
JQuery

1.- Introducció JQUERY – DOM - AJAX

• JQUERY

<https://jquery.com/>

jQuery és una llibreria de **JavaScript** molt popular, que simplifica la manipulació del **DOM** (Document Object Model) i la gestió d'esdeveniments (**events**) en pàgines web.

El lema de **jQuery** és "**write less, do more**" (escriu menys, fes més) i ens ofereix una manera eficient de treballar amb JavaScript en el desenvolupament web.

Les seues principals característiques són:

- És la llibreria JavaScript més utilitzada, amb una ampla adopció a nivell mundial.
- Conté una sintaxi senzilla i funcions optimitzades, el que simplifica enormement la tasca de codificar en **JS**.
- Facilita la selecció i manipulació d'elements del DOM.
- Simplifica el maneig d'esdeveniments.
- Ofereix una àmplia gamma d'efectes i animacions.
- Proporciona compatibilitat amb Ajax per realitzar crides asíncrones al servidor.
- Proporciona compatibilitat amb Ajax per realitzar crides asíncrones al servidor.

Nota: Per entendre el concepte de llibreria. Podriem dir que **una llibreria** és com una caixa d'eines plena de funcions que ja estan escrites per altres persones.

En comptes de començar des de zero cada vegada que necessitem fer alguna cosa, podem usar les funcions de la llibreria per fer-ho més fàcilment i ràpidament. Amb llibreries com **jQuery**, podem realitzar tasques complicades en la construcció de pàgines web de manera més senzilla i eficient, ja que ja tenen les funcions necessàries per fer-ho. És com tenir un conjunt d'eines que et faciliten la feina.

• Funcions de jQuery

De totes les funcions que ens ofereix **jQuery** per a re diverses tasques en el desenvolupament web. Algunes de les funcions més importants inclouen:

- **Manipulació del DOM:** Per afegir, eliminar o modificar elements de la pàgina web (document HTML) dinàmicament.
- **Manipulació de CSS:** Per canviar estils CSS dels elements seleccionats, i per tant. L'aspecte de la web
- **Gestió d'esdeveniments:** Afegir o eliminar gestors d'esdeveniments com **clics**, **desplaçaments**, etc.
- **Efectes i Animacions:** Per crear animacions i efectes visuals en els elements del document.
- **AJAX:** Per realitzar crides asíncrones al servidor i actualitzar parts de la pàgina web sense necessitat de recarregar-la completament.
- **Utilitats:** Per a diverses tasques com treballar amb arrays, objectes, etc.

jQuery també inclou altres funcions avançades i pràctiques com l'animació, la manipulació d'esdeveniments, el tractament de formularis i la validació, entre altres, que fan que siga una eina valuosa per a qualsevol desenvolupador web.

• AJAX

AJAX, o **Asynchronous JavaScript and XML**, és una tecnologia utilitzada en desenvolupament web que **permet als navegadors web realitzar crides asíncrones al servidor sense necessitat de recarregar completament la pàgina web**. Això significa que es poden **actualitzar les parts específiques de la pàgina web amb dades noves del servidor** en segon pla, sense interrompre l'experiència de l'usuari.

Les cridades **AJAX** impliquen enviar una **petició HTTP** al servidor des de la pàgina web, normalment a través de JavaScript. Aquesta petició pot ser:

- **per recuperar dades**, com ara informació de la base de dades o contingut dinàmic
- **per enviar dades al servidor**, com ara formularis emplenats per l'usuari.

Un cop rebuda la resposta del servidor, **JavaScript** pot manipular el contingut de la pàgina per actualitzar-la amb les dades noves, tot sense necessitat de recarregar la pàgina completa.

jQuery facilita l'ús d'**AJAX** amb les seves funcions i mètodes simplificats. En lloc d'escriure el codi AJAX pur, que pot ser complicat i repetitiu, **jQuery** ofereix mètodes com **\$.ajax()**, **\$.get()**, i **\$.post()** que simplifiquen la creació i gestió de cridades AJAX, amb només unes poques línies de codi, fent que siga més fàcil implementar funcionalitats com la càrrega dinàmica de contingut, l'enviament de dades de formulari, o l'actualització de la interfície d'usuari sense recarregar la pàgina completa.

