## Структура кода

В большинстве редакторов строку кода можно закомментировать, нажав комбинацию клавиш Ctrl+/ для однострочного комментария и что-то вроде Ctrl+Shift+/ – для многострочных комментариев

## Переменные

**[Имена переменных](https://learn.javascript.ru/variables" \l "variable-naming)**

В JavaScript есть два ограничения, касающиеся имён переменных:

Имя переменной должно содержать только буквы, цифры или символы $ и \_.

Первый символ не должен быть цифрой.

**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:

let имя = '...';

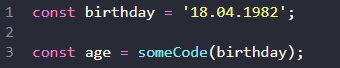
let 我 = '...';

Технически здесь нет ошибки, такие имена разрешены, но есть международная традиция использовать английский язык в именах переменных.

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.

**[Какие буквы (заглавные или строчные) использовать для имён констант?](https://learn.javascript.ru/variables" \l "kakie-bukvy-zaglavnye-ili-strochnye-ispolzovat-dlya-imyon-konstant)**



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

Обычно мы используем буквы в верхнем регистре для констант, которые «жёстко закодированы». Или, другими словами, когда значение известно до выполнения скрипта и записывается непосредственно в код.

В нашем примере, birthday именно такая переменная. Поэтому мы можем использовать заглавные буквы.

В отличие от предыдущей, переменная age вычисляется во время выполнения скрипта. Сегодня у нас один возраст, а через год уже совсем другой. Она является константой, потому что не изменяется при выполнении кода. Но она является «менее константной», чем birthday: она вычисляется, поэтому мы должны сохранить её в нижнем регистре.

## Типы данных

**Строка** (string) в JavaScript должна быть заключена в кавычки.

В JavaScript существует три типа кавычек.

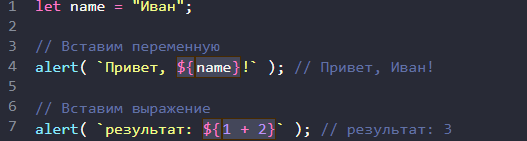
Двойные кавычки: "Привет".

Одинарные кавычки: 'Привет'.

Обратные кавычки: `Привет`.

Двойные или одинарные кавычки являются «простыми», между ними нет разницы в JavaScript.

Обратные же кавычки имеют расширенную функциональность. Они позволяют нам встраивать выражения в строку, заключая их в ${…}. Например:



Выражение внутри ${…} вычисляется, и его результат становится частью строки. Мы можем положить туда всё, что угодно: переменную name, или выражение 1 + 2, или что-то более сложное.

Обратите внимание, что это можно делать только в обратных кавычках. Другие кавычки не имеют такой функциональности встраивания!



## Логические операторы

Подсказка: передача пустого ввода в приглашение prompt возвращает пустую строку ''. Нажатие клавиши Esc во время запроса возвращает null.

## Функции

**[Выбор имени функции](https://learn.javascript.ru/function-basics" \l "function-naming)**

Например, функции, начинающиеся с "show" обычно что-то показывают.

Функции, начинающиеся с…

"get…" – возвращают значение,

"calc…" – что-то вычисляют,

"create…" – что-то создают,

"check…" – что-то проверяют и возвращают логическое значение, и т.д.

Примеры таких имён:

showMessage(..) // показывает сообщение

getAge(..) // возвращает возраст (получая его каким-то образом)

calcSum(..) // вычисляет сумму и возвращает результат

createForm(..) // создаёт форму (и обычно возвращает её)

checkPermission(..) // проверяет доступ, возвращая true/false