

Эта страница была переведена с английского языка силами сообщества. Вы тоже можете внести свой вклад, присоединившись к русскоязычному сообществу MDN Web Docs.

const

Значение констант не может быть изменено новым присваиванием, а также не может быть переопределено. Константы (`const`) подчиняются области видимости уровня блока так же, как переменные, объявленные с использованием ключевого слова `let`.

Синтаксис

```
const name1 = value1 [, name2 = value2 [, ... [, nameN = valueN]]];
```

nameN

Имя константы. Подчиняется тем же правилам, что и [идентификаторы](#) обычных переменных.

valueN

Значение константы. Может быть любым из тех типов, которые могут быть для переменных, включая функцию.

Описание

Это объявление создаёт константу, чья область действия может быть как глобальной, так и локальной внутри блока, в котором она объявлена. Глобальные константы не становятся свойствами объекта `window`, в отличие от `var`-переменных. Инициализация константы обязательна; необходимо указать значение одновременно с объявлением (смысл в том, что потом это значение изменить уже нельзя).

`const` создаёт константу (новую именованную ссылку на область памяти), доступную только для чтения. Это не означает, что указываемое значение неизменно, но это означает, что идентификатор не может быть переназначен. Например, если константа указывает на объект, то сам объект может быть изменён.

Все соображения по поводу [временных мёртвых зон](#), применимы и к `let` и к `const`.

Имена констант не могут совпадать с именами функций или переменных той же области видимости.

Примеры

Ниже описаны примеры, демонстрирующие поведение `const`. Попробуйте их выполнить в [простом JavaScript-редакторе](#) или в консоли браузера.

JS

```
// Примечание: Идентификаторы констант могут быть объявлены как в верхнем,  
// так и в нижнем регистре. Но правилом хорошего тона является использование  
// верхнего регистра.
```

```
// определим MY_FAV как константу и присвоим ей значение 7  
const MY_FAV = 7;
```

```
// данное присваивание выдаст ошибку - Uncaught TypeError: Assignment to constant variable.  
MY_FAV = 20;
```



```
// попробуем переопределить константу, будет брошено исключение - Uncaught SyntaxError: Identifier 'MY_FAV' has already been declared
const MY_FAV = 20;

// имя MY_FAV зарезервировано константой выше, данная операция
// выкинет исключение
var MY_FAV = 20;

// здесь также будет ошибка
let MY_FAV = 20;

// Важно отметить свойства области видимости уровня блока
if (MY_FAV === 7) {
  // Всё нормально. Создать константу или переменную MY_FAV можно.
  // (работает так же как и let при объявлении в блоке переменных, которые не const)
  const MY_FAV = 20;

  // MY_FAV теперь 20
  console.log("my favorite number is " + MY_FAV);

  // это попадёт в глобальный контекст и выдаст ошибку
  var MY_FAV = 20;
}

// MY_FAV все ещё имеет значение 7
console.log("my favorite number is " + MY_FAV);

// Выдаст ошибку, потому что константа не инициализирована - Uncaught SyntaxError: Missing initializer in const declaration
const FOO;

// const также работает с объектами
const MY_OBJECT = {"key": "value"};

// Попытка переопределить ссылку на объект вызовет исключение - Uncaught TypeError: Assignment to constant variable.
MY_OBJECT = {"OTHER_KEY": "value"};

// но, свойства объекта (ключи) можно изменять
MY_OBJECT.key = "otherValue"; // Используйте Object.freeze() для того, чтобы сделать объект неизменяемым

// То же самое применимо к массивам
const MY_ARRAY = [];
// Например, можно добавлять элементы в массив
MY_ARRAY.push("A"); // ["A"]
// Но менять ссылку на объект массива нельзя. Это вызовет исключение - Uncaught TypeError: Assignment to constant variable
MY_ARRAY = ["B"]
```

Спецификация

Specification
ECMAScript Language Specification # sec-let-and-const-declarations

Поддержка браузерами

[Report problems with this compatibility data on GitHub](#)

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS
const	Chrome 21	Edge 12	Firefox 36	Opera 9	Safari 5.1	Chrome 25 Android	Firefox 36 for Android	Opera 10.1 Android	Safari on

Tip: you can click/tap on a cell for more information.

Full support See implementation notes.

Смотрите также

- [var](#)
- [let](#)
- [Константы в справочнике JavaScript](#)

This page was last modified on 4 авг. 2023 г. by [MDN contributors](#).