En resum, **AJAX** és una tecnologia que permet actualitzar parts específiques d'una pàgina web sense necessitat de recarregar-la completament, i **jQuery** facilita l'ús d'AJAX amb mètodes simplificats i fàcils d'utilitzar. Això fa que sigui més senzill per als desenvolupadors implementar interaccions dinàmiques i en temps real en les seves aplicacions web.

• Integració jQuery, Dom i Ajax

La integració del **DOM**, **AJAX** i **jQuery** permet als desenvolupadors web crear aplicacions web dinàmiques i interactives amb facilitat i eficàcia

El DOM (Document Object Model) proporciona una representació estructurada d'un document HTML o XML com un arbre de nodes, permetent accedir i manipular el contingut, l'estil i la estructura d'una pàgina web mitjançant codi JavaScript. **El DOM és la base que permet la interactivitat i la dinàmica en una pàgina web.**

AJAX (Asynchronous JavaScript and XML): és una tecnologia que permet fer crides asíncrones al servidor des de JavaScript sense necessitat de recarregar completament la pàgina web. Això permet actualitzar parts específiques de la pàgina amb dades noves del servidor en segon pla, oferint una experiència d'usuari més fluida i ràpida.

La integració del **DOM**, **AJAX** i **jQuery** és comuna en el desenvolupament web modern. Per exemple:

- **Manipulació del DOM amb jQuery:** jQuery facilita l'accés i la manipulació del DOM amb mètodes com **\$(selector).html()**, **\$(selector).text()**, **\$(selector).attr()**, etc. Això permet als desenvolupadors canviar el contingut, l'estil i la estructura de la pàgina web de manera més senzilla i concisa.

- **Ús d'AJAX amb jQuery:** jQuery simplifica la creació i gestió de crides AJAX amb mètodes com **\$.ajax()**, **\$.get()**, **\$.post()**, etc. Això permet als desenvolupadors realitzar crides asíncrones al servidor i actualitzar parts específiques de la pàgina web sense necessitat de recarregar-la completament.
- **Gestió d'esdeveniments amb jQuery:** jQuery facilita l'afegiment de gestors d'esdeveniments als elements del DOM amb mètodes com **\$(selector).click()**, **\$(selector).hover()**, **\$(selector).on()**, etc. Això permet als desenvolupadors respondre a les accions de l'usuari de manera més senzilla i eficient.

En els últims anys, JQuery està experimentant una disminució en la seua popularitat com a resultat de l'evolució de les tecnologies web, encara que segueix sent ampliament utilitzat, la tendència actual és utilitzar tecnologies més modernes i frameworks front-end com React, Vue.js i Angular, així com l'ús de JavaScript modern (ES6+).

1.- Inicialització jQuery

- **Inicialitzar jQuery**

Hem d'agregar **jQuery** a la pàgina web per a que estiga disponible per a ser utilitzat en el codi de **JavaScript**. Podem agregar **jQuery** des d'un **CDN** extern (**Content Delivery Network**).

Un **CDN**, o **Xarxa de Distribució de Contingut**, és una xarxa de servidors distribuïts globalment que emmagatzemen i serveixen contingut web, com arxius multimèdia i scripts, per a millorar la velocitat de càrrega i l'experiència de l'usuari final.

