**What is javasctipt obfuscator ?**

Javascript Obfuscator converts the JavaScript source code into obfuscated and completely unreadable form, preventing it from analyzing and theft while fully retaining the functionality of the original code. It's a 100% safe JavaScript minifier and the best JavaScript compressor.

Through a form of masking that makes source code unintelligible to the naked eye, such obfuscators raise the barrier to undesirable inspection of Javascript that is otherwise discoverable through the viewing of HTML and .JS files. Though it's hardly foolproof, obfuscators can be useful in situations where easy inspection of source code could lead to the exposure of sensitive information that developers would prefer to be less discoverable (like API keys and application secrets).

**Why should I use javascript obfuscator?**

It's simple really, JavaScript is a scripting programming language so it needs to be front end and visible to browsers. This has several drawbacks and the most important one is that JavaScript source code can be easily viewed by anyone. This means that anyone can steal your JavaScript code, modify it and reuse it for his own purposes. One of the ways to protect your code is to obfuscate it to encrypt the code and make it unreadable.

* Protect your code and intellectual property.
* Compact js to also make your file sizes smaller and increase page speed.
* Work with jQuery!, NODE.JS and other JS frameworks.
* Always to preserve the operational qualities of the code

**Is JavaScript Obfuscator suitable for my project?**

Yes, JavaScript Obfuscator is very flexible and advanced product and it can be used for project of any kind and size.

**Is it possible to recover original file from obfuscated version?**

No, it's impossible to recover it - the names of symbols and comments are gone forever. So keep the original version of your code in a safe place!

**Does Javascript Obfuscator optimize for speed?**

In most cases smaller code is faster code, since download time is usually the most important speed factor in web applications. Optimizations that reduce redundancies speed up the run time of code as well.

**Do you keep my JavaScript source code?**

No. The source code obfuscation is processed in server memory. You don't need to upload any Javascript files to the server. For more information please read our Terms of Service and Privacy Policy.

For example :

Following JS code removes duplicates and sort using bubble sort

**Normal JS code**

**var arrUnSort =['a','b','c','d','a','b','d','d','e','b','c','i','j','z'];**

**Array.prototype.removeSortandDuplicate = function(){**

**//this is pointing to array**

**var passNum = this.length;**

**// to repeat loops n\*n times**

**while(passNum>0){**

**for(var i=0;i<this.length;i++){**

**//remove duplicates from an array**

**if(this[i]==this[i+1]){**

**var \_indx = this.indexOf(this[i]);**

**this.splice(\_indx,1);**

**}**

**//sort and array using bubble sort**

**if(this[i]>this[i+1]){**

**var \_temp = this[i];**

**this[i] = this[i+1];**

**this[i+1] =\_temp;**

**}**

**}**

**passNum = passNum-1;**

**}**

**return this;**

**}**

**console.log("Deobfuscate JS O/p" , arrUnSort.removeSortandDuplicate());**

**JS obfuscate code**

**//JS obfuscate code**

**var \_0x535a=["\x61","\x62","\x63","\x64","\x65","\x69","\x6A","\x7A","\x72\x65\x6D\x6F\x76\x65\x53\x6F\x72\x74\x61\x6E\x64\x44\x75\x70\x6C\x69\x63\x61\x74\x65","\x70\x72\x6F\x74\x6F\x74\x79\x70\x65","\x6C\x65\x6E\x67\x74\x68","\x69\x6E\x64\x65\x78\x4F\x66","\x73\x70\x6C\x69\x63\x65","\x4F\x62\x66\x75\x73\x63\x61\x74\x65\x20\x4A\x53\x20\x4F\x2F\x70","\x6C\x6F\x67"];var arrUnSort=[\_0x535a[0],\_0x535a[1],\_0x535a[2],\_0x535a[3],\_0x535a[0],\_0x535a[1],\_0x535a[3],\_0x535a[3],\_0x535a[4],\_0x535a[1],\_0x535a[2],\_0x535a[5],\_0x535a[6],\_0x535a[7]];Array[\_0x535a[9]][\_0x535a[8]]=function(){var \_0xa47dx2=this[\_0x535a[10]];while(\_0xa47dx2>0){for(var \_0xa47dx3=0;\_0xa47dx3<this[\_0x535a[10]];\_0xa47dx3++){if(this[\_0xa47dx3]==this[\_0xa47dx3+1]){var \_0xa47dx4=this[\_0x535a[11]](this[\_0xa47dx3]);this[\_0x535a[12]](\_0xa47dx4,1)};if(this[\_0xa47dx3]>this[\_0xa47dx3+1]){var \_0xa47dx5=this[\_0xa47dx3];this[\_0xa47dx3]=this[\_0xa47dx3+1];this[\_0xa47dx3+1]=\_0xa47dx5}};\_0xa47dx2=\_0xa47dx2-1};return this};console[\_0x535a[14]](\_0x535a[13],arrUnSort[\_0x535a[8]]())**

