

Fonaments de la programació

Crear funcions en JavaScript

`{.js}`
JavaScript



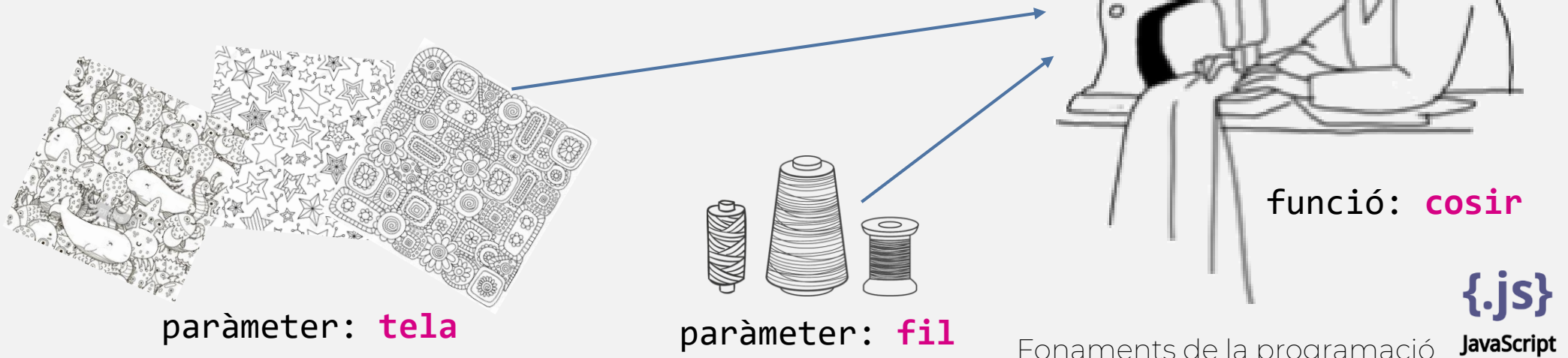
UNIÓ EUROPEA
Fons Europeu de
Desenvolupament Regional

Què és una funció?

Una funció és **un conjunt d'instruccions** que **realitzen** unes determinades **tasques** i a les que **podem invocar mitjançant el seu nom**.

Les funcions **permeten reutilitzar el codi**, estalviant temps de programació.

- ✓ Una funció només s'executa quan s'anomena (es crida).
- ✓ Podem o no, **passar-li dades**, coneguts com a **paràmetres**.
- ✓ També **poden retornar valors** si així es requereix.



Estructura d'una funció

```
function nomFuncio([parameter1], [parameter2], ... [parameterN]) {  
    ... instruccions...  
    [return valor;]  
}
```

nomFuncio:

Nom que se li dóna a la funció.

Cal seguir les mateixes normes que per crear noms de variables.

Llista paràmetres (opcional): Llista de paràmetres, **també anomenats arguments**, separats per comes. Aquests són les dades d'entrada del Funció.

Un Funció pot rebre zero o més arguments.

Els parèntesis a continuació del nom del Funció són obligatoris encara que estiguin buits.

return:

Una funció pot devolver un valor al codi de trucada com a resultat i per fer-ho utilitzem la instrucció **return**



Cal tenir molta cura del lloc on es declaren les variables.

El codi fora de la funció no veu les variables locals.

Exemples

Funció sense paràmetres ni return

```
...  
<input type='button' onClick='saludar()' Value='Clica aquí' />  
  
<script>  
  
    function saludar(){  
        alert("Hola tothom");  
    }  
</script>  
...
```

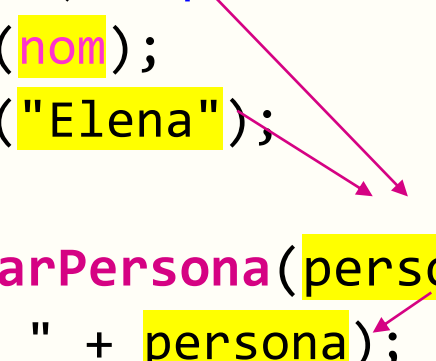
Esta página dice

Hola tothom

Aceptar

Funció amb paràmetres i sense return

```
...  
let nom = prompt("A qui vols saludar?" , "");  
saludarPersona(nom);  
saludarPersona("Elena");  
  
function saludarPersona(persona){  
    alert("Hola " + persona);  
}  
...
```



Esta página dice

A qui vols saludar?

