suma","resta","multiplicación" or "división. Numbers must be numbers. If they are not numbers, it will return: " Los valores no son números" If the operation is not one of those mentioned, it will return: "Introduce operación válida". Otherwise, it will return the result of the operation. In case of division, if the second number is 0, it will return: " No dividir por 0"
4. The function receives two parameters: a DNI number and a letter. DNI will be a number between 0 and 999999999, or it will return: " El número no es válido ". If it is a valid number, and the letter is module 23 of the number in the array of letters shown below, it will return: "El

DNI es correcto". Otherwise, it will return: "La letra o el número proporcionados no son correctos"

```
var letras = ['T', 'R', 'W', 'A', 'G', 'M', 'Y', 'F', 'P', 'D', 'X', 'B', 'N', 'J', 'Z', 'S',  
'Q', 'V', 'H', 'L', 'C', 'K', 'E', 'T'];  
letra = letra.toUpperCase();
```

5. The function receives one parameter. If it is not an array it will return "No es posible". If it is an array it will return the third value of the array. If there is no third value, it will return "No es posible"

6. The function receives a parameter. It will check if it is A,B,C,D or E (Change it first to uppercase):

If it is A, return "Nota entre 8 y 10".

If it is B, return "Nota entre 6.5 y 8".

If it is C, return "Nota entre 5 y 6.5".

If it is D, return "Nota entre 4 y 5".

If it is E, return "Nota entre 0 y 4".

If it is not one of those letters, return "Introduce una nota válida".

7. The function receives one parameter. If it is not a positive integer (or string-number), it will return: "Introduce un número positivo". Otherwise, return how many times it must be added to reach or exceed 100

8. The function receives one parameter. It will check that it is a number from 1 to 9 (or string-number). If not, it will return: "Introduce un número de 1 a 9". Otherwise, it will return the multiplication table of the number in the following format:

Table of 1:

1x1=1,2x1=2,3x1=3,4x1=4,5x1=5,6x1=6,7x1=7,8x1=8,9x1=9.

Table of 6:

1x6=6,2x6=12,3x6=18,4x6=24,5x6=30,6x6=36,7x6=42,8x6=48,9x6=54.

9. The function receives two parameters: Rows and columns. The values have to be numbers under 10. (String-numbers are not valid). It will generate a string of an HTML table with the rows and the columns specified. If the requirements are not met, it will return: "Introduce dos números positivos menores que 10". If they are met, it will return the table (without spaces). Examples:

Table 1x1:

```
<table><tr><td></td></tr></table>
```

Table 2x3:

```
<table><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>
```

10. The function gets the current date and returns the day and month as a string in the following format: "Hoy es el día 3 del mes 3". Use `.getDay()` and `.getMonth()` Date methods.
11. The function receives one parameter. Check if it is a string. If it is not, it will return: "Mete un string". If it is a string, it will return:
  - If the length is greater than 10: *substring: and characters 2,3,4*  
For example: "extraordinaire morning" returns "substring:xttr"
  - If its length is less or equal to 10: *length: and its length*  
Example: "hello" returns "length:5"
12. The function receives one parameter. Check if it is a string. If it is not, it will return: "Mete un string". If it is a string, it will return the inverted string. Example: "hola" will return "aloh"
13. The function receives a parameter. Check if it is a string. If it is not, it will return: "Mete un string". If it is a string, it will return the string in uppercase. Example: "hola" will return "HOLA"
14. The function receives a parameter. Check if it is a string. If it is not, it will return: "Mete un string". If it is a string, it will return *true* if it contains "DAM", and *false* if it does not. "DAM" has to be in upper case. Examples:  
"I study dam" → false  
"DAM Course" → true
15. The function receives three parameters. Check if they are strings. If any is not, it will return: "Alguno no es string"; If they are all strings, it should return a *stringify* of a generated json:

```
{firstname:parameter1,lastname1: parameter2,lastname2: parameter3}
```

16. The function receives one parameter. Check if it is a string. If it is not, it will return: "Mete un string". If it is a string, it will return the same string but changing the word "suspender" by "aprobar".

Example: "Voy a suspender" → "Voy a aprobar"

17. The function receives one parameter, which should be a numeric array (no need to check). It returns a string containing an unordered html list with the elements of the array (no spaces)
18. The function receives one parameter, which should be a numeric array (no need to check). It returns a string containing an ordered html list with the elements of the array in reverse order. (no spaces)
19. The function receives one parameter, which should be a JSON (no need to check). The JSON will have the keys *nombre*, *apellido1* and *apellido2*, but they can be in any order. Return the values as a string: "*nombre apellido1 apellido2*." (With spaces and a final dot.)
20. The function receives two parameters. The first one will always be an array and the second a number (no need to check). It should return a *stringify* of a JSON generated from position:

`{resultado:array[position]}`

If the value does not exist, or it is not an array or it is not a number, it will return "No es posible"