

---

---

# JavaScript

## 1. Introduction and language overview

### Practice

---

---

Alan Rodas Bonjour

Part 1: This part can be answered in paper, as they correspond to theory concepts.

Parte 1: Esta parte puede ser contestada en papel, ya que corresponden a conceptos teóricos.

**1) Which ones of the following are valid JS types**

¿Cuáles de los siguientes son tipos válidos en JS?

- |           |              |
|-----------|--------------|
| a) number | g) undefined |
| b) null   | h) string    |
| c) int    | i) text      |
| d) array  | j) str       |
| e) object | k) function  |
| f) float  | l) char      |

**2) What is the correct syntax for a function definition in JS?**

¿Cuál es la sintaxis correcta para la definición de una función en JS?

- |                     |                                |
|---------------------|--------------------------------|
| a) func f() { ... } |                                |
| b) def f(): ...     | c) function f() { ... }        |
|                     | d) number function f() { ... } |

**3) Which of the following are valid repetition forms in JS**

¿Cuáles de los siguientes son formas de repetición válidas en JS?

- |  |                                   |
|--|-----------------------------------|
| a) for (var i = 0; i < 5; i++) { ... } | g) for (elem of object) { ... }   |
| b) repeat (5) { ... }                  | h) foreach i in [1, 2, 3] { ... } |
| c) foreach elem in array { ... }       | i) while ( condition ) { ... }    |
| d) for (elem in array) { ... }         | j) do ( condition ) while { ... } |
| e) for (elem of array) { ... }         | k) while ( condition ) do { ... } |
| f) for (key in object) { ... }         | l) do { ... } while ( condition ) |

### 4) Which of the following are Truthy values?

¿Cuales de los siguientes son valores “verdaderos”?

- |            |              |
|------------|--------------|
| a) true    | f) 0         |
| b) false   | g) undefined |
| c) “hello” | h) null      |
| d) 7       | i) ““        |
| e) -5      | j) “ “       |

Part 2: This part can be performed in a console, such as <https://jsconsole.com>

Parte 2: Esta parte puede realizada en una consola, como en <https://jsconsole.com>

### 5) Create a variable by the name “a” and assign it the value 5. Then display the variable in the console.

Cree una variable con el nombre “a” y asígnele el valor 5. Luego muestre la variable en la consola.

### 6) Create other two variables, “b” and “c”, and assign them the values 6 and 7.

Cree otras dos variables, “b” y “c”, y asignarles los valores 6 y 7.

### 7) Consider the three variables as the sides of a triangle. Write an expression that calculates the perimeter of the triangle.

Considere las tres variables como los lados de un triángulo. Escriba una expresión que calcule el perímetro del triángulo.

### 8) Write an expression that checks if the variable “b” is between 10 and 30. Change the value of the variable and try the expression in different scenarios.

Defina una expresión que indique si la variable “b” está entre 10 y 30. Cambie el valor de la variable e intente probar la expresión en diferentes escenarios.

### 9) Write a function **toCelsius(degrees)** that takes the temperature as the number of degrees in Fahrenheit and returns the same temperature in Celsius. To convert Farenheit to celsius you have to subtract 32 and then multiply by 5/9.

Defina una función **toCelsius(degrees)** que toma la temperatura como el número de grados en Fahrenheit y describe la misma temperatura en Celsius. Para convertir los Fahrenheit a Celsius se debe sustraer 32 y luego multiplicar por 5/9.

### 10) Create a function **isLeap(year)** that receives a number that represents a year, such as 2013. Then, write an expression that states if the year stored in the variable is a leap year or not. A year is a leap if it's divisible by 4 or by 100, but not by 400.

Cree una función **isLeap(year)** que recibe un número que representa un año, como 2013.

Luego escriba una expresión que indique si el año almacenado en la variable contiene un año bisiesto o no. Un año es bisiesto cuando es divisible por 4 o por 100, pero no por 400.

**11)** Create the function **abs(n)** that takes a number and returns the absolute value of the number.

Cree una función **abs(n)** que toma un número y describe el valor absoluto de dicho número.

**12)** Create a function **largest(a, b, c)** that takes three numbers and returns the largest number between the three.

Cree una función **largest(a, b, c)** que toma tres números y describe el valor más alto entre los tres dados.

**13)** Modify the previous function so that the third argument is optional, and it's only compared if it's not undefined.

Modifique la función previa de forma tal que el tercer argumento sea opcional, y que solo se compare si no es undefined.

