



Requisits per al bon funcionament de la sessió

Regles d'ús durant la sessió



Recordeu mantenir desconnectades la vostra càmera i micròfon per evitar interferències.



Les preguntes es fan principalment pel xat. Si no disposeu d'aquesta icona, pengeu i torneu a accedir. Si segueix sense aparèixer, activeu el micròfon al final de la sessió.



Si us plau, no activeu aquesta opció: compartir pantalla.



En els pròxims dies, les persones inscrites rebran per correu electrònic la presentació i els recursos utilitzats durant aquesta sessió.

Moltes gràcies! Comencem en breu...



Ajuntament
de Barcelona

shorturl.at/fF156

Aprèn a fer animacions amb Jquery

Amb JavaScript i HTML

Maig 2022



La nostra missió

Impulsar la política econòmica i el desenvolupament local per promoure la millora de la qualitat de vida de les ciutadanes i ciutadans de Barcelona a través del foment de l'ocupació, l'impuls de l'emprenedoria i el suport a les empreses, responenent a les diferents necessitats de les persones en el seu territori i des de la perspectiva de l'economia plural, que inclou, entre altres, l'economia social i solidària.

Potenciar l'empoderament de la ciutadania i el reequilibri entre territoris per assolir un model just de desenvolupament econòmic, així com de creació, manteniment i repartiment de l'ocupació.





Què fem?

Acompanyem la ciutadania durant tot el procés de recerca de feina

barcelonactiva.cat/treball

Donem suport a les persones emprenedores per fer realitat la seva idea de negoci, sigui de forma col·lectiva, comunitària o individual

barcelonactiva.cat/emprenedoria

I tot això ho fem en clau de territori, incloent la perspectiva de gènere i la diversitat des d'una visió d'economia plural

Ajudem les empreses i organitzacions a créixer, connectar-se amb l'ecosistema i consolidar-se amb models socialment responsables

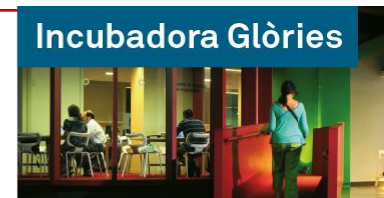
barcelonactiva.cat/empreses

Oferim formació tecnològica a les persones que cerquen feina, emprenedores i professionals

barcelonactiva.cat/cibernarium



Equipaments al servei de la ciutat



Barcelona Activa, molt present als barris



barcelona.cat/barcelonactiva



¿Qué es jQuery?

jQuery permite manipular elementos del DOM (textos, imágenes, enlaces, etc.) , cambiar el diseño CSS o realizar peticiones Ajax.



Bootstrap 4, necesita jQuery para funcionar



Importar JQuery

1. Descargar de la [web](#) de JQuery e importar en el fichero:

```
<head>  
<script src="jquery-3.3.1.min.js"></script>  
</head>
```

2. Importar del repositorio de Google:

```
<head>  
<script  
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1  
/jquery.min.js"></script>  
</head>
```




Creamos la estructura en JQuery

```
<!DOCTYPE html>
<html>
<head>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
</script>
  <script>
    $(document).ready(function(){
      alert("Hola mundo");
    });
  </script>
</head>
<body>
  Mi web
</body>
</html>
```

Código:

https://drive.google.com/file/d/1EY2OtXlnr3KyHL-W_gX7i8LgNqisARvS/view?usp=sharing



¿Tenemos el JQuery Cargado? Hay 2 formas

Se ejecutará solamente cuando el Modelo de Objetos del Documento (DOM) esté listo para ejecutar código Javascript.

```
$(document).ready(function(){  
  
    // jQuery methods go here...  
  
});
```

```
$(function(){  
  
    // jQuery methods go here...  
  
});
```



Sintaxis básica

`$(selector).action();`

- **\$** define que utilizamos JQuery
- El ("**selector**") para buscar un elemento HTML
- **action()** para hace algo con ese elemento

Ejemplos:

`$("p").hide()` - esconde todos los elementos <p>.

`$(".test").hide()` - Esconde todos los elementos con class="test".

