# JS use strict

"use strict" – это директива, введенная стандартом ECMAScript 5. Назначение директивы "use strict" состоит в том, чтобы показать, что следующий за ней программный код (в сценарии или функции) является строгим кодом.

* Строгим считается программный код верхнего уровня (не внутри функций), если в сценарии имеется директива "use strict".
* Строгим считается тело функции, если она определяется внутри строгого программного кода или если она содержит директиву "use strict".
* Строгим считается программный код, передаваемый методу eval(), если вызов eval() выполняется из строгого программного кода или если строка с кодом содержит директиву "use strict".

*Строгий программный код выполняется в строгом режиме.*

Согласно стандарту ECMAScript 5, строгий режим определяет ограниченное подмножество языка, благодаря чему исправляет некоторые недостатки языка, а также обеспечивает более строгую проверку на наличие ошибок и повышенный уровень безопасности.

Ниже перечислены различия между строгим и нестрогим режимами (первые три имеют особенно большое значение):

1. В строгом режиме не допускается использование инструкции with.
2. В строгом режиме все переменные должны объявляться.
3. В строгом режиме функции, которые вызываются как функции (а не как методы), получают в ссылке this значение undefined.
4. В строгом режиме попытки присвоить значения свойствам, недоступным для записи, или создать новые свойства в нерасширяемых объектах порождают исключение TypeError.
5. В строгом режиме программный код, передаваемый функции eval(), не может объявлять переменные или функции в области видимости вызывающего программного кода, как это возможно в нестрогом режиме.
6. В строгом режиме объект arguments в функции хранит статическую копию значений, переданных функции. В нестрогом режиме объект arguments ведет себя иначе – элементы массива arguments и именованные параметры функции ссылаются на одни и те же значения.
7. В строгом режиме возбуждается исключение SyntaxError, если оператору delete передать неквалифицированный идентификатор, такой как имя переменной, функции или параметра функции.
8. В строгом режиме попытка удалить ненастраиваемое свойство приведет к исключению TypeError.
9. В строгом режиме попытка определить в литерале объекта два или более свойств с одинаковыми именами считается синтаксической ошибкой.
10. В строгом режиме определение двух или более параметров с одинаковыми именами в объявлении функции считается синтаксической ошибкой.
11. В строгом режиме не допускается использовать литералы восьмеричных целых чисел (начинающихся с 0, за которым не следует символ x). (В нестрогом режиме некоторые реализации позволяют использовать восьмеричные литералы.)
12. В строгом режиме идентификаторы eval и arguments интерпретируются как ключевые слова, и вы не можете изменить их значения. Вы сможете присвоить значения этим идентификаторам, объявив их как переменные, использовав их в качестве имен функций, имен параметров функций или идентификаторов блока catch.
13. В строгом режиме ограничивается возможность просмотра стека вызовов. Попытки обратиться к свойствам arguments.caller и arguments.callee в строгом режиме возбуждают исключение TypeError. Попытки прочитать свойства caller и arguments функций в строгом режиме также возбуждают исключение TypeError.