

# JavaScript: объекты

---

**WEB  
COURSE  
ORT DNIPRO**

---

**ORTDNIPRO.ORG/WEB**

# 1. Объект / Ассоциативный массив

# Объекты ?!



# Объекты в JavaScript

```
1
2
3  let person = {
4      name: 'Jhon',
5      lastName: 'Smith',
6      age: 28,
7      city: 'London',
8      canDrive: true
9  }
10
11
12  console.log( person.name, person.lastName );
13
14  person.country = 'UK';
15  person['phone'] = '+1234567890';
16
17
```

**Объекты в JavaScript** – структура данных (т.н. *ассоциативный массив*), в которой ключами (**индексами**) к ячейкам с данными выступают **строки**.

**Объекты в JavaScript** применяются повсеместно, **строки**, **массивы** и даже каждый **тег** в разметке – представляет собой **объект**.

## Объект - ссылочная структура

В переменных хранятся не сами **объекты** а **ссылки** на области памяти где объекты расположены, поэтому при «копировании» переменной присваивается **ссылка на объект**. И обе переменные позволяют работать с одним и тем же **объектом**.

## 2. Document Object Model

# DOM на практике

Lorem ipsum dolor sit amet consectetur adipisicing elit.  
Quam hic quae quisquam corporis quibusdam rerum libero.

Click Me!

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Officia delectus doloribus assumenda incidunt, quos, quaerat illo possimus, architecto enim odit nam! Distinctio consectetur non, rerum possimus officia, consequuntur esse iste sequi ab quisquam magnam quae voluptatibus illo ex amet dolores hic dolor nisi neque quia? Dolores officiis aperiam error nihil ad dolorum eligendi! Pariatur fuga nobis tempore eum obcaecati laborum dolore asperiores nemo cupiditate dolor quos fugiat excepturi, dolorem ipsum voluptatum ipsam eaque nam? Delectus dicta esse ipsam ab, aut voluptas sint labore aliquid rem, voluptatem beatae saepe repudiandae reprehenderit eius eligendi debitis minima quia quod veritatis. Aut architecto veniam laborum unde ullam similique voluptate provident, laboriosam corporis praesentium labore. Ex ab, asperiores voluptatum inventore, velit quam, omnis beatae eius pariatur quae quas quos blanditiis dicta illum. Dolore repellendus soluta animi doloremque tempore consequuntur, exercitationem debitis a architecto ut aliquam, fugit quisquam laudantium dicta voluptates nam officia, incidunt velit nesciunt quia maxime earum alias cum molestias? Architecto, consequatur optio. Consectetur quis iusto aut adipisci odio, iure nesciunt corporis illum quibusdam culpa error, cupiditate, tenetur sit doloribus esse amet laborum sed quia dolorum? Et atque accusantium soluta quibusdam quisquam rerum necessitatibus commodi eos! Saepe, nulla aspernatur pariatur dolorum dicta velit non.

Dolor eius, corporis repudiandae similique facere veritatis. Vero illo quisquam facilis vel molestiae voluptate natus consequatur deserunt omnis error ex, amet quo esse, nisi alias laborum eos temporibus nihil doloremque consectetur tempore ut. Pariatur nam, cupiditate natus ex non quam sapiente, similique quod dolor veritatis velit, iste mollitia. Perferendis, aliquam. Esse, corporis eum nisi, et eveniet sequi possimus deserunt porro eius, maiores natus debitis. Voluptate, deserunt reiciendis, harum iure illo ratione accusamus, dignissimos molestiae placeat dolores vitae autem voluptates laborum id expedita pariatur similique

Воспользуйтесь шаблоном в репозитории занятия:  
[./src/demo-template](#)

# DOM – Document Object Model

*(объектная модель документа)*

Стандарт который определяет из каких **объектов** браузер собирает **дерево документа**, и какие **свойства** есть у этих **объектов**. В соответствии со стандартом **DOM** каждый **тег HTML-документа**, в **JavaScript** представлен **объектом**.



