

PLA 6: Manipulació del DOM

Mòdul → 2

Dedicació → 7 dies (35 hores)

Lliurament → 5 de juny de 2024

Descripció i objectius

En aquesta sisena PLA juguem amb tota una sèrie de mecanismes diferents mitjançant els quals podem modificar l'estructura d'una pàgina web des del nostre codi JavaScript. Gràcies al que es coneix com a **Document Object Model o DOM**, JavaScript pot accedir i alterar els elements d'un document HTML. Això ens permet als desenvolupadors crear interaccions dinàmiques i adaptatives amb els usuaris, convertint així les pàgines estàtiques en entorns interactius i vius.

L'objectiu de les PLA 5, 6 i 7 és afegir interactivitat en una pàgina web per tal de convertir-la en una autèntica aplicació web, gràcies a la versatilitat que ens ofereix el llenguatge de programació JavaScript.

Recordeu que, en paral·lel a la resolució de la PLA, convé que seguiu treballant en el vostre projecte final. Hores d'ara ja sabeu com estructurar una pàgina web amb HTML, com afegir-hi estils amb CSS (opcionalment amb Bootstrap o amb algun altre framework similar) i com afegir-hi funcionalitat amb codi JavaScript. Amb el que veurem en aquesta PLA també sabreu com modificar-ne dinàmicament el contingut.

Consideracions prèvies

Per resoldre aquesta PLA us resultarà d'especial interès la secció [Document Object Model \(DOM\)](#) de la MDN. El primer [article introductori](#) que enllacen és un bon punt de partida.

Però no us limiteu només a aquesta font, ja que a Internet trobareu una gran quantitat d'informació de gran interès sobre aquest tema. I, com sabeu, teniu també al vostre abast el llibre [JavaScript Notes for Professionals](#) on, fent servir l'índex o el cercador, trobareu part de la informació que necessiteu. Per aquesta PLA, però, aquest recurs no és tan útil com en altres PLA, tot i que sí que us pot anar bé per trobar com fer determinades coses en JavaScript, no necessàriament relacionades amb la manipulació del DOM.

D'altra banda, tot i que podeu avançar en l'estudi de JavaScript sense haver de dominar encara els conceptes de `callback function` i `higher order function`, convé que dediqueu algunes hores a llegir i mirar d'entendre aquests dos articles:

- [JavaScript Callback Function – Explained in Plain English](#) (també disponible en forma de vídeo, recomanat fer totes dues coses: llegir l'article i mirar el vídeo)
- [What are Higher-Order Functions in JavaScript?](#)

Pel que fa al DOM, tal com expliquen a [JavaScript HTML DOM](#), cal saber que ens permet fer un munt de coses amb un document web, com per exemple:

- change all the HTML elements in the page
- change all the HTML attributes in the page
- change all the CSS styles in the page
- remove existing HTML elements and attributes
- add new HTML elements and attributes
- react to all existing HTML events in the page
- create new HTML events in the page

I tot això sense que l'usuari hagi de refrescar la pàgina ni haver de fer cap nova crida al servidor. Us adoneu de l'enorme quantitat de possibilitats que això posa a les nostres mans?

Vídeos recomanats

Al canal de YouTube de [Bluuweb](#) trobareu la playlist [Curso de JavaScript Moderno](#) amb una sèrie de vídeos que mostren un exemple força complet de manipulació del DOM. Recomanat mirar aquests sis, que sumen un parell d'hores:

- #01 ¿Qué es el DOM?
- #02 Leer y Modificar HTML
- #03 createElement
- #04 Fragment + createElement (no más reflow)
- #05 Template vs innerHTML vs createElement
- #06 addEventListener (click), Event delegation y stopPropagation

Si aneu justos de temps prescindiu, de moment, dels vídeos #04 i #05, però torneu-hi més endavant. Al mateix canal podeu trobar més vídeos amb contingut interessant, per bé que l'estil de l'autor és força peculiar, especialment quan fa sessions en directe.

