

# What is This?

Seguirà una panoramica approfondita su come funziona la parola chiave `this` in JavaScript

## Cos'è this?

La parola chiave `this` è un riferimento speciale che punta al contesto di esecuzione della funzione corrente.

Se viene utilizzato fuori da un funzione fa riferimento al contesto globale.

Nei browser, questo oggetto è `window`.

## Nel contesto delle Funzioni

In una funzione normale this si riferisce all'oggetto che chiama la funzione. Se la funzione viene chiamata nel contesto globale, `this` punterà all'oggetto globale.

```
function mostraQuesto() {  
    console.log(this);  
}  
  
mostraQuesto(); // In modalità non-strict, registra l'oggetto Window  
  
const obj = {  
    nome: 'Esempio',  
    mostraQuesto: mostraQuesto  
};  
obj.mostraQuesto(); // Registra l'oggetto obj
```

### Spiegazione esempio:

- Quando una funzione viene invocata come chiamata di funzione, JavaScript imposta per default this all'oggetto globale.
- Quando una funzione viene invocata come metodo di un oggetto, this punta all'oggetto.

## Contesto del Metodo

Quando una funzione è chiamata come metodo di un oggetto, this fa riferimento all'oggetto. Questo è importante per accedere alle proprietà e ai metodi dell'oggetto stesso.

```
const persona = {  
    nome: 'Alice',  
    saluta: function() {  
        console.log(`Ciao, mi chiamo ${this.nome}`);  
    }  
};  
  
persona.saluta(); // Registra: "Ciao, mi chiamo Alice"
```

## Contesto del Costruttore

Quando si utilizza una funzione costruttore con la parola chiave new, this si riferisce alla nuova stanza. Le funzioni costruttore sono il modo in cui JavaScript simula le classi.

```
function Persona(nome) {  
    this.nome = nome;  
}  
  
const alice = new Persona('Alice');  
console.log(alice.nome); // Registra: "Alice"
```

# Errori Comuni con `this`

### ESEMPIO ERRORE

Uno degli errori più comuni in JavaScript è perdere il riferimento a this quando si passa una funzione come callback. In questi casi, this potrebbe non fare più riferimento all'oggetto atteso.

```
const obj = {  
    nome: 'Alice',  
    saluta: function() {  
        console.log(`Ciao, mi chiamo ${this.nome}`);  
    }  
};  
  
setTimeout(obj.saluta, 1000); // `this` è perso, registra: "Ciao, mi chiamo undefined"
```

### SOLUZIONE

- Il this perde il suo contesto, causando this a riferirsi all'oggetto globale o a undefined.
- Per risolvere questo problema, puoi utilizzare .bind() per mantenere il contesto corretto.

```
setTimeout(obj.saluta.bind(obj),  
1000); // Registra correttamente:  
"Ciao, mi chiamo Alice"
```