# Чтобы управлять тегом его сначала нужно найти...

`document.querySelectorAll("css_selector")`

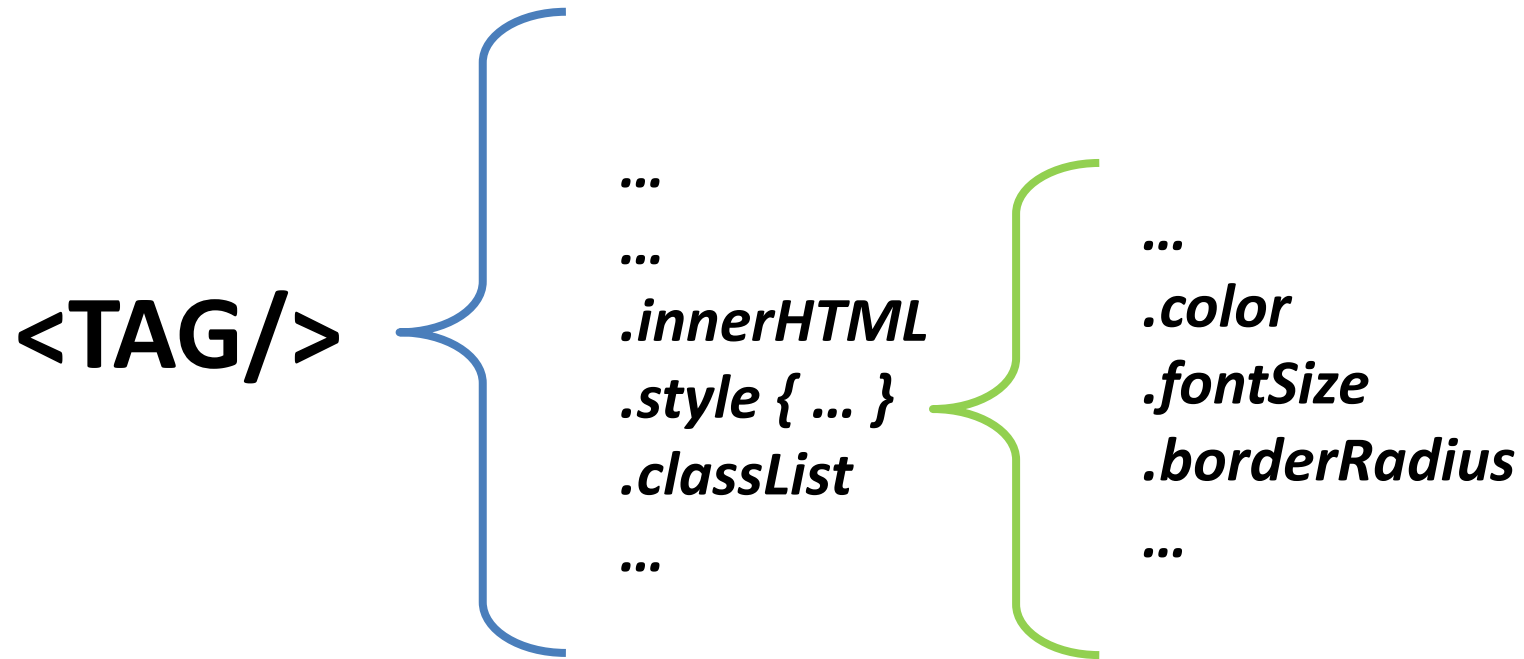
возвращает **псевдомассив объектов** (*тегов*) которые соответствуют **css-селектору** переданному в качестве **параметра** функции;

`document.querySelector("css_selector")`

возвращает первый найденный, в документе, **объект** (*тег*) соответствующий **css-селектору** переданному в качестве **параметра** функции;

**id** – элементы у которых есть атрибут **id** можно использовать без поиска, такие элементы доступны как **глобальные переменные** (с именем совпадающим с **id**).

## Из чего «сделан» тег?



Каждому **тег**, в JavaScript, **представлен объектом**, который хранит всё **содержимое**, все **стили** и все **атрибуты тега**. Разумеется их можно **менять**.

# Изменение содержимого элемента и/или его свойств

У тегов (элементов HTML-документа) есть ряд свойства определяющие его содержимое и внешний вид:

...

**.innerHTML** – свойство определяющее (или задающее) содержимое тега (его контент), т.е. всё то что находится между открывающимся и закрывающимся тегом;

**.style** – свойство определяющее объект со всеми поддерживаемыми браузером стилевые свойства;

**.classList** – свойство определяющее список классов тега (в виде массива, методы **.classList.add()** и **classList.remove()** позволяют добавлять и удалять классы тега). А метод **.classList.contains()** позволяет узнать есть ли класс в списке.

...

# 3. JavaScript и интерактивность

# События и интерактивность

**Функции** тесно связаны с обработкой событий, мы можем указать браузеру какую функцию необходимо вызывать, в случае наступления какого-либо события. События это не только клик мышкой, но и, например изменение данных в элементе ввода.

События **oninput** у элемента ввода отвечает за моменты когда осуществляется ввод данных в элемент ввода, т.е. в процессе ввода, например когда пользователь вводит новые символы.

**Отслеживание** этого события позволяет в динамике реагировать на действия пользователя.

## 4. Немного практики

# Элементы, события и интерактивность в JavaScript

Размер шрифта  px. Фоновый цвет

Lorem ipsum dolor sit amet consectetur, adipisicing elit.  
Atque, distinctio!

