

Destructuring & Spread Operator

Práctica Integradora

Objetivo

Queremos seguir explorando el potencial de Javascript y poniendo en práctica los distintos recursos que venimos usando en clases anteriores. Practicaremos también los métodos de Destructuring y Spread Operator que conocimos en la clase sincrónica.

¡ Éxitos !



Micro desafío 1:

Instrucciones

El Tech Leader presenta al equipo un nuevo proyecto de **compra y venta de figuras coleccionables**. Una de las primeras acciones a tomar es unificar en un único **array de objetos literales** todas las listas de las marcas y características de las figuras que hoy se encuentran por separado en archivos de texto (**Formato JSON**). Para lograr este objetivo nos encarga las siguientes tareas:

1. Crear una carpeta para el proyecto (**coleccionables**).
2. Crear una subcarpeta (**datos**) y dentro de ella tres archivos (**figuras1.json - figuras2.json - figuras3.json**). Cada uno de los archivos debe contener un array de objetos literales con todas las figuras que se tienen en stock. Por cada figura hay que detallar esta información:
 - i. Para la marca **Hot Toys**:
 1. **marca** (Hot Toys)
 2. **nombre** (Spider-Man, Hellboy, Tony Stark, Black Panther, Batman, Captain America, Superman)
 3. **precio** (Indicar el precio)
 4. **stock** (Indicar la cantidad)
 - ii. Para la marca **Bandai**:
 1. **marca** (Bandai)
 2. **nombre** (Gamora, Hulk, Thor, Vegeta, Sailor Pluto, Sailor Saturn, Wonder Woman)
 3. **precio** (Indicar el precio)
 4. **stock** (Indicar la cantidad)
 - iii. Para la marca **Star Wars**:
 1. **marca** (Star Wars)
 2. **nombre** (Luke Skywalker, Scout Trooper, Boba Fett, Dark Trooper, Kessel Run, Jedi, Ahsoka Tano)
 3. **precio** (Indicar el precio)
 4. **stock** (Indicar la cantidad)

3. Crear en la carpeta raíz del proyecto un archivo (**collectibles.js**). Este archivo será un **módulo propio**, en el que debes construir una función (**importar**). Esta recibirá como parámetro el nombre de la marca de las figuras coleccionables. Tendrá la responsabilidad de leer el [archivo en formato JSON](#) y devolver un [array de objetos](#) de cada una de las marcas que reciba.

No olvides que este archivo es un **módulo propio** y por tal motivo una vez creado el mismo... ¿te acuerdas cuál debe ser la última línea del archivo?

Micro desafío 2:

Instrucciones

1. Crear un nuevo archivo (**app.js**). En este, deberás importar el **módulo** creado (**collectibles.js**).
2. Crear una variable por cada una de las marcas, invocando a la función (**importar**) y pasarle como argumento cada una de las marcas de las figuras coleccionables (**hotToys - bandai - starWars**).
3. Crear un único array (**unifiedCollectibles**) con todas las listas de las marcas de las figuras coleccionables. Usa [Spread Operator](#).
4. Crear un **objeto literal** (**collectibles**) que tenga como primer atributo (**figuras**) y contenga la lista de todas las figuras importadas.
5. Dentro del **objeto literal**, crea las funcionalidades solicitadas por el cliente: a.
Desarrollar la funcionalidad (**listFigures**). Tendrá la responsabilidad de [mostrar al usuario](#) todos los datos de las diferentes figuras almacenadas. Puedes utilizar los métodos [forEach\(\)](#) o [for...of\(\)](#).
b. Crear una funcionalidad (**figuresByBrand**) que reciba por parámetro la **marca de la figura**. Tendrá la responsabilidad de retornar todas aquellas figuras que correspondan a la marca recibida por parámetro. Deberíamos utilizar el método [filter](#).
c. Comprobar las funcionalidades usando el [console.log\(\)](#) e invocando a cada una de ellas pasando los referidos parámetros.



Desafío extra

Instrucciones

Si llegaste hasta acá, vienes trabajando muy bien poniendo en práctica los diferentes contenidos vistos hasta el momento. **¡Felicitaciones!**

Para que no te quedes con las ganas y puedas seguir practicando, te proponemos que desarrolles las siguientes funciones. No olvides verificar los valores resultantes utilizando la instrucción `console.log()`.

1. En la carpeta raíz del proyecto, crear un nuevo archivo (**extra.js**).
2. Crear un **array** de números y asignarle un total de diez (10) valores numéricos. 3. Crear variables que contengan los valores numéricos dispuestos en el array en la posición (**0, 2 y 4**) y con el resto de los valores numéricos crear un nuevo array. Utiliza los recursos aprendidos en esta clase: [Destructuring](#) y [Spread operator](#).
4. Crear un **objeto literal** para almacenar los datos de nuestra mascota, considerando los atributos:
 - a. nombre
 - b. tipo de mascota (Perro - Gato)
 - c. color
 - d. raza

Haciendo uso de la **desestructuración**, crea una variable por cada una de las claves del objeto literal y al final mostrar al usuario:

```
Hola, les presento a mi mascota su nombre es: Bony, es un hermoso Perro, de  
color: Dorado y su raza es: Golden retriever.
```