

# Optional catch Binding

**Summary:** in this tutorial, you will learn how to use the optional catch binding in the `try...catch` statement.

## Introduction to the optional catch binding

The `try...catch` statement is used to handle any errors that may occur. Generally, you place the code that may cause an error in the `try` block and the code that handles the error in the `catch` block, like this:

```
try {  
    // code that may cause an error  
} catch (error) {  
    // code that handles the error  
}
```

In the `catch` block, you can access the `Error` object that contains detailed information on the error.

In practice, you may want to use the `try...catch` statement to check if a feature is implemented in the web browser. If it isn't, you want to fall back to a less desirable feature with broader support, for example:

```
try {  
    // check if a feature is implemented  
} catch (error) {  
    // fall back to a less desirable feature  
}
```

In this case, the `error` object is declared but never used.

ES2019 introduced the optional catch binding that allows you to omit the `catch` binding and its surrounding parentheses, like this:

```
try {  
  
} catch {  
  
}
```

## Summary

- Since ES2019, you can omit the `catch` binding in the `try...catch` statement.