

# JavaScript Callback-Funktionen

## Was sind Callback-Funktionen?

A callback is a function passed as an argument to another function. [w3Schools](#)

- Callback-Funktionen sind Funktionen in JavaScript, die als Parameter an eine andere Funktion übergeben werden und später aufgerufen werden können. Sie werden häufig verwendet, um asynchrone Operationen zu behandeln oder um Code modularer und wiederverwendbarer zu machen.

## Wieso gibt es Callback-Funktionen?

- Callback-Funktionen ermöglichen es, Code asynchron auszuführen, anstatt ihn blockierend zu machen. Sie werden verwendet, um auf das Ergebnis einer asynchronen Operation zu reagieren oder um eine Funktion nach Abschluss einer bestimmten Aufgabe auszuführen.

## Wie funktionieren Callback-Funktionen?

- Eine Callback-Funktion wird als Parameter an eine andere Funktion übergeben und von dieser Funktion aufgerufen, sobald eine bestimmte Bedingung oder Aktion erfüllt ist.
- Die übergebene Callback-Funktion kann entweder vordefiniert sein oder als anonyme Funktion direkt übergeben werden.

## Beispiel für die Verwendung von Callback-Funktionen:

```
function asyncOperation(callback) {
  setTimeout(() => {
    const result = 42;
    callback(result);
  }, 1000);
}

function handleResult(result) {
  console.log('Ergebnis:', result);
}

asyncOperation(handleResult);
```

## Häufige Anwendungsgebiete von Callback-Funktionen:

- Behandlung von asynchronen Operationen wie Datei- oder Netzwerkzugriffen.
- Event-Handler für Benutzerinteraktionen wie Mausklicks oder Tastatureingaben.
- Verwendung in Higher-Order-Funktionen wie Array-Methoden (z.B. `forEach`, `map`, `filter`), um benutzerdefinierte Logik auf jedes Element anzuwenden.