[CDN](#)

Podem trobar les diferents versions de **jQuery CDN** a <https://releases.jquery.com/>

- Per afegir el codi **jQuery** a la nostra pàgina usem el següent script

```
<script  
  src="https://code.jquery.com/jquery-3.7.1.min.js">  
</script>
```

- **Esperar a que la pàgina estiga carregada:**

Després de carregar jQuery ens hem d'assegurar que la pàgina estiga completament carregada abans de començar a executar el codi. Ho farem envoltant el codi jQuery dins d'una funció que es crida quan el document està completament carregat, ho fem amb la funció. **\$(document).ready()**

```
$(document).ready(function() {  
  // Ací posem el nostre codi jQuery  
});
```

Esta funció s'executa quan tot el document **HTML** estiga carregat, i en eixe moment, s'executarà el codi **jQuery** contingut dins de la funció.

2.- Sintaxi jQuery

La sintaxi es prou simple i similar a la de JavaScript. Es basa en seleccionar els elements HTML sobre els que es vol actuar i efectuar alguna acció sobre ells.

La sintaxis bàsica es:

```
$(selectorCss).action()
```

Utilitzem el **\$** per accedir a **jQuery**, seguit d'un **selector CSS** per identificar els elements sobre els quals volem actuar, i finalment una **acció** que s'executarà sobre estos elements.

Per exemple:

```
$("#p").hide(); // Amaga tots els elements <p>.
$(".test").show(); // Mostra tots els elements amb class="test".
$("#test").fadeIn(); // Mostra poc a poc l'element amb id="test".
$(".:button").toggle(); // Per a tots els elements <button> o inputs amb type
button, alterna entre hide i show.
```

• Events

jQuery permet respondre a diferents esdeveniments com clics, desplaçaments del ratolí, etc. Utilitza la funció **on()** per afegir gestors d'esdeveniments.

Exemples de com manejar esdeveniments:

```
$(document).on("mouseenter", "p", function(event) {
    alert("Mouse Enter!");
});
```

En este exemple, establim un gestor d'esdeveniments per a l'esdeveniment " **mouseenter** " als elements **<p>**. Quan movem el ratolí sobre qualsevol element **<p>**, s'executa la funció especificada. En aquest cas, mostra una finestra d'alerta amb el missatge " **Mouse Enter!**".

```
$(document).on("click", "a.actiu", function(event) {
    event.preventDefault(); // Evita el comportament per defecte del clic.
    alert("Clic!");
});
```

En este exemple, establim un gestor d'esdeveniments per a l'esdeveniment " **click** " als elements **<a>** que tenen la classe " **actiu** ". Al fer clic en qualsevol d'estos enllaços, s'executa la funció especificada.

En aquest cas, s'evita el comportament per defecte del clic amb **event.preventDefault()** per evitar que l'enllaç es redirigeisca l'usuari a una nova pàgina, i després mostra una finestra d'alerta amb el missatge " **Clic!**".

• Obtenció i modificació de dades

jQuery proporciona mètodes per obtenir i modificar dades del contingut **HTML**. Alguns d'aquests mètodes són:

- **text()**: Retorna o determina el valor textual dels elements seleccionats.
- **html()**: Retorna o determina el valor dels elements seleccionats (incloent etiquetes html)
- **val()**: Retorna o determina el valor dels camps de formulari.
- **attr()**: Retorna o determina el valor d'un atribut determinat.

Exemple:

Exemple jQuery

En un lugar de **la Mancha** de cuyo nombre no *quiero acordarme*.

[Enllaç d'exemple](#) Mostrar Text Mostrar HTML Mostrar Valor Mostrar Atribut

HTML

```
<!DOCTYPE html>
<html lang="ca">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemple jQuery</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

<div class="container">
  <h1>Exemple jQuery</h1>
  <p id="paragraf">En un lugar de <strong>la Mancha</strong> de cuyo nombre no
<em>quiero acordarme</em>.</p>
  <input type="text" id="inputText" value="Valor de l'input">
  <a href="https://www.exemple.com" id="link">Enllaç d'exemple</a>
  <button id="btnText">Mostrar Text</button>
  <button id="btnHtml">Mostrar HTML</button>
  <button id="btnVal">Mostrar Valor</button>
  <button id="btnAttr">Mostrar Atribut</button>
  <div id="resultat" class="result"></div>
</div>

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script src="script.js"></script>

</body>
</html>
```

