



PAC Mòdul 2 (JavaScript) - Pokedex

Inici:

27/06/22

00:00h

Hora Central
Europea d'Estiu
(CEST)

Lliurament:

18/09/22

24:00h

Hora Central
Europea d'Estiu
(CEST)

Descripció i enunciat

JavaScript

Crear una Pokedex i un combat de cartes

Aquesta serà l'entrega avaluable del Mòdul 2. Tot i així, es recomana anar fent la resta d'exercicis opcionals proposats per anar-se familiaritzant amb JavaScript i veure exemples concrets que es podran aplicar en aquesta entrega.

Introducció

L'objectiu d'aquesta pràctica es construir una versió senzilla d'una Pokedex i un joc de combat de cartes.

A la pàgina oficial de Pokemon es pot veure un exemple de Pokedex, tot i que l'objectiu és construir una versió mínima d'això. Una Pokedex és un llistat de Pokemons des d'on es pot consultar informació detallada de cadascun d'ells.

Bàsicament, deixant de banda la temàtica de Pokemon, l'exercici tracta de consultar informació que ens arriba d'un recurs extern i un cop rebuda, maquetarla desde la nostra aplicació i afegir-li una mica de funcionalitat extra. Tasques de l'exercici

Crear una web amb HTML, CSS i JS que tingui 2 pàgines. La pàgina principal (index.html) és on hi haurà una llista de Pokemons amb informació bàsica (nom, imatge, valor d'atac i valor de defensa), on al seleccionar-ne un, es recarregarà la pàgina per mostrar més informació de Pokemon seleccionat (nom, imatge, valor d'atac, valor de defensa i tipus). S'haurà de crear un sistema de navegació per tornara la pàgina anterior.

La segona pàgina serà combat.html. Aquesta pàgina haurà de mostrar un llistat de Pokemons com si fóssin cartes i estiguéssin girades boca per avall. De manera que no sabem quin Pokemon hi ha a cada carta. S'hauran de seleccionar 2 cartes fent clic, aquestes s'hauran de girar per revelar el Pokemon de cada carta i s'hauran de comparar els valors d'atac i defensa per decidir quin dels 2 guanya el combat.

S'han de crear 2 pàgines, la pàgina inicial (index.html) i una pàgina de combat (combat.html). Cadascuna d'elles ha de contenir una barra de menú on es pugui navegar d'una pàgina a una altra, i un selector (radio button) que permeti escollir si es vol veure la web amb una paleta de colors clars o foscos. El funcionament s'especifica més avall.

Pàgina inicial (index.html)

- Consulta aleatòria a la API

Al carregar la pàgina inicial, s'ha de fer una consulta a la API de Pokemons (PokeAPI) i recuperar les dades de 10 Pokemons de forma aleatoria. Cadascun dels Pokemons s'ha de mostrar maquetat com una targeta o carta i ha de contindre la següent informació: nom, atac, defensa i una imatge.

Aquesta informació es pot extreure de la API consultant l'ID del Pokemon. Per exemple, si es vol consultar el Pokemon amb ID = 1 s'ha de fer una consulta a la URL: <https://pokeapi.co/api/v2/pokemon/1/>

Per tant, s'haurà d'executar la crida 10 vegades, consultant cada cop la URL <https://pokeapi.co/api/v2/pokemon/{ID}> generant un número aleatori (random) a cada crida per obtenir un nou ID. Idealment s'hauria de controlar que no es generin elements duplicats.

La petició a la PokeAPI es pot fer utilitzant Fetch API

```
fetch('http://example.com/movies.json')
```

```
.then(response => response.json())
```

```
.then(data => console.log(data));
```

Es pot obtenir la informació navegant per l'objecte fins a les rutes (veure imatge adjunta: exemple-rutes-api)

Nom = Objecte.name

Imatge = Objecte.sprites.front_default

Atac = Objecte.stats[1].base_stat

Defensa = Objecte.stats[2].base_stat

- Informació ampliada de cada Pokemon

Les targetes de la pàgina principal també han de contenir un botó o un enllaç que permeti accedir a una pàgina amb informació més detallada del Pokemon. Aquesta informació es carregarà sobre la mateixa pàgina index.html, indicant a la mateixa URL un paràmetre amb l'ID del Pokemon que es vol consultar. Per exemple: <index.html?pokeID=5>

Al carregar la pàgina index.html s'ha de consultar si la URL conté el paràmetre pokeID. Si el conté, només s'ha de mostrar la informació del Pokemon en qüestió i un botó per tornar enrere. En cas contrari, ha de mostrar el llistat inicial de Pokemons.

