SOC Servei d'Ocupació de Catalunya
Generalitat de Catalunya
MINISTERIO DE TRABAJO Y ECONOMÍA SOCIAL
SERVICIO PÚBLICO DE EMPLEO ESTATAL SEPE
Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

CIEF

MÓDULO 1. MF0951_2
INTEGRAR COMPONENTES SOFTWARE EN PÁGINAS WEB

UNIDAD FORMATIVA 1.
UF1305 INTEGRAR COMPONENTES SOFTWARE EN PÁGINAS WEB.

ÍNDEX

4.2 **Funciones y metodos**

_Declaración de funciones

_Ambito de variables Locales vs Externas

_Parámetros

_Valores predeterminados

_Devolviendo un valor

_Parámetros rest y spread

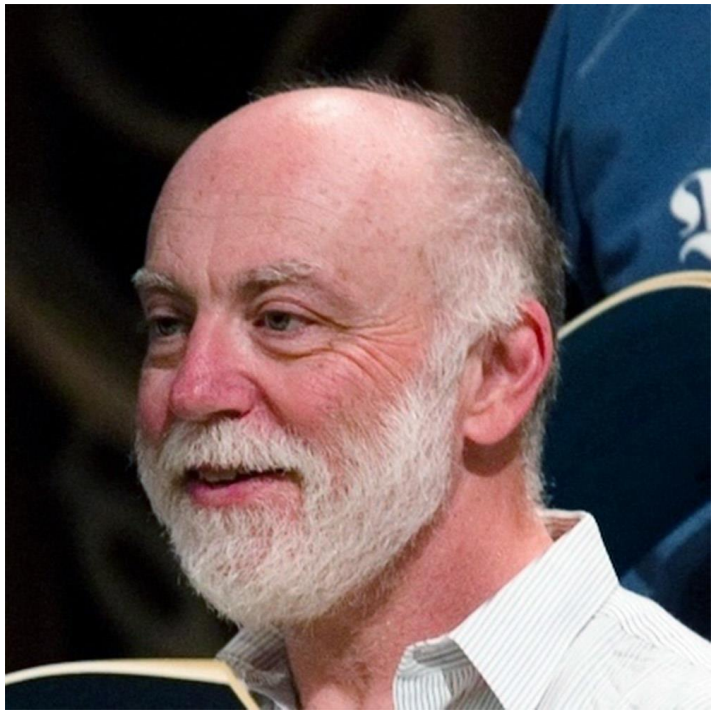
_Nomenclatura de funciones

_Funciones == Comentarios

SOC Servei d'Ocupació de Catalunya
Generalitat de Catalunya
MINISTERIO DE TRABAJO Y ECONOMÍA SOCIAL
SERVICIO PÚBLICO DE EMPLEO ESTATAL SEPE
Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

"Iterar es humano, 'recursivar' es divino“

-- L. Peter Deutsch



L. Peter Deutsch es el fundador de Aladdin Enterprises y el creador de Ghostscript, un intérprete libre de PostScript. También es el autor de numerosos RFCs.

SOC Servei d'Ocupació de Catalunya
Generalitat de Catalunya
Ministerio de Trabajo y Economía Social
Servicio Público de Empleo Estatal SEPE
Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

CIEF

4.

Javascript. II

SOC Servei d'Ocupació de Catalunya

Generalitat de Catalunya

MINISTERIO DE TRABAJO Y ECONOMÍA SOCIAL

SERVICIO PÚBLICO DE EMPLEO ESTATAL SEPE

Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

CIEF

4.2.