CSS

```
body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  background-color: #f2f2f2;
}

.container {
  width: 80%;
  margin: 20px auto;
  padding: 20px;
  background-color: #fff;
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

h1 {
  font-size: 24px;
  margin-bottom: 20px;
  color: #333;
}

p {
  margin-bottom: 15px;
  color: #666;
}

button {
  padding: 10px 20px;
  border: none;
  background-color: #007bff;
  color: #fff;
  font-size: 16px;
  cursor: pointer;
  border-radius: 5px;
  transition: background-color 0.3s ease;
}

button:hover {
  background-color: #0056b3;
}

.result {
  margin-top: 10px;
  padding: 10px;
  border: 1px solid #ccc;
  border-radius: 5px;
  background-color: #f9f9f9;
  color: #333;
}
```


JS

```
$(document).ready(function() {  
    // Mostrar text del paràgraf  
    $('#btnText').click(function() {  
        let text = $('#paragraf').text();  
        $('#resultat').text('Text del paràgraf: ' + text);  
    });  
  
    // Mostrar HTML del paràgraf  
    $('#btnHtml').click(function() {  
        let html = $('#paragraf').html();  
        $('#resultat').text('HTML del paràgraf: ' + html);  
    });  
  
    // Mostrar valor de l'input  
    $('#btnVal').click(function() {  
        let valor = $('#inputText').val();  
        $('#resultat').text('Valor de l\'input: ' + valor);  
    });  
  
    // Mostrar atribut href de l'enllaç  
    $('#btnAttr').click(function() {  
        let href = $('#link').attr('href');  
        $('#resultat').text('Atribut "href" de l\'enllaç: ' + href);  
    });  
});
```

Exemple jQuery

En un lugar de la Mancha de cuyo nombre no quiero acordarme.

[Enllaç d'exemple](#)Mostrar TextMostrar HTMLMostrar ValorMostrar Atribut

HTML del paràgraf: En un lugar de la Mancha de cuyo nombre no qiero acordarme.

Al clicar els botons ens eixirà respectivament:

Text del paràgraf: En un lugar de la Mancha de cuyo nombre no quiero acordarme.

HTML del paràgraf: En un lugar de la Mancha de cuyo nombre no qiero acordarme.

Valor de l'input: HOLA AAAA!!!!

Atribut "href" de l'enllaç: https://www.exemple.com

• Afegir Dades:

jQuery permet afegir elements del **HTML**. S'utilitzen els següents mètodes per a aquesta finalitat:

- **append()**: Agrega al final del elements seleccionats.

```
$("#p").append("<span>Adeu</span>"); // Afegeix un element <span> amb el text "Adeu" a cada element <p>
```

- **prepend()**: Agrega al principi del elements seleccionats.

```
$("#p").prepend("<span>Hola</span> "); // Afegeix un element <span> amb el text "Hola" a cada element <p>
```

- **before()**: Agrega abans del elements seleccionats.

```
$("#p").before("<b>Salutació: </b>"); // Afegeix l'etiqueta <b> amb el text Salutació abans de cada element <p>
```

- **after()**: Agrega després del elements seleccionats.

```
$("#p").after("<br>"); // Afegeix un salt de línia (<br>) després de tots els elements <p>
```

- **wrap()**: Agrega un element contenidor.

```
$("#p").wrap("<div class='container'></div>"); // Envolta tots els elements <p> amb un div amb la classe 'container'
```

Exemple: imaginem que tenim este HTML i este CSS:

<pre><!DOCTYPE html> <html lang="es"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial- scale=1.0"> <title>Exemple jQuery</title> <link rel="stylesheet" href="styles.css"> </head> <body> <p>Jaume</p> <p>Andrei</p> <script src="https://code.jquery.com/jquery- 3.6.0.min.js"></script> <script src="script.js"></script> </body> </html></pre>	<pre>/* Estils per a l'element */ span { color: blue; /* Canvia el color del text a blau */ font-weight: bold; /* Posa el text en egreta */ } /* Estils per a l'element .container */ .container { background-color: lightgray; /* Canvia el color de fons a gris clar */ padding: 10px; /* Afegeix un espai interior de 10px */ }</pre>
--	--