**14)** Create a function **pathJoin(base, folder, os)** that takes three strings and returns a path that has a valid format according to the "os". If the "os" is "windows", then, the base and the folder should be joined using "\". If it's any other than "windows", it should be joined through "/". e.g. `pathJoin("Users", "John", "window")` should return "Users\John", and `pathJoin("Users", "John", "linux")` should return "Users/John".

Cree una función **pathJoin(base, folder, os)** que toma tres strings y retorna un path que tiene un formato válido acorde a "os". Si "os" es "windows" entonces base y folder deben unirse mediante "\". Si es otro distinto a windows, se deben unir mediante "/". e.g. `pathJoin("Users", "John", "window")` debe describir "Users\John", y `pathJoin("Users", "John", "linux")` debe describir "Users/John".

**15)** The strings can be treated as objects, and can respond to the "length" message, returning the length of the string. e.g. "HELLO".length returns 5. Also, they respond to the message `charAt(index)` that returns the character at the given index, where indices start at 0. e.g. "HELLO".charAt(0) returns "H", and "HELLO".charAt(2) returns "L". Try to use this expressions in the console.

Los strings pueden ser tratados como objetos, y responden al mensaje "length", describiendo la longitud del string. Por ej. "HOLA".length describe 4. Además responde al mensaje `charAt(index)` que describe el caracter en el índice dado, donde los índices empiezan en 0. Por ej. "HOLA".charAt(0) describe "H", y "HOLA".charAt(2) describe "L". Pruebe utilizar estas expresiones en la consola.

**16)** Using the previously mentioned expressions for strings, write a procedure **printLetters(word)** that takes a string and prints to the console each letter of the word at a time.

Usando las expresiones previamente mencionadas para strings, escriba un procedimiento **printLetters(word)** que toma un string e imprime en la consola cada letra de la palabra, una letra a la vez.

**17)** Using the previously mentioned expressions for strings, write a procedure **printLetters(word)** that takes a string and prints to the console each letter of the word at a time.

Usando las expresiones previamente mencionadas para strings, escriba un procedimiento **printLetters(word)** que toma un string e imprime en la consola cada letra de la palabra, una a la vez.

**18)** Try in the console the expresion: "1" == 1, and the expression "1" === 1. Analyze the results and think about the difference between == and ===.

Pruebe en la consola la expresión "1" == 1, y la expresión "1" === 1. Analice los resultados y piense sobre las diferencias entre == e ===.

**19)** Write a function **arrayFromTo(start, end)** that takes two numbers where the second is greater than the first, and returns an array that contains each number from the "start" to the "end" including both. e.g. **arrayFromTo(3, 7)** returns [3,4,5,6,7].

Escriba una función **arrayFromTo(start, end)** que toma dos numeros, donde el segundo es más grande que el primero, y describe un array que contiene los números entre "start" y "end", inclusive. e.g. **arrayFromTo(3, 7)** describe [3,4,5,6,7].

**20)** Write a function **arrayFromToStepping(start, end, step)** that takes three numbers where the second is greater than the first, and returns an array that contains numbers from the "start" to the "end" including both by steps of the given "steps". e.g. **arrayFromToStepping(3, 13, 3)** returns [3,6,9,12].

Escriba una función **arrayFromToStepping(start, end)** que toma tres numeros, donde el segundo es más grande que el primero, y describe un array que contiene los números entre "start" y "end", inclusive. separados por pasos de a "step" e.g. **arrayFromToStepping(3, 12, 3)** describe [3,6,9,12].

**21)** Write a function **maxIn(array)** that takes a non empty numeric array and returns the largest element in the array.

Escriba una función **maxIn(array)** que toma un array numérico no vacío y describe el elemento más grande del array.

**22)** Arrays are also objects and respond to the "append(element)" message by adding the given element to the end of the array. e.g. given `var arr = [1,2,3]`, calling `arr.append` makes the array become [1,2,3,4]. Try this in the console.

Los arrays son objetos que responden al mensaje "append(element)" agregando un elemento al final del array. Ej. dado `var arr = [1,2,3]`, llamar a `arr.append(4)` hace que el array se modifique quedando como [1,2,3,4]. Pruebe esto en la consola.

**23)** Using the previous knowledge, write the function **squared(array)**, that takes a numeric array and returns a new array that contains each element of the input squared. e.g. **squared([2,3,4])** returns [4,9,16].