La informació complerta dels Pokemon que s'ha de mostrar en aquesta pàgina ha de ser:

Nom = Objecte.name

Imatge frontal = Objecte.sprites.front_defau

Imatge posterior = Objecte.sprites.back_default

Atac = Objecte.stats[1].base_stat

Defensa = Objecte.stats[2].base_stat

Tipus (poden ser varis) = Objecte.types[x].type.name : s'hauran de mostrar tots els tipus

- Filtre de cerca

La pàgina inicial també ha de tenir un filtre textual de Pokemons. Ha de ser un input de tipus "search" en el que, a mesura que es van escrivint lletres, només s'han de mostrar aquells Pokemon que el seu nom contingui el text escrit. Aquesta comprovació s'ha de fer per cada lletra que s'escriui al filtre (event input).

Per exemple, si tenim les cartes: Charmander, Bulbasaur, Charizard i Blastoise, si s'escriu al filtre "cha", només han de ser visibles les cartes Charmander i Charizard.

Pàgina combat (combat.html)

La pàgina de combat, al carregar-se ha de fer una consulta i retornar 10 Pokemons aleatoris. Aquests s'han de mostrar en format targeta com a l'apartat anterior, però inicialment tota la informació ha d'estar oculta. Les targetes han d'estar "girades" com si fóssin cartes boca avall i només es mostra la part posterior de la targeta. Per tant, inicialment l'usuari només veurà 10 targetes totes igual.

L'usuari haurà de seleccionar dos targetes fent clic sobre les que vulgui. Les targetes seleccionades hauran de mostrar la informació de la targeta: nom, imatge, atac i defensa.

Un cop seleccionades les dues cartes, aquestes entren en combat seguint la següent lògica:

- La primera carta seleccionada és la carta atacant. S'utilitzarà el seu valor "atac".

- La segona carta seleccionada és la carta que defensa. S'utilitzarà el seu valor "defensa".

- Si el valor d'atac de la primera carta és superior al valor de defensa de la segona, la primera carta guanya el combat. S'ha de mostrar un missatge "{nom del Pokemon de la primera carta} ataca i guanya a {nom del Pokemon de la segona carta}".

- Si, en canvi, el valor d'atac de la primera carta és igual o inferior al valor de defensa de la segona carta, guanya la segona carta. El missatge ha de ser: "{nom del Pokemon de la primera carta} ataca i perd contra {nom de Pokemon de la segona carta}".

A continuació s'ha d'activar un botó (que inicialment ha d'estar desactivat) per tornar la pàgina i les cartes al seu estat inicial i poder tornar a efectuar un nou combat. Es pot recarregar la pàgina o tornar a l'estat inicial tots els elements afectats.

Opcionalment es pot fer ús de data-attributes per emmagatzemar els valors d'atac, defensa i nom si resulta més còmode.

Per mostrar la informació de la targeta, es pot fer de cop o fent ús d'aluna animació amb CSS. Es pot utilitzar l'efecte anomenat flip card, que simula una targeta en 3D donant la volta.

Barra de menú transversal: selecció de tema clar/fosc

A la barra superior de menú, a banda dels enllaços per poder navegar de la pàgina d'inici a la de combat i viceversa, també hi ha d'haver un selector de tipus radio button amb 2 opcions: clar i fosc. Aquest selector, per defecte ha d'estar seleccionat amb l'opció de tema clar.

Els estils inicials de la pàgina i les targetes han de ser amb un color de fons clar amb una tipografia de color fosc. Si l'usuari selecciona l'opció de tema fosc, els estils de la pàgina han de canviar i els colors de fons i de les targetes han de ser foscos i la tipografia ha de ser clara (sempre a la inversa del color de fons per tal que hi ha un bon contrast i la legibilitat sigui accessible).

La selecció que fagi l'usuari s'ha de guardar al navegador com a Local Storage, de manera que quan l'usuari torni a carregar la web, el sistema recordi les seves preferències. Si havia seleccionat el mode fosc, la web s'ha de carregar directament en mode fosc. I el mateix pel mode clar. Si l'usuari no ha seleccionat cap preferència, s'haurà de carregar en mode clar.

Opcionalment es pot configurar perquè la web es carregui segons estigui configurat el navegador o el sistema operatiu de l'usuari. En aquest cas, s'haurà d'afegir una tercera opció al radio button que sigui "sistema" i que agafi els estils segons l'usuari tingui configurat el seu equip (clar o fosc).

A tenir en compte

No cal estar familiaritzat amb el món de Pokemon. S'ha decidit utilitzar aquesta API perquè és molt completa i és gratuïta. És una API àmpliament utilitzada en exemples de programació de treball amb API, i és fàcil trobar exemples.

Tot i que l'entrega està calendaritzada en varies setmanes, cal tenir en compte que durant l'agost es proposa fer una pausa i reprendre el curs al setembre.

Objectius

El principal objectiu d'aquesta pràctica és treballar amb JavaScript i veure com es pot interactuar amb el DOM, creant i eliminant elements HTML i modificant i interactuant amb CSS.

