

Раздел

[Основы JavaScript](#)

Навигация по уроку

Переменная

Аналогия из жизни

Имена переменных

Константы

Придумывайте правильные имена

Итого

Задачи (3)

Комментарии

Поделиться

[Редактировать на GitHub](#)[🏠 → Язык программирования JavaScript](#)
[→ Основы JavaScript](#)

12-го сентября 2020

Переменные

JavaScript-приложению обычно нужно работать с информацией. Например:

1. Интернет-магазин – информация может включать продаваемые товары и корзину покупок.
2. Чат – информация может включать пользователей, сообщения и многое другое.

Переменные используются для хранения этой информации.

Переменная

Переменная – это «именованное хранилище» для данных. Мы можем использовать переменные для хранения товаров, посетителей и других данных.

Для создания переменной в JavaScript используйте ключевое слово `let`.

Приведённая ниже инструкция создаёт (другими словами: *объявляет* или *определяет*) переменную с именем «message»:

```
1 let message;
```

Теперь можно поместить в неё данные, используя оператор присваивания `=`:

```
1 let message;  
2  
3 message = 'Hello'; // сохранить строку
```

Строка сохраняется в области памяти, связанной с переменной. Мы можем получить к ней доступ, используя имя переменной:

```
1 let message;  
2 message = 'Hello!';  
3  
4 alert(message); // показывает содержимое переменной
```

Для краткости можно совместить объявление переменной и запись данных в одну строку:

```
1 let message = 'Hello!'; // определяем переменную и прис  
2  
3 alert(message); // Hello!
```

Мы также можем объявить несколько переменных в одной строке:

```
1 let user = 'John', age = 25, message = 'Hello';
```

Такой способ может показаться короче, но мы не рекомендуем его. Для лучшей читаемости объявляйте каждую переменную на новой строке.

Многострочный вариант немного длиннее, но легче для чтения:

```
1 let user = 'John';
```

Раздел

Основы JavaScript

Навигация по уроку

Переменная

Аналогия из жизни

Имена переменных

Константы

Придумывайте правильные имена

Итого

Задачи (3)

Комментарии

Поделиться



Редактировать на GitHub



Некоторые люди также определяют несколько переменных в таком вот многострочном стиле:

```
1 let user = 'John',
2   age = 25,
3   message = 'Hello';
```

...Или даже с запятой в начале строки:

```
1 let user = 'John'
2   , age = 25
3   , message = 'Hello';
```

В принципе, все эти варианты работают одинаково. Так что это вопрос личного вкуса и эстетики.

var вместо let

В старых скриптах вы также можете найти другое ключевое слово: `var` вместо `let`:

```
1 var message = 'Hello';
```

Ключевое слово `var` — почти то же самое, что и `let`. Оно объявляет переменную, но немного по-другому, «устаревшим» способом.

Есть тонкие различия между `let` и `var`, но они пока не имеют для нас значения. Мы подробно рассмотрим их в главе [Устаревшее ключевое слово "var"](#).

Аналогия из жизни

Мы легко поймём концепцию «переменной», если представим её в виде «коробки» для данных с уникальным названием на ней.

Например, переменную `message` можно представить как коробку с названием `"message"` и значением `"Hello!"` внутри:



Мы можем положить любое значение в коробку.

Мы также можем изменить его столько раз, сколько захотим:

```
1 let message;
2
3 message = 'Hello!';
4
5 message = 'World!'; // значение изменено
6
7 alert(message);
```

При изменении значения старые данные удаляются из переменной:



Раздел

Основы JavaScript

Навигация по уроку

Переменная

Аналогия из жизни

Имена переменных

Константы

Придумывайте правильные имена

Итого

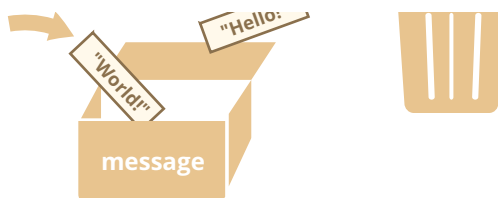
Задачи (3)

Комментарии

Поделиться



Редактировать на GitHub



Мы также можем объявить две переменные и скопировать данные из одной в другую.

```
1 let hello = 'Hello world!';
2
3 let message;
4
5 // копируем значение 'Hello world' из переменной hello
6 message = hello;
7
8 // теперь две переменные содержат одинаковые данные
9 alert(hello); // Hello world!
10 alert(message); // Hello world!
```

⚠ Повторное объявление вызывает ошибку

Переменная может быть объявлена только один раз.

