# Strict mode

## Overview

The **strict mode** was introduced in ECMAScript 5. It is a way to **add a strict checking in JavaScript** that would make fewer mistakes.

JavaScript **allows strict mode code and non-strict mode code to coexist**. So you can add your new JavaScript code in strict mode in old JavaScript files.

Strict mode introduces several restrictions to the JavaScript code, like eliminating some silent errors by throwing errors.

You can introduce a strict mode in your JavaScript code by writing this simple statement.

**Syntax :** 'use strict' ;
OR
"use strict" ;

You can apply strict mode to an entire script or individual function -

- Write this at the top of the whole script to **apply strict mode globally.**
- Or write it inside functions to **apply it to a particular function only.**

## Using strict mode globally

**Example :**

```
x = 10 ;
console.log(x) ; // Outputs 10
```

```
"use strict"
x = 10 ;
console.log(x) ; // ReferenceError: x is not defined
```

- Creating variables without a literal is **not allowed in strict mode**

## Using strict mode within a function

**Example :**

```
function abc(a, a) {
    console.log(a + a);
}
abc(10, 20) ;  // Output 40
```

```
function abc(a, a) {
    'use strict' ;
    console.log(a + a);
 }
 abc(10, 20);  // Duplicate parameter name not allowed in this context
```

# Not Allowed in Strict Mode

Names that can not be a variable name under strict mode-

- eval
- arguments

**Example :**

```
"use strict" ;
var eval = 10 ;
```

**Output :** `SyntaxError: Unexpected eval or arguments in strict mode`

```
"use strict" ;
var arguments = 10 ;
```

**Output :** `SyntaxError: Unexpected eval or arguments in strict mode`

## this keyword in strict mode

Strict mode puts a restriction on the value of this, and it will be **undefined if in a global context,** the function is not bound to any object. Whereas in the sloppy mode, it was set to 'window' object.

**Inside a function**, if the **value of this is not set by the call, it will be undefined.** This is because the function is called directly and not as a method( window.abc( ) ).

**Example :**

```
function abc( ) {
        'use strict'
        console.log(this);
}

abc( ); // Prints – undefined
window.abc( ); // Prints window object
```