Funciones

SOC Servei d'Ocupació de Catalunya

Generalitat de Catalunya

MINISTERIO DE TRABAJO Y ECONOMÍA SOCIAL

SERVICIO PÚBLICO DE EMPLEO ESTATAL SEPE

Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

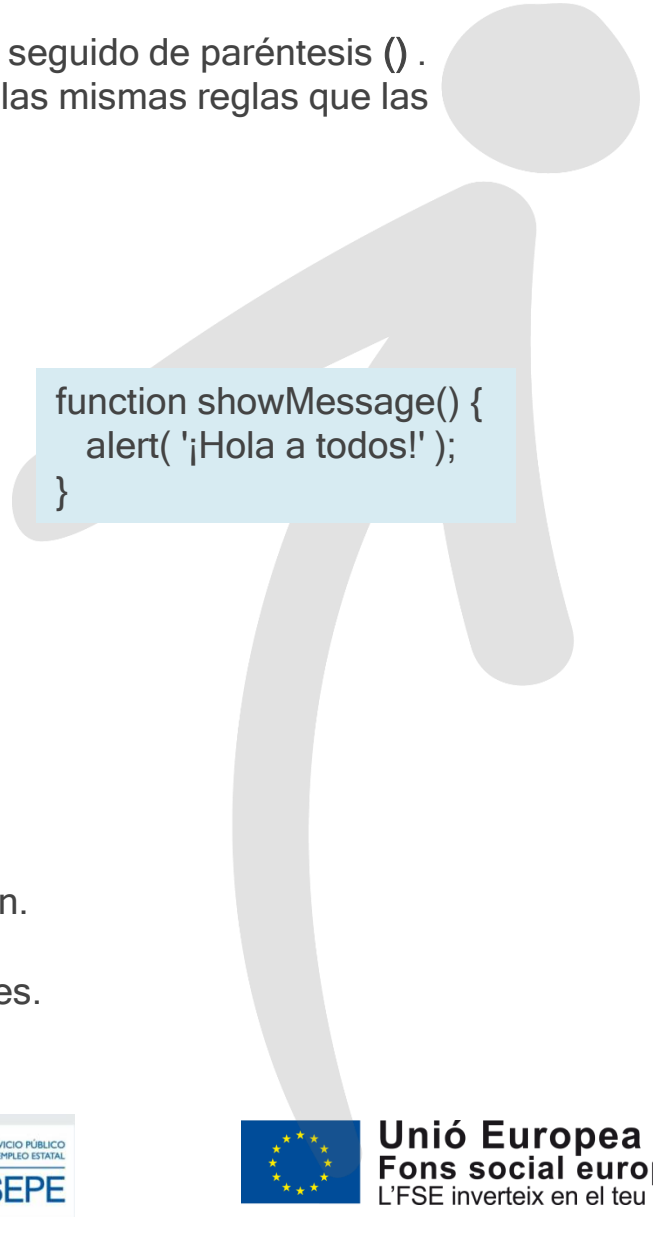
4.2. Funciones y métodos_ Declaración de funciones

Una función de JavaScript se define con la functionpalabra clave, seguida de un **nombre** , seguido de paréntesis **()** .
Los nombres de funciones pueden contener letras, dígitos, subrayados y signos de dólar (las mismas reglas que las variables).
Los paréntesis pueden incluir nombres de parámetros separados por comas:
(*parámetro1, parámetro2, ...*)

El código a ejecutar, por la función, se coloca entre corchetes: **{}**

```
function name(parameter1, parameter2, ... parameterN) { ...body... }
```

```
function showMessage() {
  alert( '¡Hola a todos!' );
}

showMessage();
showMessage();
```



```
function showMessage() {
  alert( '¡Hola a todos!' );
}
```

Los parámetros de la función se enumeran entre paréntesis () en la definición de la función.
Los argumentos de la función son los **valores** recibidos por la función cuando se invoca.
Dentro de la función, los argumentos (los parámetros) se comportan como variables locales.

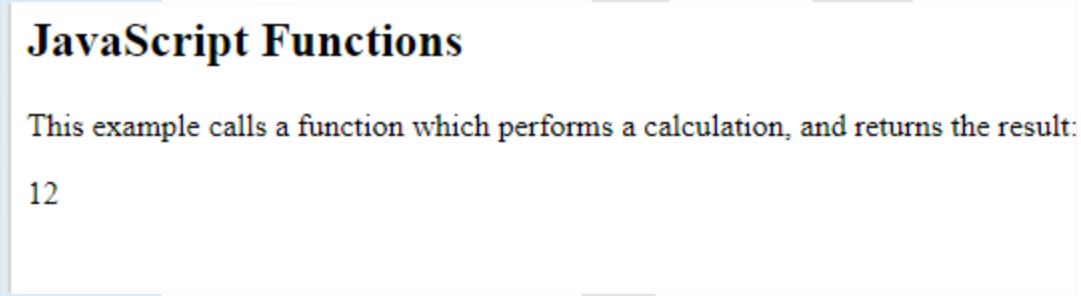
SOC Servei d'Ocupació de Catalunya
Generalitat de Catalunya
MINISTERIO DE TRABAJO Y ECONOMÍA SOCIAL
SERVICIO PÚBLICO DE EMPLEO ESTATAL SEPE
Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