Повторное объявление той же переменной является ошибкой:

```
1 let message = "Это";
2
3 // повторение ключевого слова 'let' приводит к ошибке
4 let message = "Другое"; // SyntaxError: 'message'
```

Поэтому следует объявлять переменную только один раз и затем использовать её уже без `let`.

ℹ Функциональные языки программирования

Примечательно, что существуют [функциональные](#) языки программирования, такие как [Scala](#) или [Erlang](#), которые запрещают изменять значение переменной.

В таких языках однажды сохранённое «в коробку» значение остаётся там навсегда. Если нам нужно сохранить что-то другое, язык заставляет нас создать новую коробку (объявить новую переменную). Мы не можем использовать старую переменную.

Хотя на первый взгляд это может показаться немного странным, эти языки вполне подходят для серьёзной разработки. Более того, есть такая область, как параллельные вычисления, где это ограничение даёт определённые преимущества. Изучение такого языка (даже если вы не планируете использовать его в ближайшее время) рекомендуется для расширения кругозора.

Имена переменных

В JavaScript есть два ограничения, касающиеся имён переменных:

1. Имя переменной должно содержать только буквы, цифры или символы `$` и `_`.
2. Первый символ не должен быть цифрой.

Примеры допустимых имён:

```
1 let userName;
```

```
2 let test123;
```

Если имя содержит несколько слов, обычно используется **верблюжья нотация**, то есть, слова следуют одно за другим, где каждое следующее слово начинается с заглавной буквы: `myVeryLongName`.

Самое интересное – знак доллара '\$' и подчёркивание '_' также можно использовать в названиях. Это обычные символы, как и буквы, без какого-либо особого значения.

Эти имена являются допустимыми:

```
1 let $ = 1; // объявили переменную с именем "$"
2 let _ = 2; // а теперь переменную с именем "_"
3
4 alert($ + _); // 3
```

Примеры неправильных имён переменных:

```
1 let 1a; // не может начинаться с цифры
2
3 let my-name; // дефис '-' не разрешён в имени
```

Регистр имеет значение

Переменные с именами `apple` и `AppLE` – это две разные переменные.

Нелатинские буквы разрешены, но не рекомендуются

Можно использовать любой язык, включая кириллицу или даже иероглифы, например:

```
1 let имя = '...';
2 let 我 = '...';
```

Технически здесь нет ошибки, такие имена разрешены, но есть международная традиция использовать английский язык в именах переменных. Даже если мы пишем небольшой скрипт, у него может быть долгая жизнь впереди. Людям из других стран, возможно, придётся прочесть его не один раз.

Зарезервированные имена

Существует **список зарезервированных слов**, которые нельзя использовать в качестве имён переменных, потому что они используются самим языком.

Например: `let`, `class`, `return` и `function` зарезервированы.

Приведённый ниже код даёт синтаксическую ошибку:

```
1 let let = 5; // нельзя назвать переменную "let", с
2 let return = 5; // также нельзя назвать переменную
```

Раздел

Основы JavaScript

Навигация по уроку

Переменная

Аналогия из жизни

Имена переменных

Константы

Придумывайте правильные имена

Итого

Задачи (3)

Комментарии

Поделиться



Редактировать на GitHub

Раздел

Основы JavaScript

Навигация по уроку

Переменная

Аналогия из жизни

Имена переменных

Константы

Придумывайте правильные имена

Итого

Задачи (3)

