Front-end

JavaScript

JS. use strict

- use strict

JS. use strict

Строгий режим или Strict — это функция, представленная в ECMAScript 5 (ES5), которая позволяет разработчикам использовать более строгий и безопасный вариант JavaScript.

При активации он применяет строгие правила и ограничения, помогая выявлять распространенные ошибки и повышать качество кода. Строгий режим можно включить для каждого файла или функции, что позволяет разработчикам выбирать, где применять его ограничения.

JS. use strict

Зачем использовать строгий режим

- **Предотвращение ошибок**. Строгий режим помогает предотвратить потенциальные ошибки, которые в обычном режиме просто игнорировались бы.
- **Упрощенная отладка**. Благодаря раннему обнаружению ошибок строгий режим может ускорить и повысить эффективность отладки.
- **Улучшенная производительность**. Некоторые оптимизации возможны только в строгом режиме, поскольку он устраняет некоторые подверженные ошибкам функции языка.
- **Забота о будущем**. Строгий режим запрещает использование устаревших функций или функций, которые скоро станут устаревшими, помогая вашему коду оставаться совместимым с будущими версиями JavaScript.

JS. use strict

Как включить строгий режим

Включение строгого режима выполняется прямо в коде. Просто добавьте следующее в начало файла JavaScript:

```
"use strict";
```

Эта директива информирует интерпретатор JavaScript о необходимости применения правил Strict для всего файла или функции, в которой он появляется.

Имейте в виду, что включение строгого режима в одном скрипте или функции не влияет на другие скрипты или функции.

JS. use strict

Объявление переменной

В строгом режиме вы должны объявить переменные с помощью ключевых слов var, let или const перед их использованием. В противном случае возникнет ошибка.

Это предотвратит случайное создание глобальной переменной.

```
// Non-strict mode
undeclaredVariable = 42; // Creates a global variable

// Strict mode
"use strict";
undeclaredVariable = 42; // Throws a ReferenceError
```

JS. use strict

Повторяющиеся имена параметров

Строгий режим запрещает функции с повторяющимися именами параметров. При попытке использовать их возникнет синтаксическая ошибка.

```
// Non-strict mode
function duplicateParameters(a, a) { // No error
}

// Strict mode
"use strict";
function duplicateParameters(a, a) { // Throws a SyntaxError
}
```

JS. use strict

Значение this

В нестрогом режиме значение this внутри функции, вызываемой без явного получателя (например, в качестве отдельной функции, а не метода), по умолчанию равно глобальному объекту.

В строгом режиме **this** является **undefined**, что помогает предотвратить случайные изменения глобального объекта.

JS. use strict

Значение this

```
// Non-strict mode
function logThis() {
  console.log(this);
}
logThis(); // Logs the global object (e.g., `window` in browsers)

// Strict mode
"use strict";
function logThis() {
  console.log(this);
}
logThis(); // Logs `undefined`
```

JS. use strict

Явная ошибка если значение поля нельзя изменить или удалить

С помощью методов Object.defineProperty() или Object.preventExtensions() в JavaScript можно запретить перезаписывать поля объекта. При включённом строгом режиме попытка перезаписать поле приведёт к ошибке.

```
'use strict'

const obj = {}


Object.defineProperty(obj, 'someProp', { value: 'Alex', writable:false })

console.log(obj.someProp)

// Alex


obj.someProp = 'James'

// Uncaught TypeError: Cannot assign to read only property 'someProp' of object #<Object>
```

JS. use strict

Явная ошибка если значение поля нельзя изменить или удалить

```
'use strict'

const notExtensableObj = {}

Object.preventExtensions(notExtensableObj) // нерасширяемый объект

notExtensableObj.someProp = 'Value'
// Uncaught TypeError: Can't add property someProp, object is not extensible
```

JS. use strict

Явная ошибка если значение поля нельзя изменить или удалить

Ошибка будет выброшена в строгом режиме и при попытке удаления поля из объекта, когда это сделать нельзя.

```
const obj = {}


Object.defineProperty(obj, 'someProp', {
   value: 'Anna',
   configurable: false
})


delete obj.someProp
// Uncaught TypeError: Cannot delete property 'someProp' of #<Object>
```

JS. use strict

Явная ошибка если значение поля нельзя изменить или удалить

Если запустить все примеры кода выше без строгого режима, они выполнятся без ошибок, но значения полей не изменятся.