**Both code does same functionality but obfuscate code will not able to understand by humans and it is only able to understand by browser or JS engines.**

**Create sample.html file and paste following code and check the console.**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="jquery-1.10.2.js"></script>**

**</head>**

**<body>**

**<script>**

**var arrUnSort =['a','b','c','d','a','b','d','d','e','b','c','i','j','z'];**

**Array.prototype.removeSortandDuplicate = function(){**

**//this is pointing to array**

**var passNum = this.length;**

**// to repeat loops n\*n times**

**while(passNum>0){**

**for(var i=0;i<this.length;i++){**

**//remove duplicates from an array**

**if(this[i]==this[i+1]){**

**var \_indx = this.indexOf(this[i]);**

**this.splice(\_indx,1);**

**}**

**//sort and array using bubble sort**

**if(this[i]>this[i+1]){**

**var \_temp = this[i];**

**this[i] = this[i+1];**

**this[i+1] =\_temp;**

**}**

**}**

**passNum = passNum-1;**

**}**

**return this;**

**}**

**console.log("Deobfuscate JS O/p" , arrUnSort.removeSortandDuplicate());**

**//JS obfuscate code**

**var \_0x535a=["\x61","\x62","\x63","\x64","\x65","\x69","\x6A","\x7A","\x72\x65\x6D\x6F\x76\x65\x53\x6F\x72\x74\x61\x6E\x64\x44\x75\x70\x6C\x69\x63\x61\x74\x65","\x70\x72\x6F\x74\x6F\x74\x79\x70\x65","\x6C\x65\x6E\x67\x74\x68","\x69\x6E\x64\x65\x78\x4F\x66","\x73\x70\x6C\x69\x63\x65","\x4F\x62\x66\x75\x73\x63\x61\x74\x65\x20\x4A\x53\x20\x4F\x2F\x70","\x6C\x6F\x67"];var arrUnSort=[\_0x535a[0],\_0x535a[1],\_0x535a[2],\_0x535a[3],\_0x535a[0],\_0x535a[1],\_0x535a[3],\_0x535a[3],\_0x535a[4],\_0x535a[1],\_0x535a[2],\_0x535a[5],\_0x535a[6],\_0x535a[7]];Array[\_0x535a[9]][\_0x535a[8]]=function(){var \_0xa47dx2=this[\_0x535a[10]];while(\_0xa47dx2>0){for(var \_0xa47dx3=0;\_0xa47dx3<this[\_0x535a[10]];\_0xa47dx3++){if(this[\_0xa47dx3]==this[\_0xa47dx3+1]){var \_0xa47dx4=this[\_0x535a[11]](this[\_0xa47dx3]);this[\_0x535a[12]](\_0xa47dx4,1)};if(this[\_0xa47dx3]>this[\_0xa47dx3+1]){var \_0xa47dx5=this[\_0xa47dx3];this[\_0xa47dx3]=this[\_0xa47dx3+1];this[\_0xa47dx3+1]=\_0xa47dx5}};\_0xa47dx2=\_0xa47dx2-1};return this};console[\_0x535a[14]](\_0x535a[13],arrUnSort[\_0x535a[8]]())**

**</script>**

**</body>**

**</html>**