Es veurà com treballar amb una API externa (<https://pokeapi.co/>), realitzant operacions de consulta, i s'aprendrà a interpretar les dades rebudes en format JSON.

Finalment es treballarà amb Local Storage, per aprendre a utilitzar el navegador per guardar informació sense haver de dependre d'una Base de dades.

Recursos recomanats

Consultar apartat "Continguts i recursos"

 [exemple-rutes-api.png](#)

Continguts i recursos

Generació de nombres aleatoris

https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Math/random

Fetch API

https://developer.mozilla.org/es/docs/Web/API/Fetch_API/Using_Fetch

Consultar exemples de JavaScript al Github del curs

<https://github.com/Aleix-Marti/curs-frontend/tree/main/JS>

Exemple Pokedex

<https://www.pokemon.com/us/pokedex>

PokeAPI

<https://pokeapi.co>

Exemple CSS Flip Card

<https://codepen.io/Aleix/pen/ozWBoG>

Com crear un tema clar / fosc amb variables CSS

<https://wordpress.tv/2020/01/22/aleix-marti-carmona-creando-un-dark-theme-con-variables-css-unete-al-lado-oscuro>

7 Mètodes d'array en un minut

https://www.youtube.com/watch?v=96VjrPib6MA&ab_channel=midudev

Interacció DOM + JS

Veure el la gravació de la sessió del dilluns 18 de juliol.

Entra al campus virtual i accedeix a l'aula per visualitzar el video titulat js-scope-operadors-condicionals.mp4

/webapps/aulaca/classroom/

LoadUOCMediaWidget.action;jsessionid=1D1E2E233EB569F312D4F0E250F47D64.campins.uoc.es-aulaca?

s=4a3e682082f336d22119b95c86f797585114446de068a16f7777a5d6b772b89b1a6bb93f8d62ba34f902b883e8ad6c15cbc5bd8fd

Entra al campus virtual i accedeix a l'aula per visualitzar el video titulat js-funcions.mp4

/webapps/aulaca/classroom/

LoadUOCMediaWidget.action;jsessionid=1D1E2E233EB569F312D4F0E250F47D64.campins.uoc.es-aulaca?

s=4a3e682082f336d22119b95c86f797585114446de068a16f7777a5d6b772b89b1a6bb93f8d62ba34f902b883e8ad6c15cbc5bd8fd

Entra al campus virtual i accedeix a l'aula per visualitzar el video titulat bucles.mp4

/webapps/aulaca/classroom/

LoadUOCMediaWidget.action;jsessionid=1D1E2E233EB569F312D4F0E250F47D64.campins.uoc.es-aulaca?

s=4a3e682082f336d22119b95c86f797585114446de068a16f7777a5d6b772b89b1a6bb93f8d62ba34f902b883e8ad6c15cbc5bd8fd

Entra al campus virtual i accedeix a l'aula per visualitzar el video titulat arrays.mp4

/webapps/aulaca/classroom/

LoadUOCMediaWidget.action;jsessionid=1D1E2E233EB569F312D4F0E250F47D64.campins.uoc.es-aulaca?

s=4a3e682082f336d22119b95c86f797585114446de068a16f7777a5d6b772b89b1a6bb93f8d62ba34f902b883e8ad6c15cbc5bd8fd

Entra al campus virtual i accedeix a l'aula per visualitzar el video titulat es5-es6.mp4

/webapps/aulaca/classroom/

LoadUOCMediaWidget.action;jsessionid=1D1E2E233EB569F312D4F0E250F47D64.campins.uoc.es-aulaca?

s=4a3e682082f336d22119b95c86f797585114446de068a16f7777a5d6b772b89b1a6bb93f8d62ba34f902b883e8ad6c15cbc5bd8fd

Entra al campus virtual i accedeix a l'aula per visualitzar el video titulat objectes.mp4

/webapps/aulaca/classroom/

LoadUOCMediaWidget.action;jsessionid=1D1E2E233EB569F312D4F0E250F47D64.campins.uoc.es-aulaca?

s=4a3e682082f336d22119b95c86f797585114446de068a16f7777a5d6b772b89b1a6bb93f8d62ba34f902b883e8ad6c15cbc5bd8fd

Entra al campus virtual i accedeix a l'aula per visualitzar el video titulat arrays-metodes.mp4

/webapps/aulaca/classroom/

LoadUOCMediaWidget.action;jsessionid=1D1E2E233EB569F312D4F0E250F47D64.campins.uoc.es-aulaca?

s=4a3e682082f336d22119b95c86f797585114446de068a16f7777a5d6b772b89b1a6bb93f8d62ba34f902b883e8ad6c15cbc5bd8fd

 JavaScript UOC.pdf

 Formularis.pdf