`$("#test").hide()` - Esconde el elemento con id="test".

https://www.w3schools.com/jquery/jquery_syntax.asp



Ejercicio 0: \$("selector").text();

Injecta texto plano en el div con id "mi_texto"

```
$("#mi_texto").text("Hello World");
```



\$("#selector").html(); vs \$("#selector").append();

```
<!DOCTYPE html>
<html>
<head>
<script src="...jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#myBtn").click(function() {
        $("#demo").html("Soy un texto");
    });
});
</script>
</head>
<body>
    <div id="demo"></div>
    <button id="myBtn">Click me</button>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
<script src="...jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#myBtn").click(function() {
        $("#demo").append("Soy un texto");
    });
});
</script>
</head>
<body>
    <div id="demo"></div>
    <button id="myBtn">Click me</button>
</body>
</html>
```



Ejercicio 1

Dado el siguiente HTML:

https://github.com/ITAcademyProjects/Jquery/blob/master/jquery_ex_1.html

Utiliza JQuery para eliminar los párrafos cuando se haga click en ellos (uno a uno).

[Solución](#)



Pista del ejercicio 1

```
$("#__").remove();
```



Ejercicio 2

Dado el siguiente HTML:

https://github.com/ITAcademyProjects/Jquery/blob/master/jquery_ex_2.html

Modifica el texto que hay en <p> según el botón seleccionado.

https://www.w3schools.com/jquery/jquery_dom_set.asp

[Solución](#)



Pista del ejercicio 2

```
var str = $(this).text();  
$("#____").html( str );
```



Ejercicio 3

Dado el siguiente HTML:

https://github.com/ITAcademyProjects/Jquery/blob/master/jquery_ex_3.html

Modifica el color del texto en <p> según el botón.

https://www.w3schools.com/jquery/jquery_css.asp

[Solución](#)



Pista del ejercicio 3

```
$("___").css("color", "#3262FF");
```



Ejercicio 4

Dado el siguiente HTML:

https://github.com/ITAcademyProjects/Jquery/blob/master/jquery_ex_4.html

Al pulsar el botón mostrar capturaremos el texto que hemos escrito en el input en <p>:

https://www.w3schools.com/jquery/jquery_dom_set.asp

[Solución](#)



Pista del ejercicio 4

```
var valor = $("#input_texto").val();
```



Ejercicio 5: \$("selector").animate();

Con jQuery animate() podemos modificar los estilos CSS de los elementos de nuestra web y realizar pequeñas animaciones

- https://www.w3schools.com/jquery/jquery_animate.asp
- https://www.w3schools.com/jquery/jquery_fade.asp

[Solución](#)



Repaso de Javascript



Hola Mundo

1. Crear el fichero hola.html
2. Añadir el siguiente script (antes del `<head>`)

```
<script type="text/javascript">  
    console.log("Hola mundo");  
</script>
```

3. Con el F12 abrir la consola y ver el resultado



Comentar un código

1. En javascript podremos comentar:
 - a. una línea:

```
//alert('hola mundo!');
```

- b. un varias líneas

```
/*alert('hola mundo!');  
alert('hola me llamo Ismael');*/
```



Variables

1. ¿Cómo se asignan las variables?

- `var usuario = 'Pepe';`
- `var precio = 12.50;`
- `var es_premium = true`

2. Crea un frase utilizando 3 variables. Utiliza el concatenador +



Operador aritmético

1. `var importe_a=100; var importe_b=200;`
2. Realiza el sumatorio de las dos variables y muestra el resultado por consola.



Funciones básicas aritméticas

1. `console.log(Math.round(0.7)); -> 1`
2. `console.log(Math.ceil(7.004)); -> 8`
3. `console.log(Math.floor(7.9)); -> 7`
4. `console.log(Math.pow(2,2)); -> 4`



Funciones básicas

1. Reemplazar:

```
var nombre = 'Ismael';  
nombre = nombre.replace('a','e');  
console.log(nombre);
```

2. Obtener un trozo de una cadena o array:

```
var nombre = 'Ismael';  
var nombre_cortado = nombre.slice(0,2);  
console.log(nombre_cortado);  
  
var nombres = ['Rita', 'Pedro', 'Miguel', 'Ana', 'Vanessa'];  
var masculinos = nombres.slice(1, 3);  
console.log(masculinos);
```



If ... else ...