Комментарии

Поделиться



Редактировать на GitHub



⚠ Создание переменной без использования `use strict`

Обычно нам нужно определить переменную перед её использованием. Но в старые времена было технически возможно создать переменную простым присвоением значения без использования `let`. Это все ещё работает, если мы не включаем `use strict` в наших файлах, чтобы обеспечить совместимость со старыми скриптами.

```
1 // заметка: "use strict" в этом примере не используется
2
3 num = 5; // если переменная "num" раньше не существовала
4
5 alert(num); // 5
```

Это плохая практика, которая приводит к ошибке в строгом режиме:

```
1 "use strict";
2
3 num = 5; // ошибка: num is not defined
```

Константы

Чтобы объявить константную, то есть, неизменяемую переменную, используйте `const` вместо `let`:

```
1 const myBirthday = '18.04.1982';
```



Переменные, объявленные с помощью `const`, называются «константами». Их нельзя изменить. Попытка сделать это приведёт к ошибке:



```
1 const myBirthday = '18.04.1982';
2
3 myBirthday = '01.01.2001'; // ошибка, константу нельзя изменить
```

Если программист уверен, что переменная никогда не будет меняться, он может гарантировать это и наглядно донести до каждого, объявив её через `const`.

Константы в верхнем регистре

Широко распространена практика использования констант в качестве псевдонимов для трудно запоминаемых значений, которые известны до начала исполнения скрипта.

Названия таких констант пишутся с использованием заглавных букв и подчёркивания.

Например, сделаем константы для различных цветов в «шестнадцатеричном формате»:

```
1 const COLOR_RED = "#F00";
2 const COLOR_GREEN = "#0F0";
3 const COLOR_BLUE = "#00F";
4 const COLOR_ORANGE = "#FF7F00";
5
6 // ...когда нам нужно выбрать цвет
7 let color = COLOR_ORANGE;
8 alert(color); // #FF7F00
```

Преимущества:

Раздел

Основы JavaScript

Навигация по уроку

Переменная

Аналогия из жизни

Имена переменных

Константы

Придумывайте правильные имена

Итого

Задачи (3)

Комментарии

Поделиться



Редактировать на GitHub



- COLOR_ORANGE гораздо легче запомнить, чем "#FF7F00".
- Гораздо легче допустить ошибку при вводе "#FF7F00", чем при вводе COLOR_ORANGE.
- При чтении кода COLOR_ORANGE намного понятнее, чем #FF7F00.

Когда мы должны использовать для констант заглавные буквы, а когда называть их нормально? Давайте разберёмся и с этим.

Название «константа» просто означает, что значение переменной никогда не меняется. Но есть константы, которые известны до выполнения (например, шестнадцатеричное значение для красного цвета), а есть константы, которые *вычисляются* во время выполнения сценария, но не изменяются после их первоначального назначения.

Например:

```
1 const pageLoadTime = /* время, потраченное на загрузку
```

Значение pageLoadTime неизвестно до загрузки страницы, поэтому её имя записано обычными, а не прописными буквами. Но это всё ещё константа, потому что она не изменяется после назначения.

Другими словами, константы с именами, записанными заглавными буквами, используются только как псевдонимы для «жёстко закодированных» значений.

Придумывайте правильные имена

В разговоре о переменных необходимо упомянуть, что есть ещё одна чрезвычайно важная вещь.

Название переменной должно иметь ясный и понятный смысл, говорить о том, какие данные в ней хранятся.

Именованние переменных – это один из самых важных и сложных навыков в программировании. Быстрый взгляд на имена переменных может показать, какой код был написан новичком, а какой – опытным разработчиком.

В реальном проекте большая часть времени тратится на изменение и расширение существующей кодовой базы, а не на написание чего-то совершенно нового с нуля. Когда мы возвращаемся к коду после какого-то промежутка времени, гораздо легче найти информацию, которая хорошо размечена. Или, другими словами, когда переменные имеют хорошие имена.