4.2. Funciones y métodos_ Declaración de funciones

Una función de JavaScript es un bloque de código diseñado para realizar una tarea en particular.
Una función de JavaScript se ejecuta cuando "algo" la invoca (lo llama)

```
function myFunction(p1, p2) {
  return p1 * p2;   // The function returns the product of p1 and p2
}
```

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Functions</h2>

<p>This example calls a function which performs a calculation, and returns the
result:</p>

<p id="demo"></p>

<script>
function myFunction(p1, p2) {
  return p1 * p2;
}
document.getElementById("demo").innerHTML = myFunction(4, 3);
</script>

</body>
</html>
```



SOC Servei d'Ocupació de Catalunya
Generalitat de Catalunya
MINISTERIO DE TRABAJO Y ECONOMÍA SOCIAL
SERVICIO PÚBLICO DE EMPLEO ESTATAL SEPE
Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

4.2. Funciones y métodos _Ambito de variables Locales vs Externas

Las variables declaradas dentro de una función de JavaScript se vuelven **LOCALES** para la función.
Solo se puede acceder a las variables locales desde dentro de la función.

```
// code here can NOT use carName

function myFunction() {
  let carName = "Volvo";
  // code here CAN use carName
}

// code here can NOT use carName
```

Dado que las variables locales solo se reconocen dentro de sus funciones, las variables con el mismo nombre se pueden usar en diferentes funciones.
Las variables locales se crean cuando se inicia una función y se eliminan cuando se completa la función.

SOC Servei d'Ocupació de Catalunya
Generalitat de Catalunya
Ministerio de Trabajo y Economía Social
Servicio Público de Empleo Estatal SEPE
Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

4.2. Funciones y métodos _Parámetros

Podemos pasar datos arbitrarios a funciones usando parámetros.
En el siguiente ejemplo, la función tiene dos parámetros: from y text

```
function showMessage(from, text) {
  from = '*' + from + '*'; // hace que "from" se vea mejor
  alert( from + ': ' + text );
}

let from = "Ann";
showMessage(from, "Hola"); // *Ann*: Hola
// el valor de "from" es el mismo, la función modificó una copia local
alert( from ); // Ann
```

```
function showMessage(from, text) {
  // parámetros: from, text
  alert(from + ': ' + text);
}
showMessage('Ann', '¡Hola!'); // Ann: ¡Hola! (*)
showMessage('Ann', "¿Cómo estás?"); // Ann: ¿Cómo estás? (**)
```

SOC Servei d'Ocupació de Catalunya
Generalitat de Catalunya
MINISTERIO DE TRABAJO Y ECONOMÍA SOCIAL
SERVICIO PÚBLICO DE EMPLEO ESTATAL SEPE
Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

4.2. Funciones y métodos _Valores prederminados

Si una función es llamada pero no se le proporciona un argumento, su valor correspondiente se convierte en undefined.

```
showMessage("Ann");
```

```
function showMessage(from, text = "sin texto") {
  alert( from + ": " + text );
}
```

```
function showMessage(from, text = anotherFunction()) {
// anotherFunction() solo se ejecuta si text no fue asignado
// su resultado se convierte en el valor de texto
}
```

```
function showMessage(text) {
  // ...
  if (text === undefined) { // si falta el parámetro
    text = 'mensaje vacío';
   }
   alert(text);
}

showMessage(); // mensaje vacío
```

```
function showCount(count) {
  // si count es undefined o null, muestra "desconocido"
  alert(count ?? "desconocido");
}

showCount(0); // 0
showCount(null); // desconocido
showCount(); // desconocido
```

4.2. Funciones y métodos _Devolviendo un valor

Cuando JavaScript llega a una return de claración, la función dejará de ejecutarse.
Si la función se invocó desde una declaración, JavaScript "regresará" para ejecutar el código después de la declaración de invocación.
Las funciones suelen calcular un **valor de retorno** . El valor de retorno se "devuelve" a la "persona que llama":