1. Ejemplo:

```
var edad = 12;  
if(edad >= 18) console.log("Soy mayor de edad");  
else console.log("Soy menor de edad");
```

2. Crea un programa que reciba una nota de examen y muestre por consola si esta aprobado o suspendido.



If ... else ... con operador && ||

Crea un programa que reciba dos variables:

- precio
- producto

Si el producto es un libro y su precio es superior a 200 euros. Indicaremos por pantalla:

El/la _____ cuesta _____ euros y es muy caro.



Condicional con ===

Replica el ejercicio y reflexiona:

```
var numero_uno = 30;  
var numero_dos = "30"  
  
if(numero_uno == numero_dos){  
    console.log("Hay un 30");  
}  
  
if(numero_uno === numero_dos){  
    console.log("Hay un número 30");  
}
```




Bucle For

1. Bucle simple:

```
//El limite es 10
for(var i=0;i<10;i++){
    console.log(i)
}
```

2. Bucle de un array:

```
var array_nombres = ['Ismael','Jake','Xavi','Martí'];
console.log(array_nombres.length); //retorna 4
for(var i=0;i<array_nombres.length;i++){
    console.log(array_nombres[i]);
}
```



Bucle For

Ejercicio:

Crea un array de precios y muestra por pantalla aquellos que cuesten entre 3 y 5 euros.



Array multidimensional

```
var items = [  
  ['coche', 20000],  
  ['moto', 4000],  
  ['avion', 12000000]  
];  
console.log("Mi "+items[0][0]+" cuesta "+items[0][1]+ " euros.");
```



Funciones

```
<html>
  <head>
    <script type="text/javascript">
      function alerta() {
        alert('hola soy alert!');
      }
    </script>
  </head>
  <body>
    <span onclick="alerta();">Clic aquí</span>
  </body>
</html>
```



Funciones

1. Crea una función que reciba dos números y los sume.
2. Crea una función que reciba dos números los reste y devuelva el resultado.

Pista:

`return` resultado;



Acceder al DOM

1. Acceder al contenido de un id

```
<script type="text/javascript">  
function alerta() {  
    var boton = document.getElementById('boton_1').textContent;  
    console.log(boton)  
}  
</script>  
<span onclick="alerta();" id="boton_1">Clic aquí</span>
```



forEach

1. Bucle de nombres inteligente

```
var array_nombres = ['Ismael', 'Jake', 'Xavi', 'Martí'];  
array_nombres.forEach(function(elemento) {  
    console.log("Nombre: "+elemento);  
})  
);
```



map

```
var array_nombres = ['Ismael', 'Jake', 'Xavi', 'Martí'];
var mi_nuevo_nombre = [];
array_nombres.forEach(function(elemento) {
    mi_nuevo_nombre.push("Mi nombre es1: "+ elemento)
});
console.log(mi_nuevo_nombre);

var array_nombres = ['Ismael', 'Jake', 'Xavi', 'Martí'];
var mi_nuevo_nombre = array_nombres.map(function(elemento) {
    return "Mi nombre es2: "+ elemento;
});
console.log(mi_nuevo_nombre);
```




Ejemplo avanzado de map

```
var almuerzos = [  
  {principal: 'macarrones', postre: 'helado'},  
  {principal: 'burger', postre: 'manzana'},  
  {principal: 'verdura', postre: 'flan'}  
];  
var platosPrincipales = almuerzos.map(function(almuerzo) {  
  return almuerzo.principal;  
});  
console.log(platosPrincipales);
```



JqueryValidate



Material para la MasterClass

<https://drive.google.com/open?id=1SR2ImH9LPlgl1oeK91dK9klzq8swilg6>



¿Para qué sirve el plugin de JQuery Validate?

El plugin jQuery Validate para validar formularios del lado del cliente:

Formulario de Login

Email

Este campo es obligatorio.

Contraseña

Este campo es obligatorio.

Login



Instalación del plugin jQueryValidate

¿Qué necesitamos?