Usando el conocimiento previo, escriba una función **squared(array)**, que toma un array

numerico y describe un nuevo array que contiene los mismos elementos que la entrada, elevados al cuadrado. e.g. `squared([2,3,4])` describe `[4,9,16]`.

**24)** Write the function **`evens(array)`**, that takes a numeric array and returns a new array that contains only the element of the input that are even. e.g. `even([2,3,4,5,6,7])` returns `[2,4,6]`.

Escriba una función **`evens(array)`**, que toma un array numérico y describe un nuevo array que contiene sólo los elementos de la entrada que son pares. e.g. `evens([2,3,4,5,6,7])` describe `[2,4,6]`.

**25)** Write the function **`reversed(array)`**, that takes an array and returns a new array that contains the element of the input in reversed order. e.g. `reversed([1,2,3,4])` returns `[4,3,2,1]`.

Escriba una función **`reversed(array)`**, que toma un array numérico y describe un nuevo array que contiene todos los elementos de la entrada en orden inverso. e.g. `reversed([1,2,3,4])` describe `[4,3,2,1]`.

**26)** Write the procedure **`ladderPattern(height)`**, that takes a number and prints the following pattern as high as the given number.

Escriba el procedimiento **`ladderPattern(height)`**, que toma un número e imprime el siguiente patrón, tan alto como el número.

```
X
X X
X X X
X X X X
X X X X X
```

e.g. `height = 5`

**27)** Write the procedure **`invertedLadderPattern(height)`**, that takes a number and prints the following pattern as high as the given number.

Escriba el procedimiento **`invertedLadderPattern(height)`**, que toma un número e imprime el siguiente patrón, tan alto como el número.

```
X X X X X
X X X X
X X X
X X
X
```

e.g. `height = 5`

**28)** Write the procedure **`pyramidPattern(height)`**, that takes a number and prints the following pattern as high as the given number.

Escriba el procedimiento **pyramidPattern(height)**, que toma un número e imprime el siguiente patrón, tan alto como el número.

```
      X
    X X X
  X X X X X
X X X X X X X
X X X X X X X X X
```

e.g. `height = 5`

**29)** Write an object that represents a person, with name, surname, age and address. Save the object in a variable and perform different operations to the object, such as accessing the name, surname, and changing the address. Try this in the console.

Escriba un objeto que representa una persona, con nombre, apellido, edad y dirección. Guarde el objeto en una variable y realice distintas operaciones sobre el objeto, como acceder al nombre, al apellido y cambiar la dirección. Pruebe esto en la consola.

**30)** Write a function **initials(person)** that takes a person and returns the initials of the person. e.g. `initials({ name: "John", surname: "Doe", age: 35, address: "Evergreen Ave 743" })` returns "J.D."

Escriba una función `initials(person)` que toma una persona y describe las iniciales de la persona. e.g. `initials({ name: "John", surname: "Doe", age: 35, address: "Evergreen Ave 743" })` describe "J.D."

**31)** Write a procedure **printAttributes(object)** that takes an object and prints every attribute that the object has.

Escriba una función `printAttributes(objeto)` que toma un objeto e imprime cada atributo que tiene el objeto.

**32)** Write a function **oldest(people)** that takes an array of person, and returns the person that is the oldest of them all.

Escriba una función `oldest(people)` que toma un array de personas y que describe la persona que es más vieja entre todas.

**33)** Write a function **names(people)** that takes an array of person, and returns an array of names with no duplicates.

Escriba una función `names(people)` que toma un array de personas y describe un array con nombres de las personas, sin duplicados.

**34)** Write a function **neighbours(people)** that takes an array of person, and returns an array of array of person, where each internal array contains people that are neighbours.

Escriba una función `neighbours(people)` que toma un array de personas y describe un array

de array de personas, en donde cada array interno contiene personas que son vecinas entre sí.

**35)** Write a function **moveWith(people, name1, surname1, name2, surname2)** that takes an array of person, and four strings, where the first two represent the name and surname of a person to move (change address) and the last two the name and surname of a person to move the first one with, and returns a new array of person, with all persons in the input, but where the address of the first person has changed accordingly.

Escriba una función que toma un array de personas y cuatro strings, donde los primeros dos representan el nombre y el apellido de una persona a mudar (cambiar de dirección) y los últimos dos el nombre y apellido de la persona con la cual debe mudarse, y describe un nuevo array, con todas las personas de la entrada, pero donde la dirección de la persona fue cambiada de manera acorde.