

D.O.M.

Creando los archivos **index.html** y **app.js**:

1. Dentro del archivo **index.html** generar:
 - a. Un `<h1>` con cualquier contenido.
 - b. 10 elementos `<p>` con contenido aleatorio.
 - c. Un lista `` con 7 elementos `` con contenido de relleno.
2. En nuestro archivo **app.js**:
 - a. Capturar a todos los elementos ``. Luego recorrerlos e imprimir en consola cada uno de ellos. **WARNING:** NO usar `querySelectorAll`.
 - b. Capturar el padre de los elementos anteriores (``) y mostrarlo en consola.
 - c. Capturar al `<h1>` y cambiar su contenido interno por el siguiente:
`Javascript is the best my friends!`.
 - d. Capturar a todos los `<p>`, recorrerlos y cambiar el contenido interno de los elementos impares por el siguiente: `Párrafo número N: enlace generado desde JS`. Donde N será el número de párrafo correspondiente.

Objeto Window

1. Crear un archivo **gastos.html** y vincularlo con archivo **gastos.js**.
Importante, tener precaución de generar todo nuestro código JS dentro del `window.onload`.
2. Usando los métodos **prompt()**, **confirm()** y **alert()** e incluso cosas que hemos visto en clases pasadas. Vamos a generar un reporte de gastos diarios de una familia. Dicho reporte funciona de la siguiente manera: Pide la cantidad de integrantes de la familia, para cada uno de ellos pide nombre y luego la cifra que ésta persona gastó. Al final veremos cada nombre con su correspondiente cifra, la persona que más gastó y la persona que menos gastó. Y el total de gastos de toda la familia.
 - a. Lo primero que haremos es preguntarle al visitante recién ingresa al documento si quiere iniciar. Si la respuesta es negativa, deberá mostrarse una alerta que diga *"Gracias por haber venido"* y luego será direccionado al sitio web de Netflix.
 - b. Ahora, si la respuesta es positiva, vamos a iniciar nuestro proceso, lo primero que haremos es pedirle al visitante la cantidad de integrantes de la familia. Validar que el dato ingresado sea un número y que no sea inferior a 3, de NO ser un número deberá alertar que el valor necesario es un número y volverá a preguntar por la cantidad de integrantes. Para este proceso, puede ser de utilidad la siguientes función nativa de JS `isNaN()`. Puedes buscar qué hace la misma.
 - c. Una vez con la cantidad de integrantes listos, vamos a pedir para cada uno nombre y gastos del día. Tener en cuenta de validar que:
 - i. El nombre no puede estar vacío.
 - ii. Los gastos no puede ser un texto ni estar vacío.
 En cualquiera de esos casos, alertar del error y volver a pedir el dato. Al final se deberá generar un Array de Objetos Literales cada uno con la propiedad **nombre** y **valor**. Ejemplo:


```
integrantes[
  {nombre: "Ada", gastos: 300},
  {nombre: "Tim", gastos: 570},
  {nombre: "Vincent", gastos: 80},
]
```
 - d. Después de esto haremos mostraremos en consola dicho array.
 - e. Para finalizar tendremos que:
 - i. Imprimir en consola cuál fue el integrante que más gastó.
 - ii. Imprimir en consola cuál fue el integrante que menos gastó.

- iii. Imprimir en consola los gastos de toda la familia.
- iv. Imprimir en consola el promedio de gastos del día.

Ejercicio Integrador

Vamos a volver a nuestros archivos **gastos.html** y **gastos.js**, y ahora:

Teniendo en cuenta toda la funcionalidad existente, ya NO vamos a mostrar los resultados por consola si no que mostraremos los resultados dentro del contenido del archivo html. **ATENCIÓN:** todo lo siguiente deberá ser generado desde javascript, nuestro archivo html no debe tener NADA dentro del `<body>` excepto la vinculación del `<script>`.

1. Generar un título con el texto *"Reporte de gastos familiares"*.
2. Generar una lista dentro de la cual se muestre:
 - a. El nombre del integrante que más gastó junto con su gasto.
 - b. El nombre del integrante que menos gastó junto con su gasto.
 - c. Los gastos de toda la familia.
 - d. El promedio de gastos del día.Todos los elementos de lista, deberán tener el atributo `title` con el mismo texto que muestra cada uno.
3. Generar un `<button>` con el texto "¿Nos pasamos del presupuesto?" el cual, al ser clicado, muestre un texto llamativo diciendo si el presupuesto diario fue superado, teniendo en cuenta que para toda la familia el presupuesto máximo diario es de **\$1200**.
4. Generar otro `<button>` que al ser clicado, asigne si no existe y elimine si existe, la clase **dark-theme** al `<body>`. Crear un **css** que defina todo el set de estilos para dicha clase.