



Informática 2

Ejercicio JavaScript

Ejercicio JavaScript

Ejercicio JavaScript – Web de Star Wars

En este ejercicio vamos a implementar una página Web en la que obtendremos información de la conocida saga “Star Wars”

Para ello, vamos a obtener la información detallada usando el API público y gratuito:

<https://swapi.dev/>

Si te fijas en la documentación¹ del API en cuestión, podrás obtener información de las naves, planetas, personajes, ...

Este API nos responde un documento JSON para cada petición que le hacemos.

Por ejemplo, si pedimos el primer planeta:

<https://swapi.dev/api/planets/1/>

Nos responde

```
{  "climate": "Arid",      "diameter": "10465",    "gravity": "1 standard",
  "name": "Tatooine",     "orbital_period": "304",    "population": "200000",
  "residents": [         "https://swapi.dev/api/people/1/",
  "https://swapi.dev/api/people/2/",    ...    ],    "rotation_period": "23",
  "surface_water": "1",      "terrain": "Dessert",      "url":
  "https://swapi.dev/api/planets/1/"}
```

Es recomendable mirar que nos responde para cada petición haciendo la llamada con el navegador directamente o bien con la herramienta de consola “cURL”

A partir de los datos que nos devuelve este API (que no requiere ningún tipo de autenticación) el alumno deberá generar las siguientes páginas:

¹ <https://swapi.dev/documentation>

- index.html -> Mostrará un listado con todas las películas disponibles y un enlace a la página “película” correspondiente
- película.html -> Mostrará los datos de la película indicada a través del parámetro “id” en la URL (ej.: película.html?id=2 mostrará los datos de la película con el ID 2 del API)
- personaje.html -> Mostrará la información de un personaje en base al id indicado en la dirección (semejante a película.html)
- planeta.html -> Mostrará la información de un planeta en base al id indicado en la dirección (semejante a película.html)
- nave.html -> Mostrará la información de una nave en base al id indicado en la dirección (semejante a película.html)
- vehiculo.html -> Mostrará la información de un vehículo en base al id indicado en la dirección (semejante a película.html)
- especie.html -> Mostrará la información de una especie en base al id indicado en la dirección (semejante a película.html)

Cómo hacerlo

Para realizar la práctica correctamente se recomienda seguir los siguientes pasos:

Generar un fichero JS para interactuar con el API (que será usado por todas las páginas) y otro por página (que se encargará de modificar el contenido del árbol DOM).

- Implementar una función en JavaScript "**obtenerDatosSWAPI**" que obtenga el recurso solicitado del API anterior y nos devuelva el documento en formato JSON. Para el ejemplo anterior, la llamaríamos como:

`obtenerDatosSWAPI('planets/1')`

- Se aconseja, por comodidad, crear funciones para obtener cada tipo de recurso:
 - obtenerPeliculas()
 - obtenerPersonaje(id)
 - obtenerNave(id)
 - ...
- Si se fija en el API, si no se indica id numérico, devuelve una lista (array) con todos los recursos.

- También es importante destacar que, para cada recurso enlazado desde otro, este API nos lo informa con la URL completa, no obstante, por comodidad para usar la variable id en la URL, vamos a usar el número final.
Se aporta el código de una función para obtener este identificador numérico a partir del dato aportado por el API (una URL). Haga uso de esta función:

```
function obtenerIdURLRecursoSWAPI(url) {  
    return Number(url.match(/([0-9]*)\V?$/)[1])  
}
```

Para recuperar los parámetros de la URL usaremos la clase **URLSearchParams**, con un código semejante al siguiente:

```
let p = new URLSearchParams(document.location.search)  
let id = p.get('id')
```

Las páginas generadas anteriormente deberán cargar todos los datos en paralelo e ir pintando en cada sitio según se reciben. Por ejemplo, para una película, tras recibir la película, solicitar cada personaje, vehículo, ... indicados para recuperar el nombre y mostrar el enlace.



Se muestra un ejemplo, el diseño es una sugerencia, pero se deja libertad al alumno para hacer el diseño que más le guste.

Se valorará el uso de Flex

