# Advance Javascript

## What is ‘use strict’ in javascript?

Ans: You can not use a variable without declaring it.

When you use variable without declaring it as var in JS, it automatically creates a Global variable with that name.

In JS, there is already a global object. In the browser, that global object is known as ‘window’. In node, that global object name is Global.

1. variable = 1;
2. console.log(window.variable); // prints 1 in console.

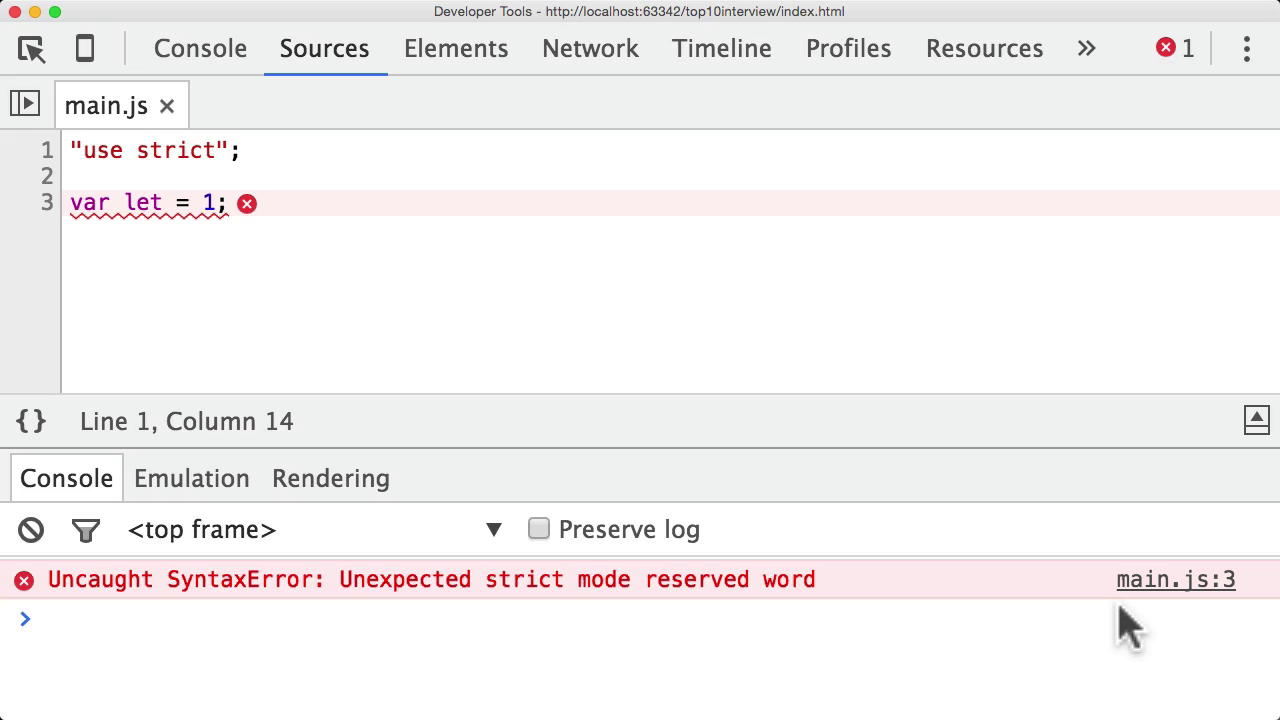
* That makes debugging easier.
* You can enable it by using the string ‘use strict’. You should use it by enclosing it via single quotes or double quotes because these are not reserved keyword. It is only a string.
* Why it is a string? Well when it was featured with JS, older version of browser could not able to support if it was a keyword. So now, if it is a string-type, when older version of browser interprets it as string and ignore it. While at the same time, Newer version of browser see this string and apply the strict mode for JS.
* You can enable it for a full JS file by describing at top of the file as

1. ‘use strict’

