

Ejercicios sobre Objetos

1. Creación de un Objeto Básico:

- Crea un objeto llamado **libro** con propiedades como **título**, **autor** y **añoDePublicacion**. Imprime cada propiedad en la consola.

2. Anidación de Objetos:

- Crea un objeto llamado **estudiante** con propiedades **nombre**, **edad** y **direccion**. **direccion** debe ser otro objeto con propiedades **calle**, **ciudad** y **país**. Imprime la dirección completa del estudiante.

3. Métodos en Objetos:

- Añade un método llamado **descripcion** al objeto **libro** que devuelva una cadena describiendo el título y el autor del libro. Invoca este método e imprime el resultado.

4. Iteración sobre Propiedades de un Objeto:

- Crea un objeto **producto** con propiedades como **nombre**, **precio** y **disponible**. Usa un bucle **for...in** para imprimir todas las propiedades y sus valores.

5. Actualización de Propiedades:

- Modifica el valor de la propiedad **precio** del objeto **producto** y luego imprime el objeto completo para verificar el cambio.

6. Comprobación de Propiedades:

- Crea una función llamada **tienePropiedad** que tome un objeto y una cadena como parámetros, y devuelva **true** si el objeto tiene una propiedad con ese nombre, o **false** en caso contrario.

7. Eliminación de Propiedades:

- Elimina la propiedad **disponible** del objeto **producto**. Imprime el objeto antes y después de eliminar la propiedad.

8. Combinar Objetos:

- Usa **Object.assign** para combinar dos objetos, **persona1** y **persona2**, en un nuevo objeto. Imprime el resultado.

9. Copiar Objetos:

- Crea una copia profunda del objeto **estudiante** utilizando el método **JSON.parse** y **JSON.stringify**. Modifica la copia y verifica que el objeto original no haya sido alterado.

10. Métodos Getters y Setters:

Licenciatura en Sistemas de Información

Programación Avanzada - 2024

Introducción a JavaScript

- Añade un getter y un setter al objeto `libro` para la propiedad `añoDePublicacion`. Usa el setter para actualizar el año de publicación y luego usa el getter para leerlo.

02

Ejercicios sobre Funciones

1. Función Suma:

- Crea una función llamada `sumar` que tome dos números como parámetros y devuelva su suma. Prueba la función con diferentes números.

2. Función que Multiplica:

- Crea una función llamada `multiplicar` que tome dos números como parámetros y devuelva su producto. Prueba la función con diferentes valores.

3. Función con Parámetro por Defecto:

- Crea una función llamada `saludar` que tome un parámetro `nombre` con un valor por defecto de "Invitado". La función debe devolver "Hola, [nombre]".

4. Función que Devuelve un Objeto:

- Crea una función llamada `crearPersona` que tome `nombre` y `edad` como parámetros y devuelva un objeto con esas propiedades.
- #### 5. Función que Modifica un Objeto:

- Crea una función llamada `actualizarEdad` que tome un objeto `persona` y un número `nuevaEdad`, y actualice la propiedad `edad` del objeto.

6. Función Recursiva:

- Crea una función recursiva llamada `factorial` que calcule el factorial de un número dado.

7. Función con Función Interna:

- Crea una función llamada `despedir` que contenga una función interna `adios`. `despedir` debe devolver el resultado de llamar a `adios`.

8. Función que Usa Otra Función:

- Crea una función llamada `procesarArray` que tome un array y una función como parámetros, y aplique la función a cada elemento del array. Usa una función como parámetro para multiplicar cada número por 2.

Licenciatura en Sistemas de Información

Programación Avanzada - 2024

Introducción a JavaScript

9. Función que Devuelve Otra Función:

- Crea una función llamada `crearMultiplicador` que tome un número `x` y devuelva una nueva función que multiplique cualquier número por `x`.

10. Función Anónima:

- Crea una función anónima que sume dos números y la asigna a una variable llamada `sumarAnonima`. Luego, invoca la función a través de esa variable.

03

Ejercicios sobre Funciones (Consumo de Datos, Mapeo de Información, Autenticación de Usuarios)

1. Consumo de Datos desde una API:

- Crea una función llamada `obtenerUsuarios` que haga una petición HTTP a la API <https://jsonplaceholder.typicode.com/users> usando `fetch`.

