

Laboratorio Práctico: Trabajando con Iteración de Arrays para la Lista de Menú de Restaurante



Tiempo estimado necesario: 20 minutos

Lo que aprenderás

En este laboratorio, explorarás cómo JavaScript genera dinámicamente secciones del menú del restaurante para desayuno, plato principal y postre utilizando los métodos de arrays de JavaScript. Aprenderás a usar `map` para llenar el menú de desayuno, `forEach` para mostrar los elementos del menú del plato principal, y un bucle `for` tradicional para renderizar los elementos del postre. Además, entenderás cómo el contenido HTML puede actualizarse dinámicamente, mostrando el poder de JavaScript en la modificación de elementos de la página web basándose en arrays predefinidos.

Objetivos de aprendizaje

Después de completar este laboratorio, podrás:

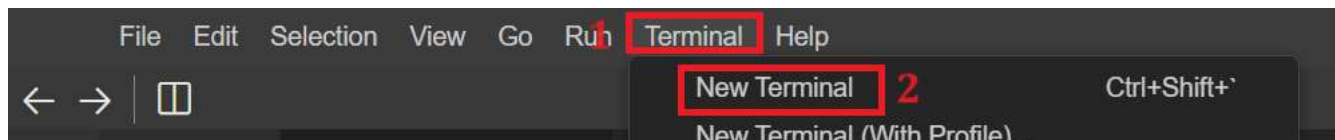
- **Entender la manipulación del DOM:** Adquirir competencia en el acceso a elementos HTML y en la actualización dinámica de su contenido utilizando JavaScript. Enfocarse en el método `getElementById` para alterar de manera precisa elementos específicos dentro de una página web.
- **Métodos de iteración de arrays:** Diferenciar entre varios métodos de iteración de arrays como `map`, `forEach` y bucles `for`. Observar sus funcionalidades distintas al iterar a través de arrays y generar contenido HTML estructurado.
- **Creación de contenido web dinámico:** Aprender el proceso de generar y renderizar contenido dinámicamente en una página web utilizando JavaScript para crear menús estructurados basados en arrays predefinidos, enfatizando la naturaleza dinámica del desarrollo web.
- **Organización de secciones del menú:** Comprender el concepto de organizar el contenido de la página web en secciones (por ejemplo, desayuno, plato principal, postre) al llenar elementos HTML con datos de arrays de JavaScript correspondientes, facilitando diseños de páginas web estructurados y legibles.
- **Aplicación de métodos de JavaScript:** Aplicar métodos de JavaScript (`map`, `forEach`, `getElementById`, `innerHTML`) para manipular archivos HTML de manera efectiva, mostrando el uso práctico de estos métodos en la generación y actualización dinámica del contenido de la página web basado en datos predefinidos.

Requisitos previos

- Conocimientos básicos de HTML.
- Navegador web con consola (Chrome DevTools, Consola de Firefox, y más).

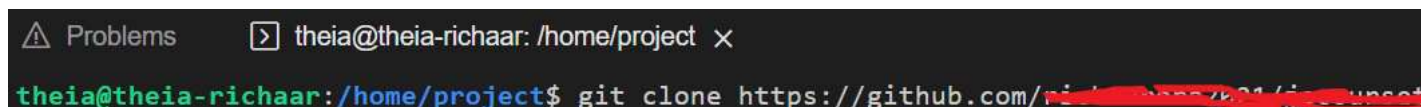
Paso 1: Configuración del entorno

1. Primero, necesitas clonar tu repositorio principal en el **Entorno de Skills Network** que has creado en el primer laboratorio y donde has subido todos los archivos y carpetas de tus laboratorios anteriores. Sigue los pasos indicados:
 - Para esto, haz clic en la terminal en el panel derecho y luego selecciona **Nueva Terminal**.



- Realiza el comando `git clone` escribiendo el siguiente comando en la terminal.
`git clone <github-repository-url>`

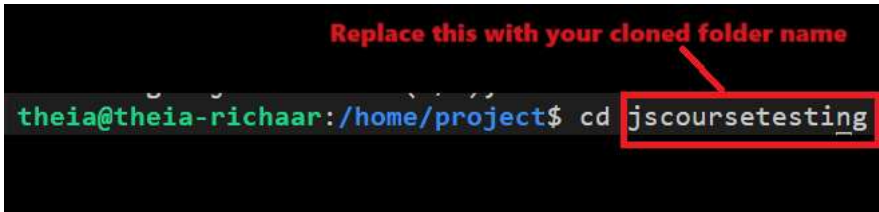
Nota: Pon tu propio enlace de repositorio de GitHub en lugar de `<github-repository-url>`.



- El paso anterior clonará la carpeta de tu repositorio de GitHub en la carpeta del proyecto en el explorador. También verás múltiples carpetas dentro de la carpeta clonada.

- Ahora necesitas navegar dentro de la carpeta clonada. Para esto, escribe el siguiente comando en la terminal:

```
cd <repository-folder-name>
```



Nota: Escribe el nombre de tu carpeta clonada en lugar de <repository-folder-name>. Realiza `git clone` si has cerrado sesión en el **Entorno de Skills Network** y no puedes ver ningún archivo o carpeta después de iniciar sesión nuevamente.

2. Ahora, selecciona la carpeta **cloned Folder Name**, haz clic derecho sobre ella y selecciona **Nueva Carpeta**. Ingresa el nombre de la carpeta como **restaurantMenu**. Esto creará la carpeta para ti. Luego selecciona la carpeta **restaurantMenu**, haz clic derecho y selecciona **Nuevo Archivo**. Ingresa el nombre del archivo como **restaurant_menu.html** y haz clic en Aceptar. Esto creará tu archivo HTML.
3. Ahora selecciona la carpeta **restaurantMenu** nuevamente, haz clic derecho y selecciona **Nuevo Archivo**. Ingresa el nombre del archivo como **restaurant_menu.js** y haz clic en Aceptar. Esto creará tu archivo JavaScript.
4. Crea la estructura básica del archivo HTML añadiendo el contenido proporcionado a continuación.

```
<!DOCTYPE html>
<html>
<head>
  <title>Menú del Restaurante</title>
</head>
<body>
  <h1>Interacción del Menú del Restaurante</h1>
  <h2>Menú de Desayuno</h2>
  <div id="breakfastTotalItems"></div>
  <div id="breakfastMenuItems"></div>
  <h2>Menú de Plato Principal</h2>
  <div id="maincourseTotalItems"></div>
  <div id="maincourseMenuItems"></div>
  <h2>Menú de Postres</h2>
  <div id="dessertTotalItems"></div>
  <div id="dessertMenuItems"></div>
  <script src="./restaurant_menu.js"></script>
</body>
</html>
```

- Este código HTML proporciona la base estructural para mostrar un menú de restaurante en tres secciones: desayuno, plato principal y postres. Contiene elementos como `<h1>` para el título general, etiquetas `<h2>` para cada sección del menú, y elementos `<div>` con IDs únicos (`breakfastTotalItems`, `breakfastMenuItems`, `maincourseTotalItems`, `maincourseMenuItems`, `dessertTotalItems`, `dessertMenuItems`) destinados a mostrar dinámicamente el contenido del menú.
- Sin embargo, por ahora, no hay contenido de menú real siendo poblado o mostrado dinámicamente. Se necesita código JavaScript para generar e insertar los elementos del menú en los respectivos elementos `<div>` identificados por sus IDs únicos.

Nota: Cuando hayas pegado el código, guarda tu archivo.

Paso 2: Definir variables y método de matriz map

1. En **restaurant_menu.js**, inicializa tres matrices llamadas **breakfastMenu**, **mainCourseMenu** y **dessertMenu**. Inicializa estas matrices con los elementos del menú como se indica en el código a continuación. Incluye este código en el archivo **restaurant_menu.js**.

```
const breakfastMenu = ['Pancakes', 'Eggs Benedict', 'Oatmeal', 'Frittata'];
const mainCourseMenu = ['Steak', 'Pasta', 'Burger', 'Salmon'];
const dessertMenu = ['Cake', 'Ice Cream', 'Pudding', 'Fruit Salad'];
```

2. Ahora, utilizaremos un bucle for y dos métodos de arreglo: el método **map** y el método **forEach** para recorrer estos arreglos y mostrar los elementos del menú en la página HTML.

3. En primer lugar, vamos a iterar a través del arreglo **breakfastMenu** utilizando el método **map**. Incluye el siguiente código en el archivo **restaurant_menu.js**.

```
const breakfastMenuItemsHTML = breakfastMenu.map((item, index) => `<p>Item ${index + 1}: ${item}</p>`).join('');
document.getElementById('breakfastMenuItems').innerHTML = breakfastMenuItemsHTML;
```

- El código proporcionado convierte los elementos del arreglo del menú de desayuno en cadenas HTML utilizando `map()` y una función de flecha para estructurar el formato HTML de cada elemento.
- Posteriormente, se ha utilizado un método de concatenación de cadenas para unir las cadenas HTML generadas en una sola cadena cohesiva utilizando `join('')` para prepararla para la inserción.
- Finalmente, este HTML concatenado se actualiza dinámicamente dentro del elemento HTML específico identificado por el ID `breakfastMenuItems` con la cadena HTML concatenada, poblando dinámicamente la página web con los elementos del menú de desayuno en párrafos formateados con etiquetas `<p>`.

Paso 3: Definición del método de array `forEach` y la iteración del bucle `for`

1. Vamos a recorrer el array **mainCourseMenu** utilizando el método de array **forEach**. Para esto, incluye el siguiente código en el archivo **restaurant_menu.js** después del código JavaScript anterior.

```
let mainCourseItem = '';
mainCourseMenu.forEach((item, index) => {
  mainCourseItem += `<p>Item ${index + 1}: ${item}</p>`;
});
document.getElementById('maincourseMenuItems').innerHTML = mainCourseItem;
```

- La variable `mainCourseItem` se inicializa como una cadena vacía (`''`). Esta variable se utilizará para acumular cadenas HTML generadas para cada elemento del menú principal.
 - El método `forEach` recorre cada elemento en el array `mainCourseMenu`. Para cada elemento en el array, se ejecuta la función de flecha `(item, index) => {...}`. Dentro de la función de flecha, se compone una cadena HTML para cada elemento del menú, incorporando el contenido del elemento y su índice correspondiente.
 - Luego, el contenido HTML se actualiza dinámicamente estableciendo la propiedad `innerHTML` del elemento HTML con el ID `maincourseMenuItems` al string acumulado `mainCourseItem`, que insertará los elementos del menú principal en el elemento específico dentro de la página web.
2. Por último, iteraremos el array **dessertMenu** utilizando la iteración del **bucle for**. Para esto, incluye el siguiente código en el archivo **restaurant_menu.js** después del código JavaScript anterior.

```
let dessertItem = '';
for (let i = 0; i < dessertMenu.length; i++) {
  dessertItem += `<p>Item ${i + 1}: ${dessertMenu[i]}</p>`;
}
document.getElementById('dessertMenuItems').innerHTML = dessertItem;
```


- La variable `dessertItem` se inicializa como una cadena vacía (`''`). Esta variable sirve como contenedor para acumular cadenas HTML generadas para cada elemento del menú de postres.
- El bucle `for` itera a través del array `dessertMenu`, comenzando desde el índice 0 y continuando hasta alcanzar la longitud del array. Para cada iteración, se crea una cadena HTML utilizando el elemento actual en el array `dessertMenu`, incluyendo el contenido del elemento y su índice (`i + 1`).
- El contenido HTML generado, compuesto por párrafos individuales `<p>` que contienen los elementos del menú de postres con sus respectivos índices, se asigna al `innerHTML` del elemento HTML identificado por el ID `dessertMenuItems`, que insertará los elementos del menú de postres en el elemento