Mostraria:



Si apliquem les funcions JQuery:

```
$(document).ready(function() {  
    // Afegir contingut al final dels paràgrafs  
    $("p").append(" <span>Adeu</span>");  
  
    // Afegir contingut al principi dels paràgrafs  
    $("p").prepend("<span>Hola</span> ");  
  
    // Afegir contingut abans dels paràgrafs  
    $("p").before("<b>Salutació: </b>");  
  
    // Afegir contingut després dels paràgrafs  
    $("p").after("<br>");  
  
    // Envolta els paràgrafs amb un div amb la classe 'container'  
    $("p").wrap("<div class='container'></div>");  
});
```



• Esborrar Dades

jQuery conté una sèrie d'elements per esborrar dades o elements **HTML**, els més destacats són:

- **empty():** buida el contingut dels elements seleccionats, però manté els elements a la pàgina, es a dir, elimina tots els fills de l'element seleccionat, però l'element existirà al DOM.

```
$("#div").empty(); // Buida tots els elements div seleccionats
```

- **remove():** elimina completament els elements seleccionats del DOM, a diferència de empty(), esborra tant el contingut dels elements i els mateixos elements seleccionats de la pàgina.

```
$("#div").remove(); // Elimina tots els elements div seleccionats
```

- **unwrap():** elimina el contenidor directe dels elements seleccionats, i manté els fills intactes. Si els elements seleccionats estan dins d'un contenidor comú, este s'elimina i els fills es mantenen al DOM.

```
<div class="contenidor">  
  <p>Contingut aquí</p>  
</div>
```

```
$("#p").unwrap(); // Elimina el div.contenidor deixant només el paràgraf
```

- **removeAttr()**: elimina un atribut dels elements seleccionats. Per exemple, si volem eliminar l'atribut href d'un enllaç <a>:

```
$("a").removeAttr("href"); // Elimina l'atribut href de tots els enllaços <a>
```

Exemple

```
<!DOCTYPE html>
<html lang="ca">
<head>
  <meta charset="UTF-8">
  <title>Exemple de Manipulació de Dades amb jQuery</title>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
  <script src="script.js"></script>
</head>
<body>
<div>
  <h2>Contingut Div 1</h2>
  <p>Aquest és un paràgraf dins del div 1.</p>
</div>
<div>
  <h2>Contingut Div 2</h2>
  <p>Aquest és un paràgraf dins del div 2.</p>
  <a href="https://example.com">Enllaç al lloc web</a>
</div>
<button id="buida">Buida Contingut Div 1</button>
<button id="elimina">Elimina Div 2</button>
<button id="unwrap">Elimina Contenedor Directe</button>
<button id="eliminar-attr">Elimina Atribut Href</button>
</body>
</html>
```

```
$(document).ready(function(){
  // Buida el contingut del div 1 quan es fa clic al botó "Buida Contingut Div 1"
  $("#buida").click(function(){
    $("div:nth-child(1)").empty();
    // Div 1 estarà buit, però encara existirà al DOM
  });

  // Elimina el div 2 completament quan es fa clic al botó "Elimina Div 2"
  $("#elimina").click(function(){
    $("div:nth-child(2)").remove();
    // Div 2 i tot el seu contingut seran eliminats completament del DOM
  });

  // Elimina el contenidor directe dels elements seleccionats quan es fa clic al botó
  "Elimina Contenedor Directe"
  $("#unwrap").click(function(){
    $("p").unwrap();
    // Elimina el div.contenedor deixant només el paràgraf
  });

  // Elimina l'atribut href de tots els enllaços quan es fa clic al botó "Elimina Atribut
  Href"
  $("#eliminar-attr").click(function(){
    $("a").removeAttr("href");
    // S'eliminarà l'atribut href de tots els enllaços <a>
  });
});
```

• Manipulació de CSS:

Per manipular els estils **CSS** dels elements, **jQuery** ofereix mètodes per afegir, eliminar i alternar classes **CSS** als elements seleccionats. Estos mètodes són molt útils per canviar dinàmicament els estils dels elements **HTML** seleccionats durant l'execució de JavaScript amb **jQuery**.