Luego, imprime en la consola la lista de usuarios obtenida.

2. Procesamiento de Datos de una API:

- Usando la función `obtenerUsuarios`, crea otra función llamada `imprimirNombresDeUsuarios` que filtre y muestre solo los nombres de los usuarios.

3. Autenticación Simulada:

- Crea una función llamada `autenticarUsuario` que tome un objeto `credenciales` con `usuario` y `contraseña`, y verifique si coinciden con un usuario predefinido. La función debe devolver `true` si la autenticación es exitosa y `false` en caso contrario.

4. Transformación de Datos:

- Crea una función llamada `mapearUsuarios` que tome un array de usuarios obtenidos de la API y devuelva un nuevo array con solo las propiedades `nombre` y `email` de cada usuario.

5. Validación de Formularios:

- Crea una función llamada `validarFormulario` que tome un objeto con los campos `nombre`, `email` y `password`. La función debe devolver `true`

Mg. Ernesto Ledesma - FCyT - UADER



Licenciatura en Sistemas de Información

Programación Avanzada - 2024

Introducción a JavaScript

si todos los campos están presentes y no están vacíos, y `false` en caso contrario.

6. Paginación de Datos:

- Crea una función llamada `obtenerPagina` que tome un array de datos y un número de página. La función debe devolver los elementos correspondientes a esa página, asumiendo que cada página tiene 5 elementos.

7. Envío de Datos a una API:

- Crea una función llamada `enviarDatos` que tome un objeto `data` y haga una petición POST a la API <https://jsonplaceholder.typicode.com/posts>. La función debe imprimir la respuesta de la API.

8. Búsqueda de Usuarios:

- Crea una función llamada `buscarUsuarioPorEmail` que tome un array de usuarios y un email como parámetros, y devuelva el usuario que coincida con el email proporcionado. Usa el método `find` para implementarlo.

9. Generación de Token de Autenticación:

- Crea una función llamada `generarToken` que tome un objeto `usuario` y devuelva un token JWT simulado como una cadena. Puedes usar una función como `btoa` (Base64) para simular la generación del token.

10. Actualización de Información del Usuario:

- Crea una función llamada `actualizarUsuario` que tome un objeto `usuario` y una lista de cambios a aplicar. La función debe retornar el usuario con las propiedades actualizadas.

04

Ejercicios sobre Operaciones con Arrays

1. Agregar y Eliminar Elementos:

- Crea un array `frutas` con los valores `["manzana", "banana", "pera"]`. Usa

`push` para agregar una fruta al final, y `pop` para eliminar la última fruta.

2. Array Bidimensional:

Mg. Ernesto Ledesma - FCyT - UADER



Licenciatura en Sistemas de Información

Programación Avanzada - 2024

Introducción a JavaScript

- Crea un array bidimensional llamado `matriz` con los valores `[[1, 2, 3], [4, 5, 6], [7, 8, 9]]`. Accede al elemento `5` e imprímelo en la consola.
- 3. Iterar sobre un Array:
 - Usa un bucle `for` para iterar sobre el array `frutas` e imprimir cada elemento.
- 4. Uso de `map`:
 - Crea una función llamada `elevaAlCuadrado` que tome un array de números y devuelva un nuevo array con cada número elevado al cuadrado. Usa `map` para implementar la función.
- 5. Uso de `filter`:
 - Crea una función llamada `filtrarMayoresDe` que tome un array de números y un valor de referencia, y devuelva un nuevo array solo con los números mayores que ese valor. Usa `filter`.
- 6. Uso de `reduce`:
 - Crea una función llamada `sumarElementos` que tome un array de números y devuelva la suma de todos los elementos del array usando `reduce`.
- 7. Uso de `some`:
 - Crea un array de números llamado `numeros` y usa `some` para verificar si algún número es mayor que 10.
- 8. Uso de `every`:
 - Crea un array de números llamado `numeros` y usa `every` para verificar si todos los números son positivos.
- 9. Uso de `find`:
 - Crea un array de objetos llamados `personas` donde cada objeto tenga `nombre` y `edad`. Usa `find` para encontrar a la primera persona mayor de 30 años.
- 10. Uso de `sort`:
 - Crea un array de palabras y ordénalo alfabéticamente usando `sort`.

Mg. Ernesto

