



# The modern mode, "use strict"

For a long time, JavaScript evolved without compatibility issues. New features were added to the language while old functionality didn't change.

That had the benefit of never breaking existing code. But the downside was that any mistake or an imperfect decision made by JavaScript's creators got stuck in the language forever.

This was the case until 2009 when ECMAScript 5 (ES5) appeared. It added new features to the language and modified some of the existing ones. To keep the old code working, most such modifications are off by default. You need to explicitly enable them with a special directive: `"use strict"`.

## “use strict”

The directive looks like a string: `"use strict"` or `'use strict'`. When it is located at the top of a script, the whole script works the “modern” way.

For example:

```
"use strict";

// this code works the modern way
...
```

Quite soon we're going to learn functions (a way to group commands), so let's note in advance that `"use strict"` can be put at the beginning of a function. Doing that enables strict mode in that function only. But usually people use it for the whole script.

### Ensure that “use strict” is at the top

Please make sure that `"use strict"` is at the top of your scripts, otherwise strict mode may not be enabled.

Strict mode isn't enabled here:

```
alert("some code");
// "use strict" below is ignored--it must be at the top

"use strict";

// strict mode is not activated
```

Only comments may appear above `"use strict"`.

### There's no way to cancel `use strict`

There is no directive like `"no use strict"` that reverts the engine to old behavior.

Once we enter strict mode, there's no going back.

## Browser console

When you use a [developer console](#) to run code, please note that it doesn't `use strict` by default.

Sometimes, when `use strict` makes a difference, you'll get incorrect results.

So, how to actually `use strict` in the console?

First, you can try to press `Shift+Enter` to input multiple lines, and put `use strict` on top, like this:

```
'use strict'; <Shift+Enter for a newline>
// ...your code
<Enter to run>
```

It works in most browsers, namely Firefox and Chrome.

If it doesn't, e.g. in an old browser, there's an ugly, but reliable way to ensure `use strict`. Put it inside this kind of wrapper:

```
(function() {
  'use strict';

  // ...your code here...
})();
```

## Should we “use strict”?

The question may sound obvious, but it's not so.

One could recommend to start scripts with `"use strict"` ... But you know what's cool?

Modern JavaScript supports “classes” and “modules” – advanced language structures (we'll surely get to them), that enable `use strict` automatically. So we don't need to add the `"use strict"` directive, if we use them.

**So, for now `"use strict";` is a welcome guest at the top of your scripts. Later, when your code is all in classes and modules, you may omit it.**

As of now, we've got to know about `use strict` in general.

In the next chapters, as we learn language features, we'll see the differences between the strict and old modes. Luckily, there aren't many and they actually make our lives better.

All examples in this tutorial assume strict mode unless (very rarely) specified otherwise.

[Previous lesson](#)[Next lesson](#)

Share

[Tutorial map](#)

## Comments

- If you have suggestions what to improve - please [submit a GitHub issue](#) or a pull request instead of commenting.
- If you can't understand something in the article – please elaborate.
- To insert a few words of code, use the `<code>` tag, for several lines – use `<pre>`, for more than 10 lines – use a sandbox ( [plnkr](#), [JSBin](#), [codepen](#)...)

© 2007—2020 Ilya Kantor

[about the project](#)

[contact us](#)

[terms of usage](#)

[privacy policy](#)