



Async

ASYNC

Supponiamo di avere una funzione che restituisce una promise

```
function resolveAfter2Seconds() {  
  return new Promise(/* ... */);  
}
```

A questo possiamo scrivere una funzione con un comportamento asincrono come la seguente

```
function asyncCall() {  
  console.log('calling');  
  resolveAfter2Seconds().then(result => {  
    // posso usare result  
  }).catch(error => {  
    // posso usare error  
  }).finally(() => {  
    // finally  
  });  
}
```

ASYNC

Con la reserved keyword `async` possiamo esplicitare il comportamento asincrono della nostra funzione e gestire la *Promise* come segue.

```
async function asyncCall() {  
  console.log('calling');  
  try {  
    var result = await resolveAfter2Seconds();  
    // posso usare result  
  } catch (error) {  
    // posso usare error  
  }  
  // finally  
}
```

Con l'espressione `await resolveAfter2Seconds()` l'esecuzione viene "bloccata" in attesa di risoluzione della promise. In caso di passaggio della *Promise* in stato *fulfilled* l'espressione ritornerà il valore tornato dal `resolve`. In caso di errore verrà lanciato un errore con il valore ritornato da `reject`.

Le operazioni che andremmo a fare nel `finally` possono essere semplicemente eseguite dopo lo statement di `catch`.