```
let x = myFunction(4, 3);   // Function is called, return value will end up in x

function myFunction(a, b) {
  return a * b;           // Function returns the product of a and b
}
```

4.2. Funciones y métodos _Devolviendo un valor

Convertir Fahrenheit a Celsius:

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Functions</h2>

<p>This example calls a function to convert from Fahrenheit to
Celsius:</p>
<p id="demo"></p>

<script>
function toCelsius(f) {
  return (5/9) * (f-32);
}
document.getElementById("demo").innerHTML = toCelsius(77);
</script>

</body>
</html>
```

El operador () invoca la función

Usando el ejemplo anterior, toCelsiusse refiere al objeto de la función y toCelsius()se refiere al resultado de la función. Acceder a una función sin () devolverá el objeto de la función en lugar del resultado de la función.

```
function toCelsius(fahrenheit) {
  return (5/9) * (fahrenheit-32);
}
document.getElementById("demo").innerHTML = toCelsius;
```

4.2. Funciones y métodos _Nomenclatura de funciones

- Las funciones son acciones.
- Suele ser un verbo.
- Debe ser breve
- Lo más preciso posible
- Describir lo que hace la función

Por ejemplo, funciones que comienzan con "show" usualmente muestran algo.

Funciones que comienza con…
- "get…" – devuelven un valor,
- "calc…" – calculan algo,
- "create…" – crean algo,
- "check…" – revisan algo y devuelven un boolean, etc.

Ejemplos de este tipo de nombres:

```
showMessage(..) // muestra un mensaje
getAge(..) // devuelve la edad (la obtiene de alguna manera)
calcSum(..) // calcula una suma y devuelve el resultado
createForm(..) // crea un formulario (y usualmente lo devuelve)
checkPermission(..) // revisa permisos, y devuelve true/false
```

Una función debe hacer exactamente lo que sugiere su nombre, no más.
- getAge – está mal que muestre una alert con la edad (solo debe obtenerla).
- createForm – está mal que modifique el documento agregándole el form (solo debe crearlo y devolverlo).
- checkPermission – está mal que muestre el mensaje acceso otorgado/denegado(solo debe realizar la verificación y devolver el resultado).

SOC Servei d'Ocupació de Catalunya
Generalitat de Catalunya
Ministerio de Trabajo y Economía Social SEPE
Unió Europea Fons social europeu L'FSE inverteix en el teu futur

4.2. Funciones y métodos _Funciones == Comentarios

Las funciones deben ser cortas y hacer exactamente una cosa.

```
function showPrimes(n) {
    nextPrime: for (let i = 2; i < n; i++) {
          for (let j = 2; j < i; j++) {
             if (i % j == 0) continue nextPrime;
          }
     alert( i ); // un número primo }
}
```

```
function showPrimes(n) {
   for (let i = 2; i < n; i++) {
     if (!isPrime(i)) continue;

     alert(i); // a prime
    }
}

function isPrime(n) {
    for (let i = 2; i < n; i++) {
       if ( n % i == 0) return false;
    }
    return true;
}
```

En matemáticas, un número primo es un número natural mayor que 1 que tiene únicamente dos divisores positivos distintos: él mismo y el 1. Por el contrario, los números compuestos son los números naturales que tienen algún divisor natural aparte de sí mismos y del 1, y, por lo tanto, pueden factorizarse.

SOC Servei d'Ocupació de Catalunya
Generalitat de Catalunya
Ministerio de Trabajo y Economía Social SEPE
Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

4.2. Funciones y métodos _Ejercicios 421

¿Es "else" requerido?