- **addClass()**: Agrega una classe CSS a un element HTML seleccionat. Per exemple, si volem afegir una classe que canvie el color de fons d'un **div**:

```
$("#div").addClass("fons-roig");
```

- **removeClass()**: Mètode que s'utilitza per eliminar una classe CSS d'un element seleccionat. Per exemple, si volem eliminar una classe que canvie la mida de la font d'un paràgraf:

```
$("#p").removeClass("mida-gran");
```

- **toggleClass()**: S'utilitza per alternar l'aplicació d'una classe CSS en un element seleccionat. Per exemple, si volem alternar una classe que faci un text en negreta cada vegada que es fa clic:

```
$("#span").toggleClass("negreta");
```

Exemple

```
<!DOCTYPE html>
<html lang="ca">
<head>
  <meta charset="UTF-8">
  <title>Exemple de Manipulació de CSS amb jQuery</title>
  <style>
    .fons-roig {
      background-color: red;
    }
    .mida-gran {
      font-size: 20px;
    }
    .negreta {
      font-weight: bold;
    }
  </style>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
  <script src="script.js"></script>
</head>
<body>
<div>
  <p>Clica el botó per afegir la classe "fons-vermell" al div.</p>
  <button id="afegir">Afegir Classe</button>
</div>
<div>
  <p>Clica el botó per eliminar la classe "mida-gran" del paràgraf.</p>
  <button id="eliminar">Eliminar Classe</button>
</div>
<div>
  <p>Clica el text per alternar la classe "negreta".</p>
  <span id="alternar">Text per alternar</span>
</div>
```

```
</body>
</html>
```

```
$(document).ready(function(){

    // Afegir classe "fons-roig" quan es fa clic al botó "Afegir Classe"
    $("#afegir").click(function(){
        $("div").addClass("fons-roig");
    });

    // Eliminar classe "mida-gran" quan es fa clic al botó "Eliminar Classe"
    $("#eliminar").click(function(){
        $("p").removeClass("mida-gran");
    });

    // Alternar classe "negreta" quan es fa clic al text
    $("#alternar").click(function(){
        $(this).toggleClass("negreta");
    });
});
```

- **En Resum:**

- **Inicialització JQuery**
 - Inclou JQuery en una pàgina web mitjançant un CDN.
 - Utilitza **\$(document).ready()** per garantir que el codi JQuery s'executi després que la pàgina s'hagi carregat completament.
- **Sintaxi JQuery**
 - La sintaxi bàsica consisteix en seleccionar elements HTML i aplicar accions sobre ells.
 - Utilitza el selector **\$**, un selector CSS i una acció.
 - Exemples d'accions: **hide()**, **show()**, **fadeIn()**, **toggle()**.
- **Obtenció i Modificació de Dades**
 - Funcions per obtenir i modificar dades del contingut HTML, com **text()**, **html()**, **val()** i **attr()**.
 - Exemple d'ús de les funcions per mostrar text, HTML, valor de l'input i atribut d'un enllaç.
- **Afegir Dades**
 - Funcions per afegir contingut HTML als elements seleccionats: **append()**, **prepend()**, **before()**, **after()** i **wrap()**.
- **Esborrar Dades**
 - Mètodes per esborrar dades o elements HTML: **empty()**, **remove()**, **unwrap()** i **removeAttr()**.
- **Manipulació de CSS**
 - Utilització de JQuery per afegir, eliminar i alternar classes CSS als elements seleccionats: **addClass()**, **removeClass()** i **toggleClass()**.