**"use strict"**

1. Usunięcie właściwości WITH
2. nie można wyświetlić wartości zmiennej jesli nie jest ona zaadeklarowana:
   1. zmienna = "Samochód"; console.log(zmienna) //ReferenceError: zmienna is not defined
3. ograniczenie możliwości nadpisania zmiennych globalnych
   1. NaN = 3; // TypeError: Cannot assign to read only property 'NaN' of object '#<Window>'
   2. Infinity = 3; // TypeError:
4. ograniczenie możliwości zadeklarowania tej samej zmiennej

var test = {

a: 1,

b: 2,

a: 3 // error

};

1. ograniczenie możliwości zadeklarowania tej samej zmiennej w funkcji
   1. function overrideParameters(a, b, a) { // SyntaxError: Duplicate parameter name not allowed in this context
2. Nie można również nadpisywać zmiennej „arguments”. Taka próba prowadzi do błędu

function argumentsExample1(a, b, c) {

"use strict";

arguments = 1; // SyntaxError: Unexpected eval or arguments in strict mode

}

1. nie można nadpisać zmiennej "arguments:

argumentsExample2(1, 2, 3);

function argumentsExample2(a, b, c) {

"use strict";

a = -1;

console.log(arguments); // 1, 2, 3

}

1. Tryb ścisły nie pozwala używać słów, które będą (lub już są) uznane za słowa kluczowe standardu. Do tych słów należą:
   1. implements
   2. interface
   3. let
   4. package
   5. private
   6. protected
   7. public
   8. static
   9. yield
2. kasowanie zmiennej lub funkcji za pomocą „delete”
3. używanie składni ósemkowej (var x = 010)
4. nadpisywanie właściwości tylko do odczytu
5. użycie zmiennej „eval” jako zmiennej (var eval = „test”)
6. Na koniec warto podać jakie wersje wspierają tryb ścisły:
   1. IE od wersji 10.
   2. Firefox od wersji 4.
   3. Chrome od wersji 13.
   4. Safari od wersji 5.1.
   5. Opera od wersji 12.

ES6

1. Wykorzystanie let, const. Działają tylko wewnątrz bloku a nie poza nim.
2. template string. Mozliwośc zapisywania ciągów znaków w róznych liniach za pomocą: ``
3. dodstanie sie do elementów tablicy za pomoca skróconego zapisu ...arr(3 kropki i nazwa tablicy)
4. wypisanie pozostałych argumentów z funkcji:

function argumenty(a1,a2, ...rest){//1,2,3,4,5,6,7

console.log('pierwsze dwa to: ',a1, a2);//1,2

console.log('pozostałe argumenty to: ', ...rest);//3,4,5,6,7

}

1. Nowa forma zapisywania funkcji
   1. old -> function wypisz(wyraz){console.log('Twój wyraz to:', wyraz);}
   2. new -> var wypisz2 = (wyraz) => {console.log('Twój wyraz to:', wyraz);}
2. Możliwośc przypisywania wartości domyślnej

function def(parm = "Auto"){console.log(parm);}

def()//auto

def('dom');//dom