

PROCESO DIRECCIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

Estimad@ aprendiz, la identificación de la presente guía está vacía para que usted pueda diligenciarla. Si tiene alguna duda pregúntele a su instructor líder.

1. IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE

- Denominación del Programa de Formación: Desarrollo de Aplicaciones en JavaScript
- Código del Programa de Formación:
- Nombre del Proyecto: Implementación y uso de funciones en JavaScript
- Fase del Proyecto: Creación y manejo de funciones en JavaScript.
- Actividad de Proyecto: Aplicación de conceptos de funciones en JavaScript para estructurar código y resolver problemas.
- Competencia: Definir, utilizar e invocar funciones en JavaScript, pasar parámetros y comprender su funcionamiento.
- Duración de la Guía: 5 Horas

2. PRESENTACIÓN

Las funciones en JavaScript son bloques de código reutilizables que ejecutan una tarea específica. Esta guía te enseñará a crear funciones, pasarles parámetros y valores, y estructurarlas para una mejor organización del código. Al finalizar, habrás comprendido los fundamentos de las funciones en JavaScript, y serás capaz de aplicar estos conceptos en proyectos más avanzados.

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

3.1 ACTIVIDAD DE REFLEXIÓN INICIAL

Reflexiona sobre los siguientes puntos y responde en un documento Word:

- ¿Por qué crees que es útil definir funciones en un programa?
- ¿Cómo piensas que la reutilización de código mediante funciones facilita el desarrollo y mantenimiento de un proyecto?

3.2 ACTIVIDAD DE APROPIACIÓN

A continuación, desarrollaremos diferentes aspectos relacionados con la creación, uso y manejo de funciones en JavaScript a través de ejemplos prácticos.

Paso 1: Crear Funciones en JavaScript Las funciones en JavaScript se definen usando la palabra clave `function`, seguida del nombre de la función, paréntesis y un bloque de código `{}`.

Ejemplo básico de función:

```
function saludar() {  
    console.log("¡Hola Mundo!");  
}
```

Paso 2: Invocar o Llamar una Función Para ejecutar la función, simplemente utiliza su nombre seguido de paréntesis.

Ejemplo:

```
saludar(); // Output: ¡Hola Mundo!
```

Paso 3: Funciones con Parámetros Los parámetros permiten que las funciones reciban valores externos, haciéndolas más flexibles y útiles. Puedes definir una función con uno o más parámetros entre los paréntesis.

Ejemplo con parámetros:

```
function saludar(nombre) {  
    console.log("¡Hola, " + nombre + "!");  
}  
  
saludar("Juan"); // Output: ¡Hola, Juan!
```

Paso 4: Funciones con Retorno de Valor Puedes devolver un valor con la palabra clave `return`. Esto es útil cuando quieres que la función proporcione un resultado que pueda ser almacenado o procesado.

Ejemplo de función con retorno:

```
function cuadrado(numero) {  
    return numero * numero;  
}  
  
let resultado = cuadrado(4); // Output: 16  
console.log(resultado);
```

Paso 5: Funciones con Parámetros Predeterminados Los parámetros predeterminados permiten definir valores iniciales en caso de que la función sea llamada sin argumentos.

Ejemplo con valor predeterminado:

```
function saludar(nombre = "Amigo") {  
    console.log("¡Hola, " + nombre + "!");  
}  
  
saludar(); // Output: ¡Hola, Amigo!  
saludar("Carlos"); // Output: ¡Hola, Carlos!
```

Paso 6: Funciones Anónimas y Funciones Flecha JavaScript permite crear funciones anónimas (sin nombre) y funciones flecha, que son una forma abreviada de declarar funciones.

Ejemplo de función anónima:

```
let sumar = function(a, b) {  
    return a + b;  
};  
  
console.log(sumar(5, 3)); // Output: 8
```

Ejemplo de función flecha:

```
let multiplicar = (a, b) => a * b;  
console.log(multiplicar(4, 2)); // Output: 8
```

Paso 7: Ejercicio Práctico Integrador Crea un programa que incluya las siguientes funciones:

- calcularAreaRectangulo(ancho, alto): Calcula y devuelve el área de un rectángulo.
- esPar(numero): Devuelve true si el número es par y false si es impar.
- concatenarCadenas(...cadenas): Recibe un número variable de cadenas y las une en una sola.

Ejemplo de implementación:

```
function calcularAreaRectangulo(ancho, alto) {  
    return ancho * alto;  
}
```

3.3. ACTIVIDAD DE TRANSFERENCIA DE CONOCIMIENTO

- **Consulta:** Investiga sobre el alcance de variables en funciones (scope) y cómo afecta el uso de variables var, let y const.
- **Evidencia:** Realiza un documento explicando los conceptos de alcance local y global, así como ejemplos del uso de var, let, y const en funciones.

Unifica las evidencias en una sólo archivo o carpeta, comprímelo y adjúntalo en el enlace que les comparta la instructora para tal fin.

¡Recuerda! No importa que algunas de las actividades sean grupales, las evidencias se montan de forma individual, de tal forma que todos los integrantes de los equipos tengan su trabajo subido en plataforma.

Si tienes dudas o inquietudes pregúntale a tu instructora.

3.4. ACTIVIDAD DE EVALUACIÓN

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
Evidencias de Conocimiento:	Explica los conceptos básicos de funciones en JavaScript y su importancia.	Técnica: Evaluación Colaborativa Instrumento: Lista de chequeo
Evidencias de Desempeño:	Desarrolla un archivo JavaScript que implemente las funciones vistas en el ejercicio integrador.	Técnica: Observación Instrumento: Lista de chequeo
Evidencias de Producto:	Entrega un archivo comprimido con los ejercicios completados y un documento de reflexión.	Técnica: Evaluación Colaborativa Instrumento: funcionalidad

4. GLOSARIO DE TÉRMINOS:

- **Función:** Bloque de código que realiza una tarea específica y puede ser llamado en cualquier parte del programa.
- **Parámetro:** Valores que se pasan a una función para su procesamiento.
- **Retorno:** Valor que una función devuelve al final de su ejecución.
- **Función Flecha:** Forma abreviada de crear funciones en JavaScript.
- **Alcance (Scope):** Contexto en el que una variable es accesible dentro del código.

5. REFERENTES BIBLIOGRÁFICOS

- <https://developer.mozilla.org/es/docs/Web/JavaScript/Guide>

6. CONTROL DEL DOCUMENTO

Nombre Autor (es)	Cargo	Dependencia		Fecha
Vanesa Medina Cuadrado	Instructora	ADSO	28 de octubre de 2024	

7. CONTROL DE CAMBIOS (diligenciar únicamente si realiza ajustes a la guía)

Nombre Autor (es)	Cargo	Dependencia		Fecha
	Instructor	ADSO	actualización	