La siguiente función devuelve true si el parámetro age es mayor a 18. De lo contrario, solicita una confirmación y devuelve su resultado:

```
function checkAge(age) {
  if (age > 18) {
    return true;
  } else {
    // ... return confirm('¿Tus padres te permitieron?');
  }
}
```

¿Funcionará la función de manera diferente si se borra else?

```
function checkAge(age) {
  if (age > 18) {
    return true;
  }
  // ...
  return confirm('¿Tus padres te permitieron?');
}
```

¿Hay alguna diferencia en el comportamiento de estas dos variantes?

Ninguna diferencia.

SOC Servei d'Ocupació de Catalunya
Generalitat de Catalunya
Ministerio de Trabajo y Economía Social SEPE
Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

4.2. Funciones y métodos _Ejercicios 422

Reescribe la función utilizando '?' o '||'

La siguiente función devuelve true si el parámetro age es mayor que 18.

De lo contrario, solicita una confirmación y devuelve su resultado.

```
function checkAge(age) {
  if (age > 18) {
    return true;
  } else {
    return confirm('¿Tienes permiso de tus padres?');
  }
}
```

Reescríbela para realizar lo mismo, pero sin if, en una sola linea.
Haz dos variantes de checkAge:

1. Usando un operador de signo de interrogación ?
2. Usando OR ||

Usando un operador signo de pregunta '?':

```
function checkAge(age) {
    return (age > 18) ? true : confirm('¿Tus padres te lo permitieron?');
}
```

Usando Ó || (la variante más corta):

```
function checkAge(age) {
  return (age > 18) || confirm('¿Tus padres te lo permitieron?');
}
```

Tenga en cuenta que aquí los paréntesis alrededor de `age > 18` no son requeridos. Existen para una mejor legibilidad.

SOC Servei d'Ocupació de Catalunya
Generalitat de Catalunya
MINISTERIO DE TRABAJO Y ECONOMÍA SOCIAL
SERVICIO PÚBLICO DE EMPLEO ESTATAL SEPE
Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

4.2. Funciones y métodos _Ejercicios 423

Función min(a, b)

Escriba una función min(a,b) la cual devuelva el menor de dos números a y b.

Por ejemplo:

```
min(2, 5) == 2
min(3, -1) == -1
min(1, 1) == 1
```

Una solución usando if:

```
function min(a, b) {
    if (a < b) {
       return a;
     } else {
     return b;
     }
}
```

Una solución con un operador de signo de interrogación '?':

```
function min(a, b) {
    return a < b ? a : b;
}
```

P.D: En el caso de una igualdad `a == b` No importa qué devuelva.

4.2. Funciones y métodos _Ejercicios 424

Función pow(x,n)

Escriba la función pow(x,n) que devuelva x como potencia de n. O, en otras palabras, multiplique x por si mismo n veces y devuelva el resultado.

```
pow(3, 2) = 3 * 3 = 9
pow(3, 3) = 3 * 3 * 3 = 27
pow(1, 100) = 1 * 1 * ...* 1 = 1
```

Cree una página web que solicite x y n, y luego muestre el resultado de pow(x,n).

PD: En esta tarea, la función solo debe admitir valores naturales de n: enteros desde 1.

```
let x = prompt('x?');
let n = prompt('n?');
function pow(x, n) {
  let b = x ** n;
  if (n < 1) {
    alert('invalid number');
  }
  return b;
}
alert(pow(x, n));
```

```
function pow(x, n) {
        let result = x;

        for (let i = 1; i < n; i++) {
            result *= x;
        }
        return result;
    }
    let x = prompt("x?", '');
    let n = prompt("n?", '');

    if (n < 1) {
        alert(`Potencia ${n} no soportada,
  use un entero mayor a 0`);
    } else {
        alert(pow(x, n));
    }
```

```
function pow(x, n) {
        return pow = (n < 1) ? "n es invalida" : pow = x ** n;
    }

    let x = prompt("x?", 0)
    let y = prompt("y?", 0)

    alert(pow(x, y));
```

Grup CIEF

el teu partner

Barcelona

Francesc Tàrrega 14

08027 Barcelona

93 351 78 00

Madrid

Campanar 12

28028 Madrid

91 502 13 40

Reus

Alcalde Joan Bertran 34-38

43202 Reus

977 31 24 36

info@grupcief.com