*Lorem ipsum dolor, sit amet consectetur adipisicing elit. Nobis accusantium quidem consectetur est tenetur reiciendis ab, voluptates repudiandae temporibus dolor totam fugit odit praesentium! Facere non, repudiandae ratione assumenda dicta quas repellat error itaque quidem ea adipisci veniam accusantium. Assumenda dolorum iure commodi fugiat sapiente praesentium harum? Beatae nam, eligendi minima mollitia, adipisci vitae totam eius cupiditate similique incidunt labore necessitatibus id! Qui, dicta aperiam. Explicabo veritatis nobis consectetur repellat eligendi eius necessitatibus quia corporis provident rem? Accusantium, quam exercitationem facere omnis ex excepturi nemo nulla temporibus vero enim obcaecati. Officia exercitationem dicta odio, necessitatibus illum cumque quas molestiae totam.*



Rem dignissimos voluptas voluptatibus atque tempore explicabo nihil ad deserunt neque aut possimus, sequi dolorem aspernatur nostrum ut! Placeat sed aspernatur libero, officiis optio repellat aperiam at, deserunt repellendus, cupiditate voluptatem corrupti quisquam quo. Distinctio, dignissimos esse? Recusandae earum voluptatibus, consectetur laboriosam aliquid possimus assumenda delectus ullam tempora obcaecati nam doloribus maxime tempore fugiat facere excepturi rerum dolor? Ipsa nesciunt labore sint deleniti error maiores voluptas, expedita ipsam ex sunt impedit, fuga enim, at vero quo incidunt blanditiis id recusandae

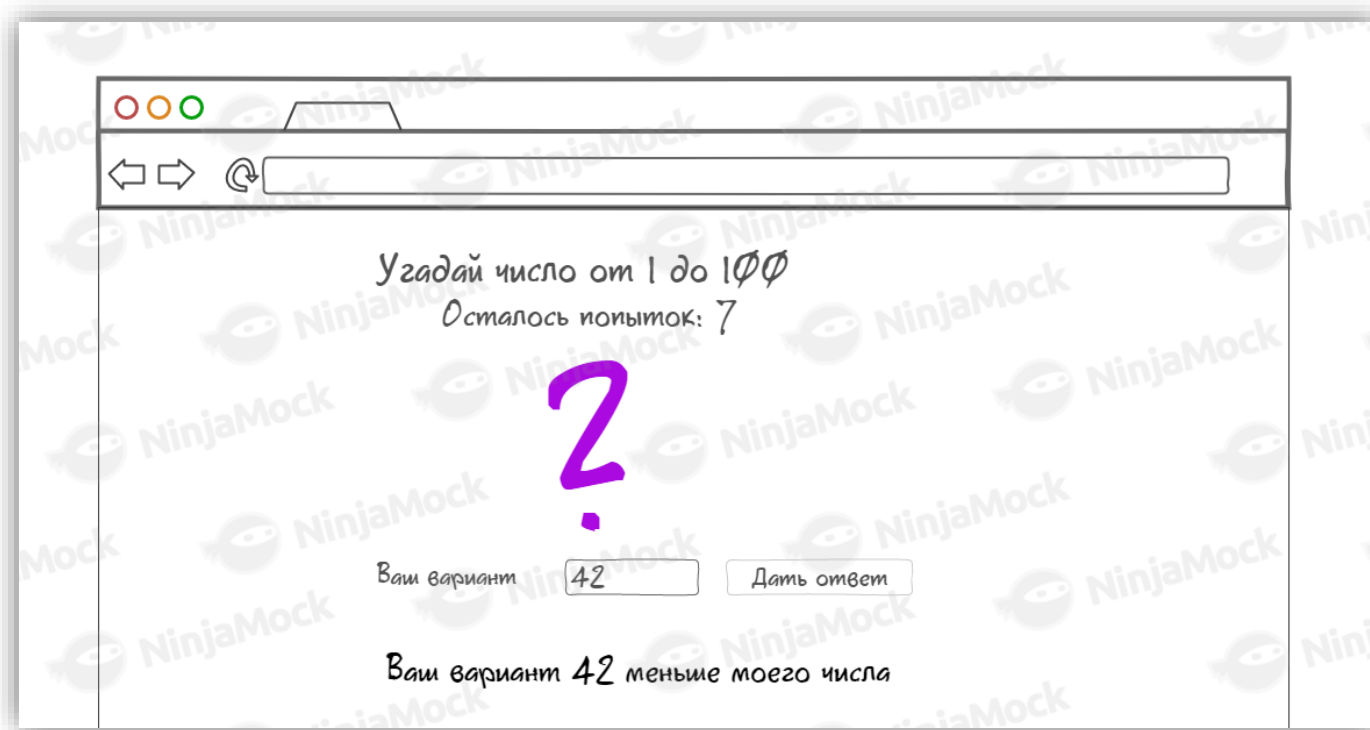
Воспользуйтесь шаблоном в  
репозитории занятия:  
[./src/accessibility-demo-template](#)

# Домашнее задание



# Игра «Угадай число», с разметкой

Скрипт **загадывает число** (от 1 до 100 включительно) и даёт игроку **10 попыток** его угадать, если пользователь не угадал – скрипт сообщает, что он проиграл и игра сообщает ему какой ответ был правильный. Если игрок угадал игра сообщает, что он выиграл. Для генерации чисел используйте функцию **Math.random()** Игра **при неправильных ответах** должна **давать подсказку** о том число которое ввёл пользователь больше или меньше загаданного.



Приведенный пример интерфейса может быть переделан вами по вашему усмотрению.