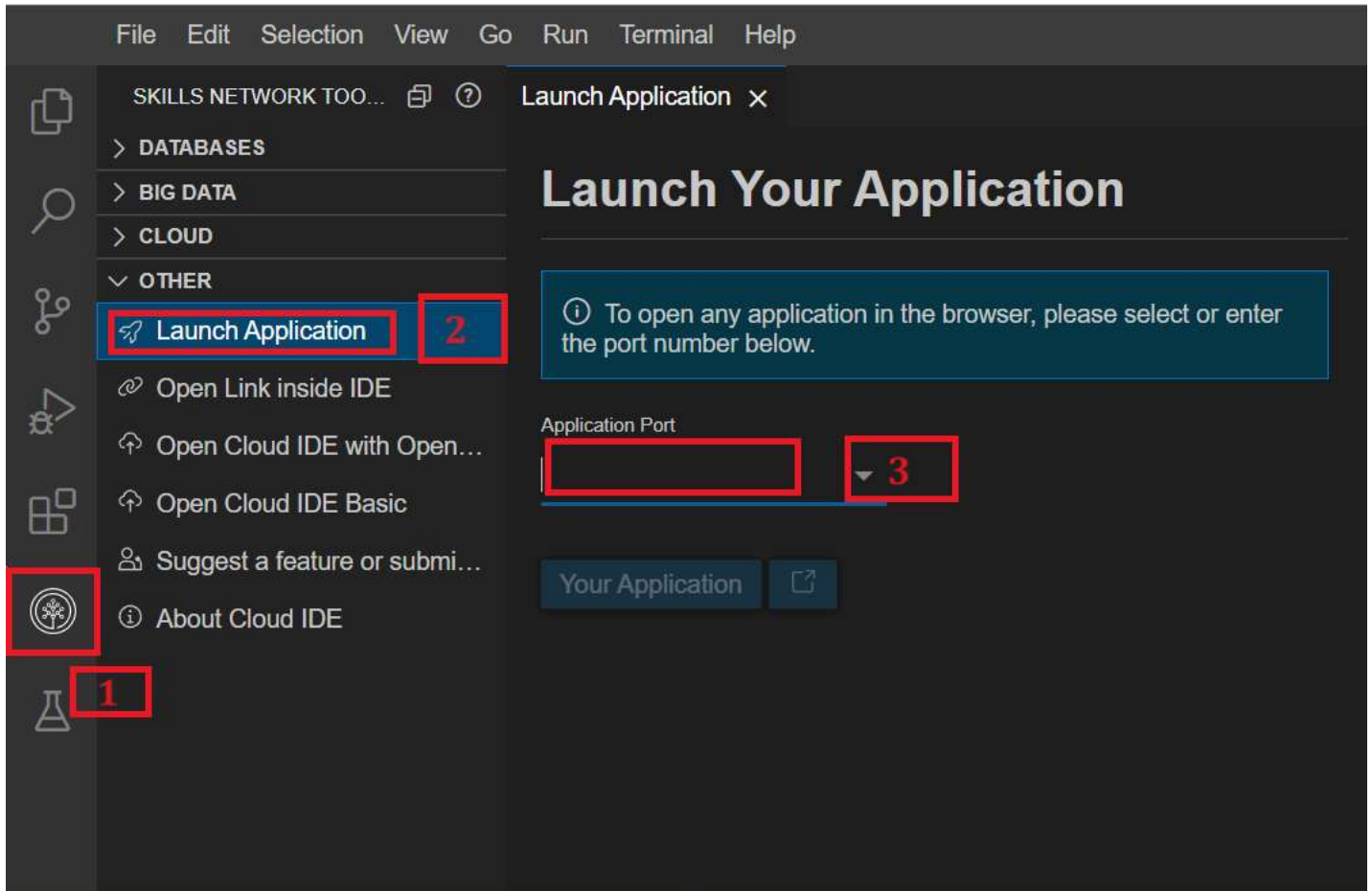
específico dentro de la página web.

Paso 4: Verificar la salida

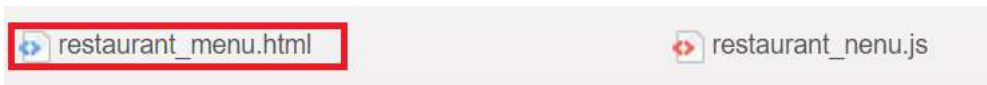
1. Para ver cómo se ve tu página HTML, haz clic derecho en el archivo **restaurant_menu.html** después de seleccionarlo, y luego selecciona “Abrir con Live Server.”
2. El servidor debería iniciarse en el puerto 5500, indicado por una notificación en la parte inferior derecha.



3. Haz clic en el botón de Skills Network a la izquierda (consulta el número 1), se abrirá la “Caja de herramientas de Skills Network”. Luego haz clic en **Lanzar aplicación** (consulta el número 2). Desde allí, ingresa el número de puerto como 5500 en el número 3 y haz clic en este botón .