Important

Aquesta és la primera PLA amb un nivell de dificultat elevat en aquest curs. Aneu poc a poc, pas a pas, i no us desespereu si de bon inici no sabeu gairebé ni per on començar. S'espera, més que mai,

participació al fòrum. **No per compartir codi amb solucions**, és clar, però sí per compartir idees («jo començaria fent això», «jo crec que tal element el podem recuperar tenint en compte que compleix tal condició», etc).

No cal resoldre tots els exercicis perfectament. Expliqueu, en cada cas, què heu fet, fins on heu arribat, quina era la manera com intentàveu resoldre el problema, etc.

Únicament com a referència, l'arxiu `scripts6-solution.js` amb totes les solucions ocupa una mica més de 300 línies de codi i s'han definit una dotzena de funcions. Els vostres números poden variar.

Tasques a resoldre

Totes les tasques d'aquesta PLA cal resoldre-les a l'arxiu `scripts6.js` proporcionat, cada una a l'espai on s'indica al mateix arxiu. Tot i que en algun moment us pot semblar que necessiteu també modificar l'arxiu `index6.html` o que us seria molt útil poder modificar també l'arxiu d'estils `styles6.css`, heu d'assumir que, a causa dels requeriments d'un client en aquest cas imaginari, aquests dos arxius no es poden alterar de cap manera. Només podeu modificar l'arxiu `scripts6.js`.

De nou, **no modifiqueu res** als arxius `index6.html` i `styles6.css`. Només heu de modificar l'arxiu `scripts6.js`, que és el que heu d'entregar al final. A l'hora de corregir la vostra PLA es farà emprant els arxius `index6.html` i `styles6.css` originals proporcionats.

En [aquest vídeo](#) trobareu una petita demo de com ha de funcionar el document web `index6.html` una vegada solucionades totes les tasques proposades.

1. Píndoles de teoria

Responen les següents preguntes a la part superior del document `scripts6.js` en els blocs de comentari multilínia indicats. Assegureu-vos que les línies tenen una amplada màxima de 100 caràcters, per tal de millorar-ne la llegibilitat. Els còpyastes, citats o no, no puntuen.

1. Com definiríes què es exactament i per a què serveix el DOM?
2. La línia on enllacem els scripts al document HTML incorpora l'atribut `defer`. Per a què serveix? Quina diferència hi fa posar-lo o no? Si no ho teniu clar, resoleu primer els exercicis i després proveu de treure l'atribut i experimenteu amb el resultat.
3. En els mètodes `getElementBy...` i `getElementsBy...` hi ha una `s` que els diferencia. Quines implicacions té això pel que fa al valor que retornen aquests mètodes? Quins són els altres dos mètodes que serveixen gairebé pel mateix?

2. Menú d'opcions segons rol seleccionat

A la pàgina web trobareu un menú dividit en dues files. A la primera podem triar entre dos rols diferents: `Usuari` i `Administrador`. A la segona hi ha tot d'opcions disponibles segons el rol escollit. Cal que escriviu el codi JavaScript necessari per implementar aquest menú.

Heu d'aconseguir que si es tria el rol `Usuari` només és mostrin les opcions en verd de la segona fila, mentre que si es tria el rol `Administrador` totes les opcions de la segona fila han d'estar disponibles. A més, el rol seleccionat ha de quedar remarcant afegint-hi els estils de la classe `role-selected` (ja existent a la fulla d'estils externa).

Per fer-ho:

1. Penseu bé amb quins selectors podeu recuperar tant el llistat de rols com el llistat d'opcions.
2. Necessitareu afegir un event als dos elements que fan de botons per escollir el rol. Penseu quin event és el més adequat i quina funció caldrà executar quan es produeixi.

3. Classificació de futbol

A la pàgina web trobareu una taula buida. Si mireu el document HTML veureu que té definit un bloc `<tbody>`, però sense cap element dins. Aquesta taula té l'identificador `classification`.