Пожалуйста, потратьте время на обдумывание правильного имени переменной перед её объявлением. Делайте так, и будете вознаграждены.

Несколько хороших правил:

- Используйте легко читаемые имена, такие как userName или shoppingCart.
- Избегайте использования аббревиатур или коротких имён, таких как a, b, c, за исключением тех случаев, когда вы точно знаете, что так нужно.
- Делайте имена максимально описательными и лаконичными. Примеры плохих имён: data и value. Такие имена ничего не говорят. Их можно использовать только в том случае, если из контекста кода очевидно, какие данные хранит переменная.
- Договоритесь с вашей командой об используемых терминах. Если посетитель сайта называется «user», тогда мы должны называть связанные с ним переменные currentUser или newUser, а не, к примеру, currentVisitor или newManInTown.

Звучит просто? Действительно, это так, но на практике для создания описательных и кратких имён переменных зачастую требуется подумать. Действуйте.

Раздел

Основы JavaScript

Навигация по уроку

Переменная

Аналогия из жизни

Имена переменных

Константы

Придумывайте правильные имена

Итого

Задачи (3)

Комментарии

Поделиться



Редактировать на GitHub



Повторно использовать или создавать новую переменную?

И последняя заметка. Есть ленивые программисты, которые вместо объявления новых переменных повторно используют существующие.

В результате их переменные похожи на коробки, в которые люди бросают разные предметы, не меняя на них этикетки. Что сейчас находится внутри коробки? Кто знает? Нам необходимо подойти поближе и проверить.

Такие программисты немного экономят на объявлении переменных, но теряют в десять раз больше при отладке.

Дополнительная переменная – это добро, а не зло.

Современные JavaScript-минификаторы и браузеры оптимизируют код достаточно хорошо, поэтому он не создаёт проблем с производительностью. Использование разных переменных для разных значений может даже помочь движку оптимизировать ваш код.

Итого

Мы можем объявить переменные для хранения данных с помощью ключевых слов `var`, `let` или `const`.

- `let` – это современный способ объявления.
- `var` – это устаревший способ объявления. Обычно мы вообще не используем его, но мы рассмотрим тонкие отличия от `let` в главе [Устаревшее ключевое слово "var"](#) на случай, если это всё-таки вам понадобится.
- `const` – похоже на `let`, но значение переменной не может изменяться.

Переменные должны быть названы таким образом, чтобы мы могли легко понять, что у них внутри.

Задачи

Работа с переменными

важность: 2

1. Объявите две переменные: `admin` и `name`.
2. Запишите строку "Джон" в переменную `name`.
3. Скопируйте значение из переменной `name` в `admin`.
4. Выведите на экран значение `admin`, используя функцию `alert` (должна показать «Джон»).

решение

Придумайте правильные имена

важность: 3

1. Создайте переменную для названия нашей планеты. Как бы вы её назвали?
2. Создайте переменную для хранения имени текущего посетителя сайта. Как бы вы назвали такую переменную?

решение

Какие буквы (заглавные или строчные) использовать для имён констант?

важность: 4

Рассмотрим следующий код:

Раздел

[Основы JavaScript](#)

Навигация по уроку

Переменная

Аналогия из жизни

Имена переменных

Константы

Придумывайте правильные имена

Итого

Задачи (3)

Комментарии

Поделиться



[Редактировать на GitHub](#)



```
1 const birthday = '18.04.1982';
2
3 const age = someCode(birthday);
```

У нас есть константа `birthday`, а также `age`, которая вычисляется при помощи некоторого кода, используя значение из `birthday` (в данном случае детали не имеют значения, поэтому код не рассматривается).

Можно ли использовать заглавные буквы для имени `birthday`? А для `age`? Или одновременно для обеих переменных?

```
1 const BIRTHDAY = '18.04.1982'; // использовать заглавные
2
3 const AGE = someCode(BIRTHDAY); // а здесь?
```

решение

Проводим [курсы по JavaScript и фреймворкам](#).

Комментарии

перед тем как писать...