1. JQuery

```
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>
```

2. JQueryValidate (librería de validación)

```
<script src="https://jquery.validate.min.js"></script>
```

3. Configuración de validaciones

```
<script src="js/login_validation.js"></script>
```



Partimos del formulario de login en HTML

```
<form id="login-form">
  <p>Formulario de Login</p>
  <div class="form-row">
    <div class="form-group col-md-6">
      <label class="text-dark" for="login-email">Email</label>
      <input class="form-control" name="login-email" type="email">
    </div>
  </div>
  <div class="form-row">
    <div class="form-group col-md-6">
      <label class="text-dark" for="login-password">Contraseña</label>
      <input class="form-control" name="login-password" type="password">
    </div>
  </div>
  <div class="form-row mt-2">
    <input class="btn" id="submit-button" type="submit" value="Login">
  </div>
</form>
```



Enviamos el formulario con el evento SUBMIT

```
$('#login-form').on('submit', function() {  
  
    if ( $('#login-form').valid() ) {  
        alert("formulario válido");  
        //form.submit();  
    } else {  
        alert("formulario NO válido");  
    }  
  
});
```

Por el momento no hay ninguna restricción en el momento de validar, por lo tanto, siempre tendremos “el formulario válido”.



Configuramos nuestra validación de login

(js/login_validation.js)

```
$("#login-form").validate({
  rules: {
    "login-email": {
      required: true,
      email: true
    },
    "login-password": {
      required: true,
      minlength: 5
    }
  },
  messages: {
    "login-email": {
      required: "Este campo es obligatorio.",
      email: "Por favor, escribe una dirección de correo válida",
    },
    "login-password": {
      required: "Este campo es obligatorio.",
      minlength: "Por favor, no escribas menos de {0} caracteres.",
    },
  },
});
```




Resultado final

Formulario de Login

Email

✖

Por favor, escribe una dirección de correo válida

Contraseña

✖

Por favor, no escribas menos de 5 caracteres.

Login



Tipos “rules” que existen:

Existen 2 tipo de “rules”:

- 1) Los que nos vienen incluidos en la librería (los típicos)
- 2) Los que podemos crear de forma personalizada



Listado de “Rules” típicos

-
- `required` – Makes the element required.
 - `remote` – Requests a resource to check the element for validity.
 - `minlength` – Makes the element require a given minimum length.
 - `maxlength` – Makes the element require a given maximum length.
 - `rangelength` – Makes the element require a given value range.
 - `min` – Makes the element require a given minimum.
 - `max` – Makes the element require a given maximum.
 - `range` – Makes the element require a given value range.
 - `step` – Makes the element require a given step.
 - `email` – Makes the element require a valid email
 - `url` – Makes the element require a valid url
 - `date` – Makes the element require a date.
 - `dateISO` – Makes the element require an ISO date.
 - `number` – Makes the element require a decimal number.
 - `digits` – Makes the element require digits only.
 - `equalTo` – Requires the element to be the same as another one
-



Ejercicio 1

Crear un campo extra en el formulario de login
y validarlo:

login.html



Ejemplo de un formulario de registro

```
<script>
$( "#register" ).validate({
  rules: {
    "first_name": "required",
    "last_name": "required",
    "email": {required: true, email: true},
    "pass": {required: true, minlength: 8} ,
    "pass2": {required: true, equalTo: "#f_pass"},
  },
  messages: {
    "first_name": {required: "Por favor introduzca su nombre"},
    "last_name": {required: "Por favor introduzca su apellido"},
    "email": {required: "Por favor introduzca su dirección de correo electrónico",
email: "Correo inválido"},
    "pass": {required: "Ingrese una contraseña", minlength: "La contraseña debe
tener al menos 8 caracteres"},
    "pass2": {required: "Repita la contraseña elegida", equalTo: "Las contraseñas
no coinciden"},
  }
});
</script>
```



Ejercicio 2

Entender y mejorar la validación del siguiente formulario de registro:

register.html

¡Ojo que la validación de city NO funciona!



datatable

Exemple:

https://drive.google.com/file/d/1jrCDj6_cVzEDt8mUndoNDTm0-ojasAtD/view?usp=sharing



ajax - JQuery

Exemple:

<https://drive.google.com/drive/folders/131UZfjHBfrGV3CT16-EjZZjxIS8SX5H?usp=sharing>



Barcelona
Activa



barcelona.cat/barcelonactiva