Al codi JavaScript trobareu definida una variable `teams` amb quatre seleccions nacionals de futbol, cadascuna amb el número de partits que ha guanyat, empatat i perdut. Cal que escriviu el codi necessari per poblar aquesta taula amb el contingut de la variable `teams`. Per fer-ho:

1. Aproveiteu que aquesta taula té l'identificador `classification`.
2. Creeu una nova fila `<tr>` per cada equip amb els elements `<td>` corresponents.
3. Afegiu aquestes files a l'element `<tbody>` de la taula.
4. A la primera fila sencera afegiu-hi la classe CSS `classification-first` (ja existent a la fulla d'estils externa).

Els equips han d'aparèixer, idealment i com és habitual en aquests casos, ordenats per punts, de més a menys. El sistema de puntuació és l'habitual: les victòries sumen tres punts, els empats un punt i les derrotes cap. Deixeu, però, aquesta part pel final si veieu que us dona problemes, ja que implementar-la no és trivial.

4. Informe de clients

A la pàgina web trobareu una taula amb un llistat de clients, cadascun amb el seu saldo i els dies que falten fins el dia en què s'espera que es produeixi el pagament. Cal que feu tres coses:

1. Quan es carrega la pàgina s'han de veure en color vermell els valors negatius de la tercera columna. Es corresponen amb els deutes que ja s'haurien d'haver pagat i ens interessa que quedin remarcats. Feu servir la classe `unpaid` (ja existent a la fulla d'estils externa).
2. Quan es carrega la pàgina, la fila dels totals ha de mostrar la suma de tots els imports de la columna en comptes del 0.00 € que hi apareix per defecte.
3. L'opció `Remove unpaid` no fa res, però al document HTML té associada una funció a través de l'atribut `onclick`. Implementeu aquesta funció perquè quan l'usuari premi aquesta opció desapareguin de la taula totes les files corresponents a clients morosos (els que ja haurien d'haver pagat). No heu d'amagar aquestes files, les heu d'eliminar del tot. A més, el saldo total s'haurà d'actualitzar per reflectir el total de les files que romanguin.

Lleig, oi, aquest ús de l'atribut `onclick`? Sí, però és un dels mecanismes més senzills per afegir interactivitat a una pàgina web i en algun projecte antic us el podríeu trobar. Quan estigui a les vostres mans, opteu sempre per assignar els events des de JavaScript.

Per fer-ho:

1. Aproveiteu que aquesta taula té l'identificador `customers`.
2. Investigueu la funció `parseFloat()`.
3. Tingueu present que per eliminar un element del document HTML podem fer servir diversos mètodes, però cal que realment s'elimini, no volem deixar-lo en mode no visible. Ha de desaparèixer del DOM.

5. Valoració amb estrelletes

Al document HTML trobareu un contenidor `<div>` que inclou uns altres cinc contenidors `<div>` amb diferents classes CSS assignades.

Cal que escriviu el codi JavaScript necessari per implementar un sistema de valoració clàssic en el qual l'usuari pot fer clic sobre qualsevol de les estrelletes per marcar una puntuació. Aquesta puntuació ha de quedar recollida a la variable `rating` ja definida a l'arxiu JavaScript, que prendrà valors entre 1 i 5. Per fer-ho:

1. Itereu els possibles conjunts d'elements que obtingueu amb un bucle de tipus `for...of`.
2. Com a la tasca 2, necessitareu assignar algun event a algun element del DOM.
3. Des de l'event que acabeu d'afegir al punt anterior, poseu o treieu classes CSS segons correspongui a cada un dels contenidors `<div>`.

6. Barra de breakout

Al document HTML trobareu un contenidor `<div>` amb la classe `bar-container` i, dins, un altre contenidor `<div>` amb la classe `bar`.

