

JavaScript: объекты

**WEB
COURSE
ORT DNIPRO**

ORTDNIPRO.ORG/WEB

1. Объект / Ассоциативный массив

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 – 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**.

Подробнее: <https://learn.javascript.ru/dom-nodes>

Чтобы управлять тегом его сначала нужно найти...

`document.querySelectorAll("css_selector")`

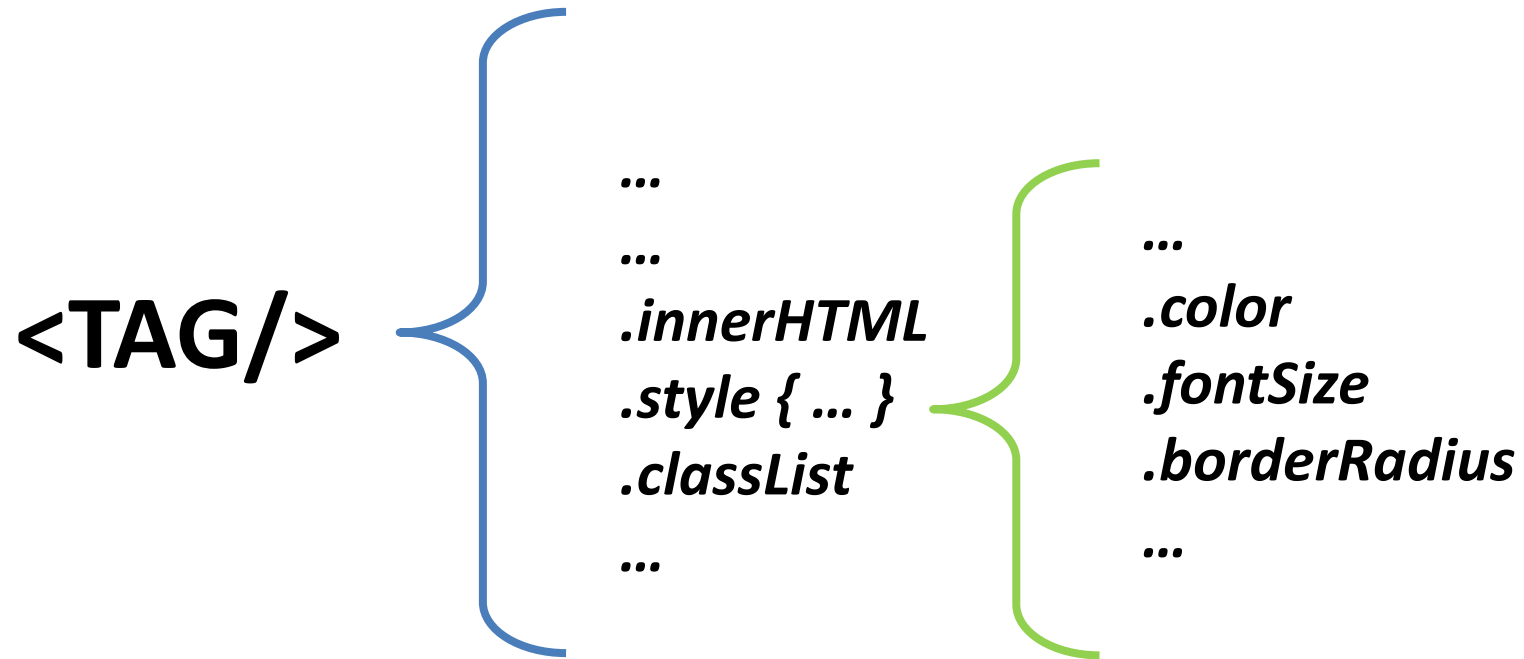
возвращает **псевдомассив объектов** (*тегов*) которые соответствуют **css-селектору** переданному в качестве **параметра** функции;

`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 #1

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](#)

5. Немного практики #2

Модальные окна / Тег <dialog>

Депозитный калькулятор

Открыть Депозитный Калькулятор

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Eum minus reiciendis expedita sed est natus dolorum vitae, magnam pariat. Sequi aliquid sapiente deserunt nemo dignissimos hic! Minus, ab quis officia illum magni quia, alias, repellat aliquam maiores necessitatibus natus. Nisi itaque culpa sit officia saepe. Unde voluptas tempore voluptate voluptates optio quam illo ipsam excepturi mollitia ipsa magnam iure itaque perferendis esse suscipit nulla dicta perspiciatis sed odit, maiores sunt! Pariatur explicabo consequuntur perferendis eum aliquid veritatis velit natus laboriosam eveniet, iste laborum? Molestias nesciunt excepturi, distinctio pariat itaque placeat sequi odio debitis, repellat vero est minima quos necessitatibus repellendus.

Quidem sint quisquam nisi voluptatum dolores explicat nesciunt, nam doloribus inv perspiciatis aperiam nemo obcaecati error distinctio voluptates totam officiis

Cupiditate fugit non volupt laborum facere quasi minus accusantium odit neque, ver adipisci ad consectetur per soluta deleniti ratione ill inventore. Asperiores unde excepturi?

