**Pràctica 2**

**Ubicació del codi**

El codi Javascript és interpretat pel nostre navegador web, por tant, és necessari incloure’l en l’HTML. Quan s’escriu l’URL d’un lloc web en un navegador es fa una petició al servidor que allotja aquest lloc perquè retorni la informació que ha de pintar l’ordinador en el navegador web. Aquesta informació es retorna, normalment, en forma de codi HTML que el navegador interpreta y pinta. Això significa que si volem que el navegador interpreti un determinat codi Javascript haurà d’anar inclòs en l’HTML. Per aquest motiu, HTML té el Tag <script>.

**Diferents formes d’incloure en nostre codi Javascript**

Per explicar les diferents possibilitats a l’hora d’incloure el nostre codi Javascript, utilitzarem com a exemple una pàgina HTML que mostri el text “Fes click aquí” i que un cop es faci click, mostrarà l’alerta “Hola Món!”.

1. **Dins l’event d’una etiqueta (No recomanable)**

Abans era molt habitual trobar-nos amb codi javascript dins de les pròpies etiquetes HTML, formant part d’un atribut d’event, como per exemple l’event “onclick”:

<!DOCTYPE html>

<html>

<head>

<title>UF1: Pràctica 2</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body>

<h1>UF1: Pràctica 2</h1>

<span onclick="alert('Hola Món!');">Fes clilck aquí</span>

</body>

</html>

Quan es fa click en la frase, s’activa l’event “onclick” i el navegador executa el codi JavaScript inclòs en l’atribut de l’etiqueta span.

El problema que dóna aquest mètode és la impossibilitat de reutilitzar el codi JavaScript. Si vols tenir diferents elements que facin la mateixa funció, hauràs d’escriure el codi en cada event de cada etiqueta. A més, amb aquest mètode el codi no es visualitza amb claredat. En aquest exemple tenim poques etiquetes HTML, però en una web amb centenars d’etiquetes i múltiples pàgines HTML, mantenir el codi pot arribar a ser molt costós.

1. **Crides a funcions dins l’event d’una etiqueta (No recomanable)**

Amb la incorporació de la etiqueta <script> es va poder reutilitzar el codi utilitzant funcions:

<html>

<head>

<title>UF1: Pràctica 2</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<script>

function alerta(){

alert('Hola Món!');

}

</script>

</head>

<body>

<h1>UF1: Pràctica 2</h1>

<span onclick="alerta();">Fes clilck aquí</span>

</body>

</html>

El resultat és el mateix que en el primer exemple però d’aquesta forma aconseguim tenir tot el codi en la capçalera (excepte la crida a la funció ) de forma que és més senzill modificar-lo. A més, podem reutilitzar el codi, ja que podem fer crides a la funció des de diferents etiquetes HTML.

1. **Tot el codi dins de l’etiqueta <head>**

Encara que la segona opció suposa una clara millora, seguim tenint que posar codi dins dels events de les etiquetes HTML. El següent exemple hauria de solucionar el problema:

<html>

<head>

<title>UF1: Pràctica 2</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<script>

document.getElementsByTagName('span')[0].onclick = function(){

alert('Hola Món!');

};

</script>

</head>

<body>

<h1>UF1: Pràctica 2</h1>

<span>Fes clilck aquí</span>

</body>

</html>

Com podem observar, la nostra etiqueta <span> ja no conté codi JavaScript i s’ha substituït per un selector (getElementsByTagName()) que es troba a l’etiqueta <script> dins del <head>.

Per tant, dins del <body> només tenim codi HTML, que és el més idoni, però com podem comprovar, aquest mètode no funciona i apareix el següent error:

TypeError: document.getElementsByTagName(...)[0] is undefined

Per què no funciona? Per entendre-ho cal entendre com el navegador interpreta el codi:

Lo que realment veiem en un navegador és el cos del l’HTML, és a dir, el contingut del Tag <body>. El navegador va interpretant el codi de forma seqüencial, es a dir, primer processa el <head> i després el <body>. Per tant, en el moment de processar el <head>, encara no existeix l’etiqueta <span> i el navegador no troba resultats en el selector (getElementsByTagName()).

Una possible solució és posar el codi JavaScript al final del <body>, així ens assegurem que s’interpreta el codi un cop el navegador ha llegit totes les etiquetes HTML:

<html>

<head>

<title>UF1: Pràctica 2</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body>

<h1>UF1: Pràctica 2</h1>

<span>Fes clilck aquí</span>

<script>

document.getElementsByTagName('span')[0].onclick = function(){

alert('Hola Món!');

};

</script>

</body>

</html>

Aquesta solució funciona i ens trobem moltes webs que la utilitzen, però lo més correcte és que dins del <body> només hi hagi codi HTML. Disposem l’etiqueta <head> per incloure les etiquetes <meta>, les declaracions CSS, etc, i també és la més adient per afegir els nostres scripts. Per sort, tenim l’event “onload” que aplicat sobre l’objecte window soluciona el problema:

<html>

<head>

<title>UF1: Pràctica 2</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<script>

window.onload = function(){

document.getElementsByTagName('span')[0].onclick = function(){

alert('Hola Món!');

};

};

</script>

</head>

<body>

<h1>UF1: Pràctica 2</h1>

<span>Fes clilck aquí</span>

</body>

</html>

Amb aquest codi (window.onload), estem indicant al navegador, que quan el navegador acabi de carregar la pàgina web executi una funció JavaScript que inclou el nostre codi.

Amb aquest mètode ja tenim solucionats tots els problemes que es plantejaven al primer exemple, però si el codi es molt extens segueix sent complicat mantenir el codi dins l’HTML encara que només es trobi al <head>, per tant, la millor opció és afegir un fitxer extern que contingui tot el nostre codi JavaScript.

1. **Tot el codi dins d’un fitxer extern**

Aquest és el mètode més adient, i consisteix en afegir un fitxer extern en el <head>, amb extensió “js”, que contingui tot el codi JavaScript:

<html>

<head>

<title>UF1: Pràctica 2</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<script src="./codi.js"></script>

</head>

<body>

<h1>UF1: Pràctica 2</h1>

<span>Fes clilck aquí</span>

</body>

</html>

El fitxer que hem afegit per al nostre exemple és “codi.js”, i ha de contenir les mateixes sentències que abans:

**Codi.js**

window.onload = function(){

document.getElementsByTagName('span')[0].onclick = function(){

alert('Hola Món!');

};

};

Hem utilitzar l’atribut “source” (src) per indicar al navegador on es troba el fitxer. En aquest exemple hem utilitzat una ruta relativa al document (./codi.js). Si no s’especifica cap ruta i només s’indica el nom del fitxer, el navegador anirà a buscar el fitxer en el mateix lloc on es trobi el fitxer HTML.

És important afegir el nostre fitxer JavaScript extern al final del <head>, ja que el JavaScript pot interactuar amb el codi CSS i frameworks de JavaScript com ara JQUERY, i és convenient que el navegador hagi interpretat aquests elements abans del nostre codi, per tal de poder utilitzar-los correctament. Això ho anirem comprovant al llarg del curs.

**EXERCICI**

Reprodueix tots els mètodes d’incloure codi JavaScript que s’expliquen en aquest document.