Cal que escriviu el codi JavaScript necessari per implementar una petita part del típic joc [Breakout](#) on hem d'anar bellugant la barra per evitar que una piloteta caigui al buit. Aquesta barra ha de poder moure's de manera horitzontal però no vertical.

Vosaltres heu d'aconseguir que, prement les tecles fletxa esquerra i fletxa dreta, la barra es bellugui. Per fer-ho:

1. Assigneu un event a l'element `<body>`.
2. Tingueu present com funciona el mecanisme de cascada de CSS. Segurament necessitareu sobreescriure una de les propietats definides a la classe `bar`. Podeu fer-ho afegint des de JavaScript un atribut `style` (CSS inline) que, per tant, passarà per sobre de qualsevol estil CSS extern.

7. Piloteta de breakout

Al document HTML trobareu un contenidor `<div>` amb la classe `ball-container` i, dins, un altre contenidor `<div>` amb una altra classe `ball` que **no podeu fer servir en cap cas al vostre codi**. Per tant, haureu de trobar una altra manera de guanyar accés a l'element que us interessa.

Cal que escriviu el codi JavaScript necessari per implementar una altra petita part del joc Breakout, aquella en què la piloteta va rebotant per les parets.

Vosaltres heu d'aconseguir que, de manera automàtica quan es carrega la pàgina, la piloteta es bellugui i vagi rebotant. Per fer-ho:

1. No podeu fer servir la classe `ball` de la piloteta per accedir-hi. Heu de fer servir el mètode `querySelector` per accedir a la `<div>` que hi ha dins del contenidor amb etiqueta `ball-container`.
2. Investigueu el mètode `setInterval()` per executar una funció de manera repetida. En aquest cas, opteu per un número de milisegons entre 20 i 50.
3. Tingueu present com funciona el mecanisme de cascada de CSS. Segurament necessitareu sobreescriure algunes de les propietats definides a la classe `ball`. Podeu fer-ho afegint des de JavaScript un atribut `style` (CSS inline) que, per tant, passarà per sobre de qualsevol estil CSS extern.

Super challenge opcional

Teniu tant la barra com la piloteta del Breakout funcionant? Doncs no esteu gaire lluny de poder ajuntar les dues parts i crear un petit joc. Potser podeu afegir-hi un marcador i algunes millores visuals, però la part principal ja la teniu. Us animeu a acabar-lo? 😊

Entrega

Un arxiu `c2409_PLA6_Cognom_Nom.js` que inclogui totes les vostres respostes. Afegiu els blocs de comentaris que considereu oportuns quan vulgueu explicar algun concepte o fer qualsevol tipus d'aclariment.

Notes finals

1. Com a totes les PLA, serà sensacional si feu servir un **repositori privat a GitHub** amb el nom `cifo-frontend-2024` i hi pugeu el contingut d'aquesta PLA (directori `pla6`). Idealment, l'historial de commits hauria de mostrar el procés evolutiu de resolució de la PLA, tasca a tasca, apartat a apartat.
2. **Mai copieu / plagieu** cap resposta ni fragment de codi sense afegir-ne les referències corresponents. En programació és habitual —i correcte— copiar petits blocs de codi de diferents llocs. Però si una part rellevant del que entregueu no és d'elaboració pròpia cal que ho mencioneu. No fer-ho suposarà una qualificació de zero punts.
3. Recordeu que les guies d'estil són molt importants i convé que les tingueu presents des del principi, per tal d'aconseguir escriure un codi font net, elegant i menys propens a errors. **En el cas de JavaScript, al tractar-se del principal llenguatge de programació que treballem en aquest curs, això pren una especial importància.**
4. Per bé que eines com GitHub Copilot o ChatGPT són d'innegable ajuda a l'hora de programar, és molt important que entengueu perfectament tot el codi que entregueu i que el sapigueu explicar quan us ho demanin. En cas contrari us serà difícil superar entrevistes de selecció laboral per accedir a posicions de programació.