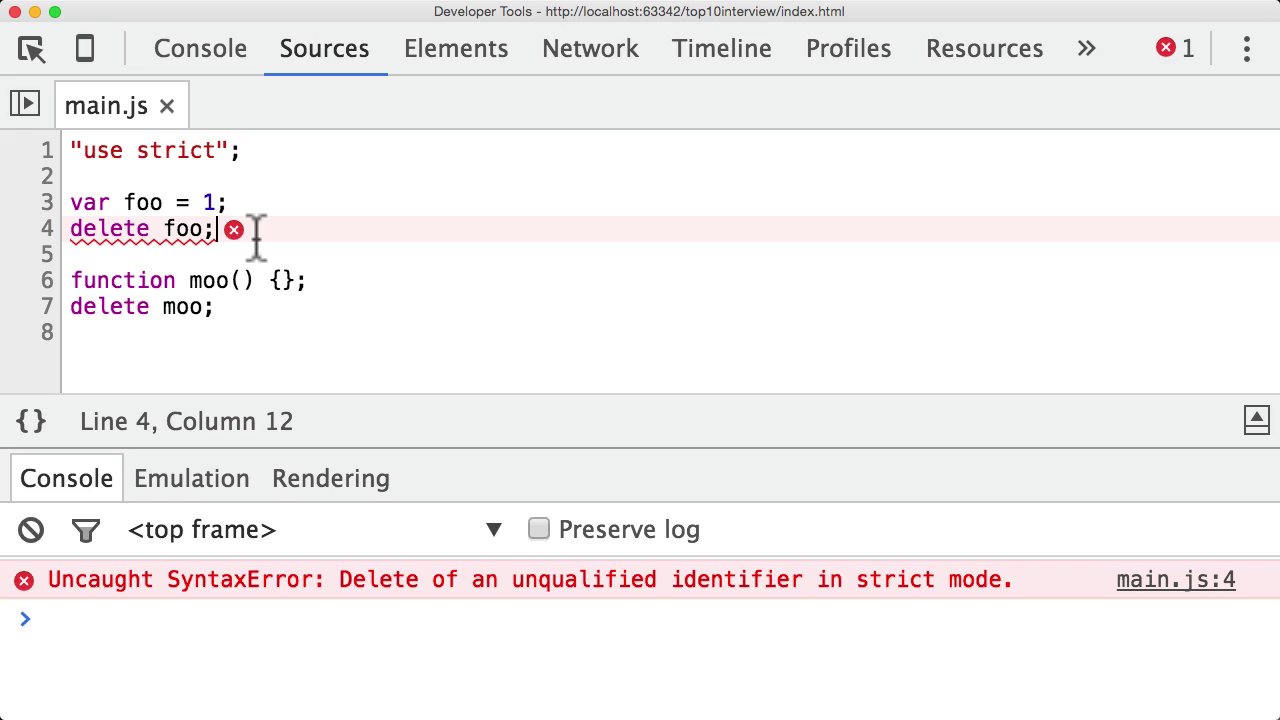
* You can enable it for a single function too as

1. // not strict mode
2. Function function\_name (){
3. ‘use  strict’
4. // strict mode applied for this function
5. }

* It helps you to prevent using the keywords as your variable name, which are reserved by the future version of JS.