4. Se abrirá tu navegador predeterminado donde verás el nombre de la carpeta **cloned-folder-name**. Haz clic en esa carpeta **cloned-folder-name** y luego haz clic en la carpeta **restaurantMenu**. Verás archivos relacionados con esta carpeta donde volverás a hacer clic en el archivo **restaurant_menu.html** como se muestra a continuación.



5. Se abrirá la página HTML y verás los elementos del menú para **Desayuno**, **Plato Principal** y **Postre**.

Restaurant Menu Iteration

Breakfast Menu

Item 1: Pancakes

Item 2: Eggs Benedict

Paso 5: Ejecutar comandos de Git

1. Realiza `git add` para agregar los archivos y carpetas más recientes escribiendo el comando dado en la terminal en el entorno de git.

```
git add --a
```

Asegúrate de que la terminal tenga la ruta como sigue:



2. Luego realiza `git commit` en la terminal. Al realizar `git commit`, la terminal puede mostrar un mensaje para configurar tu `git config --global` para `user.name` y `user.email`. Si es así, entonces necesitas ejecutar el comando `git config` también para `user.name` y `user.email` como se indica.

```
git config --global user.email "you@example.com"
```

```
git config --global user.name "Your Name"
```

Nota: Reemplaza los datos dentro de las comillas con tus propios detalles.

Luego realiza el comando de commit como se indica:

```
git commit -m "message"
```

3. Luego realiza `git push` simplemente escribiendo el comando dado en la terminal.

```
git push origin
```

- Después del comando push, el sistema te pedirá que ingreses tu nombre de usuario y contraseña. Ingresa el nombre de usuario de tu cuenta de GitHub y la contraseña que creaste en el primer laboratorio. Después de ingresar las credenciales, todas tus carpetas y archivos más recientes se enviarán a tu repositorio de GitHub.

Nota: Después de pegar el código, guarda tu archivo. Puedes usar cualquier método de salida para esto. Si editas tu código, simplemente actualiza tu navegador ejecutándose a través del número de puerto 5500. De esta manera, no hay necesidad de lanzar la aplicación una y otra vez.

Tarea de práctica

1. Puedes incluir precios para todos los artículos en `breakfastMenu`, `mainCourseMenu` y `dessertMenu`.

2. Para esto, puedes incluir los precios de la siguiente manera:

```
const breakfastMenu = ['Pancakes- $12', 'Eggs Benedict -$22.99', 'Oatmeal -$21.99', 'Frittata -$15'];
```

3. Luego, accede a ellos de manera similar y verás la salida de los elementos del menú junto con los precios también.

Restaurant Menu Iteration

Breakfast Menu

Item 1: Pancakes- \$12

Item 2: Eggs Benedict -\$22.99

Item 3: Oatmeal -\$21.99

Item 4: Frittata -\$15

Resumen

- Estructura HTML:** Define una estructura HTML para un menú de restaurante, categorizando los elementos en secciones de Desayuno, Plato Principal y Postre utilizando encabezados `<h2>` apropiados y elementos `<div>` con IDs únicos para la visualización del contenido.
- Contenido dinámico en JavaScript:** Utiliza JavaScript para poblar dinámicamente los elementos del menú dentro de sus respectivas secciones.
 - Menú de desayuno: Utiliza el método `map` para generar cadenas HTML para los elementos de desayuno e inserta estas en el elemento `<div>` designado.
 - Menú de plato principal: Utiliza `forEach` para construir cadenas HTML para los elementos del plato principal, actualizando el `<div>` relacionado con el contenido generado.
 - Menú de postres: Utiliza un bucle `for` para crear cadenas HTML para los elementos de postre e inserta estas en el elemento `<div>` designado para los elementos de postre.
- Manejo de datos:** Almacena los datos de los elementos del menú en arreglos de JavaScript separados (`breakfastMenu`, `mainCourseMenu`, `dessertMenu`) para facilitar la creación de contenido dinámico sin codificar directamente en la estructura HTML.
- Renderizado dinámico:** Ilustra cómo JavaScript interactúa con HTML, generando y actualizando dinámicamente el contenido dentro de la página web, demostrando diferentes métodos de iteración para presentar los elementos del menú bajo categorías específicas.