



Introducción a las funciones puras

En este curso, exploraremos el concepto de las funciones puras, una piedra angular de la programación funcional. Aprenderemos qué son, cómo evitar los efectos colaterales y la importancia de la inmutabilidad en el desarrollo de software robusto y mantenible.

Definición de funciones puras

1

Determinismo

Las funciones puras siempre producen el mismo resultado para los mismos parámetros de entrada.

2

Sin efectos colaterales

Las funciones puras no modifican ningún estado fuera de su ámbito de acción.

3

Independencia

Las funciones puras no dependen de variables externas a sus parámetros de entrada.

Características de las funciones puras

Modularidad

Las funciones puras son altamente modulares, lo que facilita su reutilización y pruebas.

Composición

Pueden combinarse fácilmente para crear funciones más complejas y robustas.

Paralelización

Las funciones puras pueden ejecutarse en paralelo sin riesgos de condiciones de carrera.

Efectos colaterales y su evitación

Efectos Colaterales

Los efectos colaterales son cambios en el estado del sistema que ocurren fuera del ámbito de acción de una función.

Evitación de Efectos Colaterales

Para evitar efectos colaterales, debemos mantener las funciones puras y aislar los datos mutables.

Técnicas

Algunas técnicas incluyen pasar todos los datos necesarios como parámetros y devolver nuevos objetos en lugar de modificar los existentes.

Beneficios

La evitación de efectos colaterales mejora la modularidad, testabilidad y mantenibilidad del código.



Inmutabilidad y su importancia

Inmutabilidad

La inmutabilidad implica que los valores de los datos no pueden cambiar una vez que se han creado.

Implementación

En la práctica, la inmutabilidad se logra creando nuevas estructuras de datos en lugar de modificar las existentes.

1

2

3

Beneficios

La inmutabilidad facilita la depuración, mejora el rendimiento y permite la paralelización del código.

Ejemplos de funciones puras

Suma de Números

```
function suma(a, b) {  
  return a + b;  
}
```

Concatenación de Strings

```
function concatenar(str1, str2) {  
  return str1 + str2;  
}
```

Calcular Área de un Círculo

```
function areaCirculo(radio) {  
  return Math.PI * radio * radio;  
}
```


Beneficios de utilizar funciones puras



Modularidad

Las funciones puras son altamente modulares y reutilizables.



Testabilidad

Las funciones puras son más fáciles de probar y depurar.



Rendimiento

Las funciones puras pueden ser paralelizadas sin problemas.



Mantenibilidad

El código basado en funciones puras es más fácil de mantener.