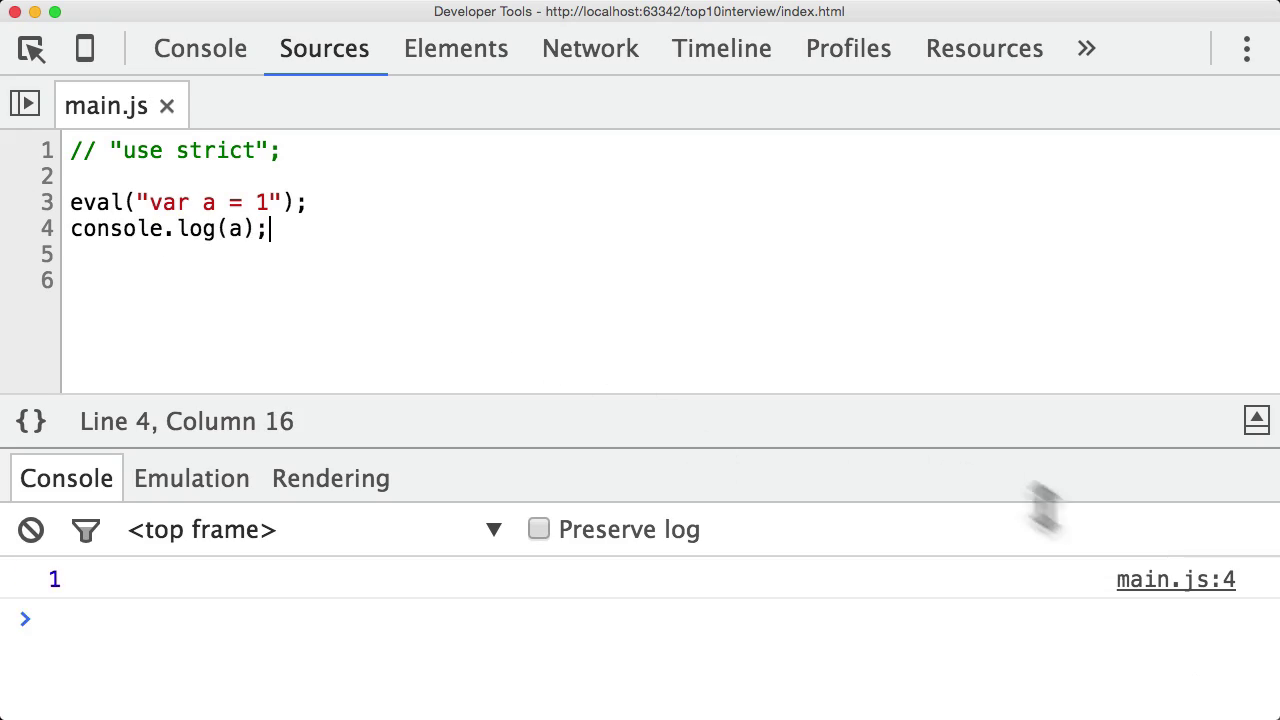


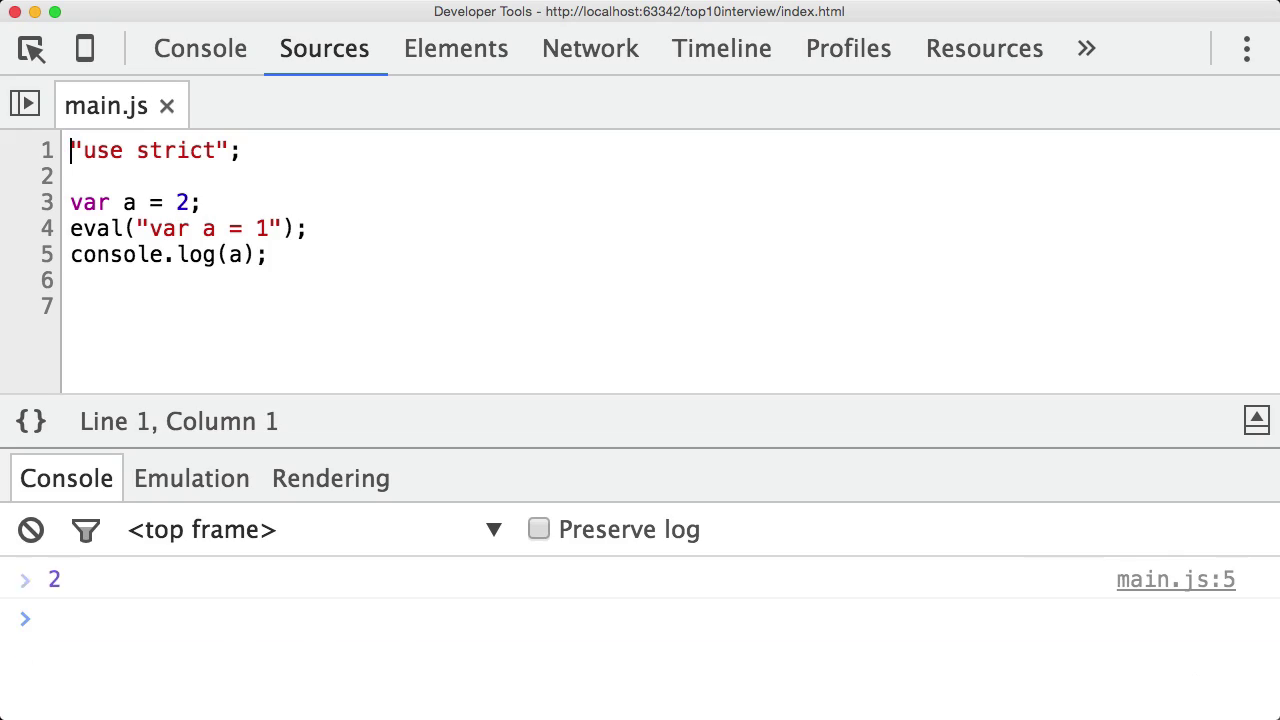
* You can not delete variable, objects or functions, inside ‘use strict mode’. Even through you can not delete argument of a function.



* It makes use of eval a little bit safe. In JS, eval let you evaluate JS expresssions by passing as string inside it.

1. eval(‘var a = 1’);

* But in non-strict mode, value of expression leaks out to JS file



So, when you use strict mode, any variable that you define inside eval, remains inside eval and doesn’t leak out to JS file.

## Ques 2: Does javascript pass variable by reference or value?

Ans: Primitive Data type such as String, Number or Boolean are passed by value while objects are passed by reference.

1. var a = 1;
2. function foo(a) { // Here variable a is passed by value
3. a=2;
4. }
5. foo(a);
6. console.log(a); // prints 1

Note: What is pass by value? Well if you change the value of the primitive data type inside a function, it won’t effect the variable outside the function.

1. var cdf = {
2. "moo": 'roo'
3. };
5. function foo(cdf) {
6. cdf = {
7. 'too': 'moo'
8. };
9. }
10. foo(cdf);
11. console.log(cdf);  // prints {moo : ‘roo’}

In JS, variable pass by reference means we can’t change, what a variable(object) points to. While we can add some property, change a property value.

1. var cdf = {
2. "moo": 'roo'
3. };
5. function foo(cdf) {
6. cdf.too = 'too';
7. }
8. foo(cdf);
9. console.log(cdf);  // prints {moo: "roo", too: "too"}

## Question: 3 What are the different types in JS?

Answer: