

Переменные

let - это новый способ объявления переменных, который был введен в ECMAScript 2015 (ES6).	const - это новый способ объявления переменных, который был введен в ECMAScript 2015 (ES6).	var - это старый способ объявления переменных, который был введен в первых версиях JavaScript.
Переменные, объявленные с помощью let, имеют блоковую область видимости, то есть они доступны только внутри блока, в котором они объявлены.	Переменные, объявленные с помощью const, имеют блоковую область видимости, то есть они доступны только внутри блока, в котором они объявлены.	Переменные, объявленные с помощью var, имеют функциональную область видимости, то есть они доступны во всей функции, в которой они объявлены.
Переменные, объявленные с помощью let, можно переопределить и изменить.	Переменные, объявленные с помощью const, не могут быть переопределены, но их свойства можно изменить.	Переменные, объявленные с помощью var, можно переопределить и изменить.
Переменные, объявленные с помощью let, не поднимаются вверх, то есть они не доступны до их объявления.	Переменные, объявленные с помощью const, не поднимаются вверх, то есть они не доступны до их объявления.	Переменные, объявленные с помощью var, поднимаются вверх, то есть они доступны до их объявления.

Переменная – это «именованное хранилище» для данных

Типы данных		Типы данных (cont)		Типы данных (cont)		Типы данных (cont)	
Примитивные типы данных	Ссылочные типы данных	<i>Boolean</i> - Представляет логическое значение: true или false.	<i>Function</i> - Функции в JavaScript также являются объектами, но они имеют дополнительные возможности, такие как вызов и выполнение кода.	<i>Null</i> - Представляет отсутствие значения. Это специальное значение, которое явно указывает, что переменная "пустая".	<i>Map, Set, WeakMap, WeakSet</i> - Эти типы данных были добавлены в ES6 для работы с коллекциями: Map : Коллекция пар "ключ-значение", где ключ может быть любого типа. Set : Коллекция уникальных значений. WeakMap и WeakSet : Аналогичны Map и Set, но используют слабые ссылки на объекты.	<i>BigInt</i> - Используется для представления целых чисел произвольной длины. Создается добавлением суффикса n к числу.	<i>Promise</i> - Используется для асинхронных операций.
<i>Number/Число</i> - Представляет числовые значения, включая целые числа и числа с плавающей точкой.	<i>Object</i> - Основной тип данных для хранения коллекций пар "ключ-значение". Объекты могут содержать свойства и методы.	<i>Undefined</i> - Значение, которое автоматически присваивается переменной, если она объявлена, но не инициализирована.	<i>Date</i> - Представляет дату и время. Является объектом.				
<i>String/Строка</i> - Представляет текстовые данные. Строки заключаются в одинарные ('), двойные (") или обратные кавычки (``).	<i>Array</i> - Массивы — это специальный тип объектов, предназначенный для хранения упорядоченных коллекций значений.						



Типы данных (cont)

Symbol - Введён в ECMAScript 6 (ES6). Представляет уникальное и неизменяемое значение, которое может использоваться как ключ объекта.

`typeof` : Оператор `typeof` используется для определения типа данных.
`typeof null` возвращает "object" (это историческая ошибка).
`typeof []` возвращает "object", так как массивы — это разновидность объектов.
`typeof function` возвращает "function", хотя функции технически являются объектами.



By **draganborn** (draganborn)

cheatography.com/draganborn/

Not published yet.

Last updated 26th February, 2025.

Page 3 of 2.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>