Repudiandae dignissimos, cupiditate eaque delectus adipisci laboriosam? Facilis corporis dicta saepe tempore inventore quasi maiores provident, vitae ducimus id nam iste quae pariat illud. Ipsum, doloribus necessitatibus libero a iusto accusantium laborum eveniet impedit odit nobis sint et optio officiis quibusdam magni omnis suscipit? Similique repellat nostrum obcaecati nobis. Repudiandae nihil alias, aliquam voluptates assumenda totam quo in odio? Voluptates facere veniam accusamus? Earum sit obcaecati error quia eius deserunt necessitatibus quidem, laborum, totam non molestiae quas quasi vitae, architecto ratione. Eaue vel cum, laboriosam ullam quam dignissimos est asperiores harum, error dicta autem atque, nulla culpa aperiam odio corporis.

Dicta tempore neque dolorum maiores eos ratione molestiae doloremque, magnam voluptatibus natus dignissimos aut tenetur rem voluptate ducimus quaerat et officia laudantium aliquam impedit quisquam optio. Voluptatum quaerat sint voluptatem tempora rem rerum, officia consequuntur illo fugiat animi perspiciatis exercitationem minus necessitatibus alias veniam? Culpa quo et asperiores repudiandae sapiente non labore saepe soluta nam. Provident corrupti culpa voluptate inventore quae architecto non sapiente facilis deserunt illo neque repellat natus, laborum aliquam voluptatem maiores ab dolorum beatae. Doloremque perspiciatis sapiente impedit exercitationem quidem. Facilis, expedita cupiditate excepturi voluptatem suscipit tempore commodi velit tempora iure minima doloremque placeat veritatis dolorem error.

×

Сумма вклада (грн.)	Срок вклада (мес.)	Ставка (% годовых)	Капитализация %
<input type="text" value="10000"/>	<input type="text" value="12"/>	<input type="text" value="12.00"/>	<input type="checkbox"/>
<div>Рассчитать</div>			
По завершению депозита вы получите: - грн.			

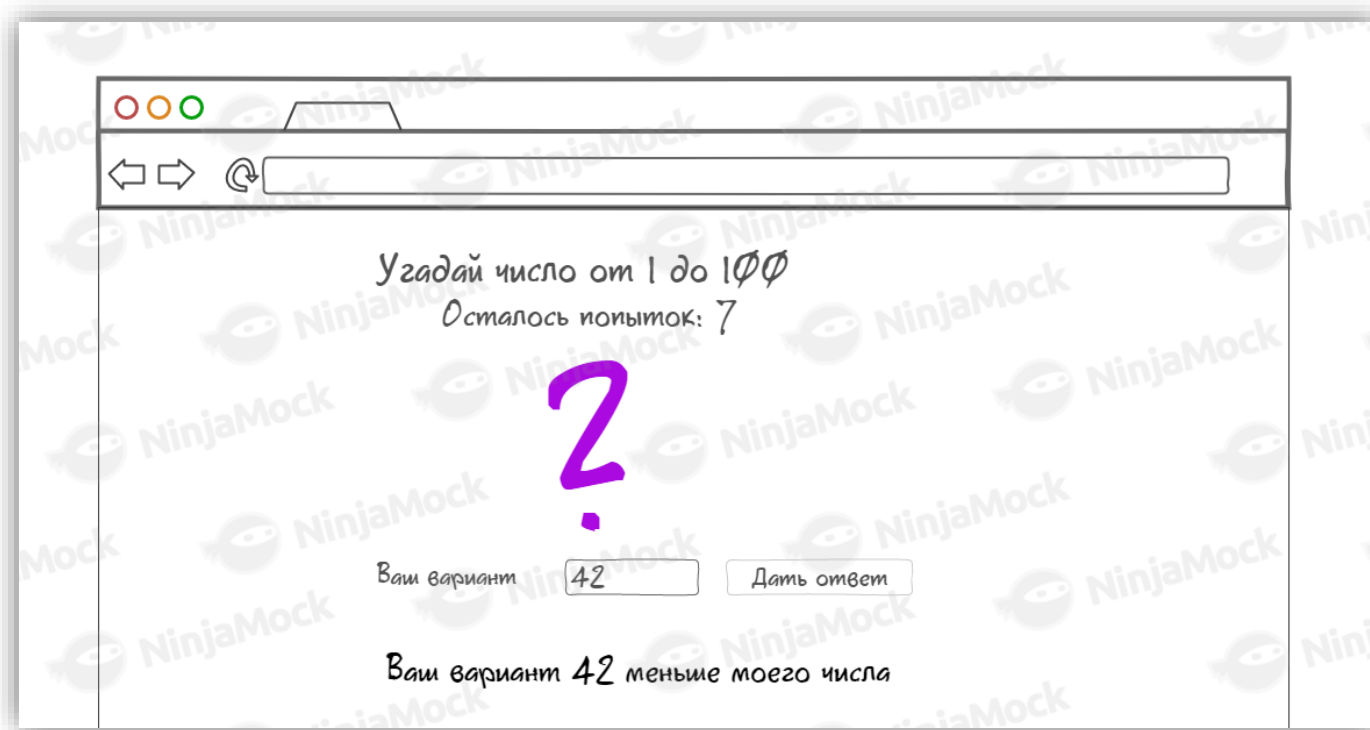
Воспользуйтесь шаблоном в репозитории занятия:

[./src/dialog-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.