1st December 2019

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
...
```

We will learn functions (a way to group commands) soon. Looking ahead, let's note that `"use strict"` can be put at the beginning of the function body instead of the whole script. 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

For the future, when you use a browser console to test features, please note that it doesn’t `use strict` by default.

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

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, the most reliable way to ensure `use strict` would be to input the code into console like this:

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

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

Always “use strict”

We have yet to cover the differences between strict mode and the “default” mode.

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

For now, it's enough to know about it in general:

1. The `"use strict"` directive switches the engine to the "modern" mode, changing the behavior of some built-in features. We'll see the details later in the tutorial.
2. Strict mode is enabled by placing `"use strict"` at the top of a script or function. Several language features, like "classes" and "modules", enable strict mode automatically.
3. Strict mode is supported by all modern browsers.
4. We recommended always starting scripts with `"use strict"`. 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 Kantorabout the projectcontact usterms of usage privacy policy