Aceptar

Cancelar

Esta página dice

Hola Laura

Aceptar

Esta página dice

Hola Elena

Aceptar

Funció sense paràmetres i amb return

```
<p id="mostraMes"></p>
<input type='button' onClick='mesHoy()' Value='Mes actual' />
```

```
...
function mesHoy() {
    document.getElementById("mostraMes").innerHTML = "Estamos en " + mesActual();
}
function mesActual() {
    let meses = ["Gener", "Febrer", "Març", "Abril", "Mai", "Juny", "Julliol",
    "Agost", "Setembre", "Octobre", "Novembre", "Desembre"];

    let mesNumero = new Date(); //Data ordinador
    let mes = meses[mesNumero.getMonth()];

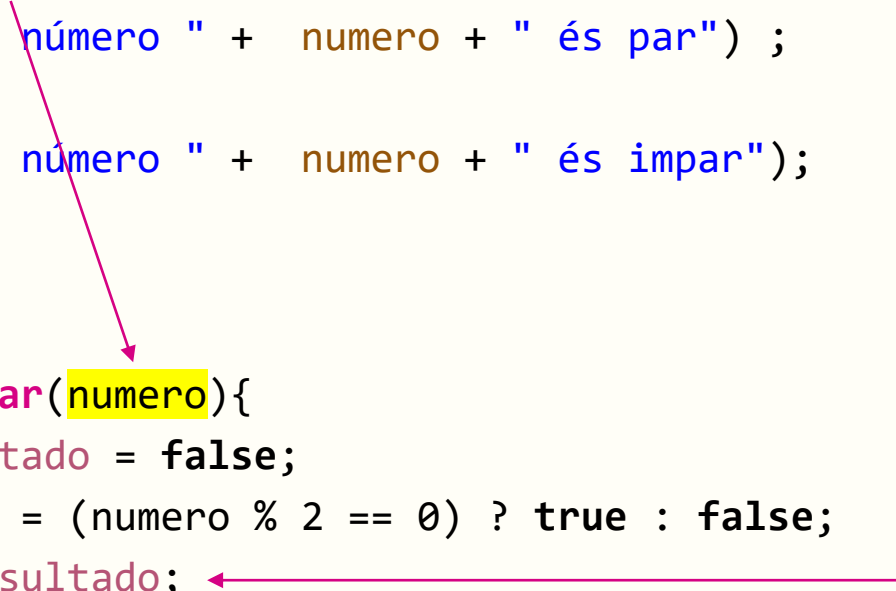
    return mes;
} // fin funció mesActual...
```

Estamos en Gener

Mes actual

Funció amb paràmetres i amb return

```
...  
let numero = Number(prompt("Indica un número enter: " , ""));  
if (esPar(numero)) { // Crida la funció  
    alert("El número " + numero + " és par" ) ;  
} else {  
    alert("El número " + numero + " és impar");  
}  
...  
  
function esPar(numero){  
    let resultado = false;  
    resultado = (numero % 2 == 0) ? true : false;  
    return resultado;  
}
```



A pink arrow points from the `esPar(numero)` call in the `if` statement to the `esPar` function definition. Another pink arrow points from the `return resultado;` line to a text box on the right.

La funció torna un valor booleà.

Invocació de funcions

El codi dins de la funció s'executarà quan "alguna cosa" invoqui (truca) la funció:

- Quan es produeix un esdeveniment (per exemple, quan un usuari fa clic en un botó)
Exemple: `<input type='button' onClick='mesHoy()' Value='Mes actual' />`
(Diapositives 6 y 8)
- Quan s'invoca des del codi JavaScript
Exemple: `if (esPar(numero)) { ...`
(Diapositives 7 y 9)
- Automàticament (autoinvocat)
(Se verán en el itinerario)