

# JavaScript: Document Object Model

---

WEB  
COURSE  
ORT DNIPRO

---

ORTDNIPRO.ORG/WEB

# 1. 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")`

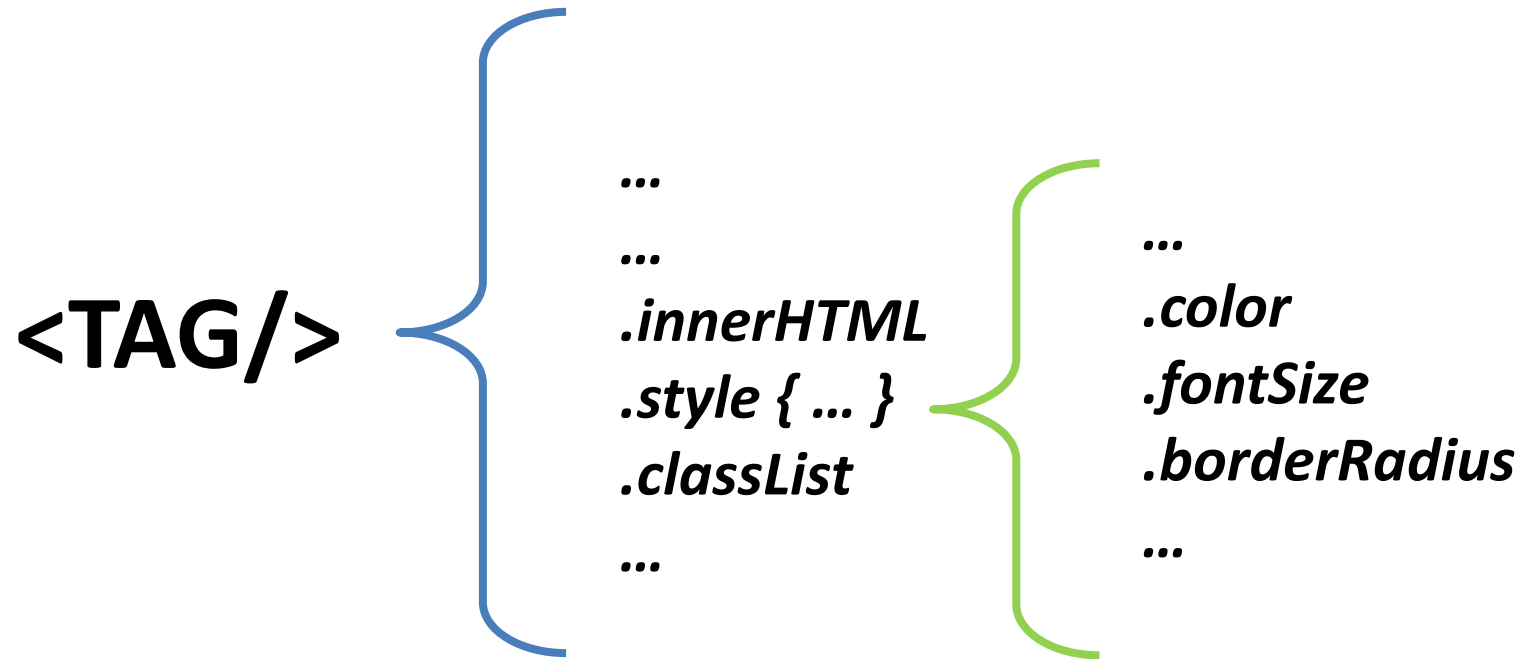
возвращает **псевдомассив объектов** (*тегов*) которые соответствуют **css-селектору** переданному в качестве **параметра** функции;

`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 объектов (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.

...

## 2. Немного практики



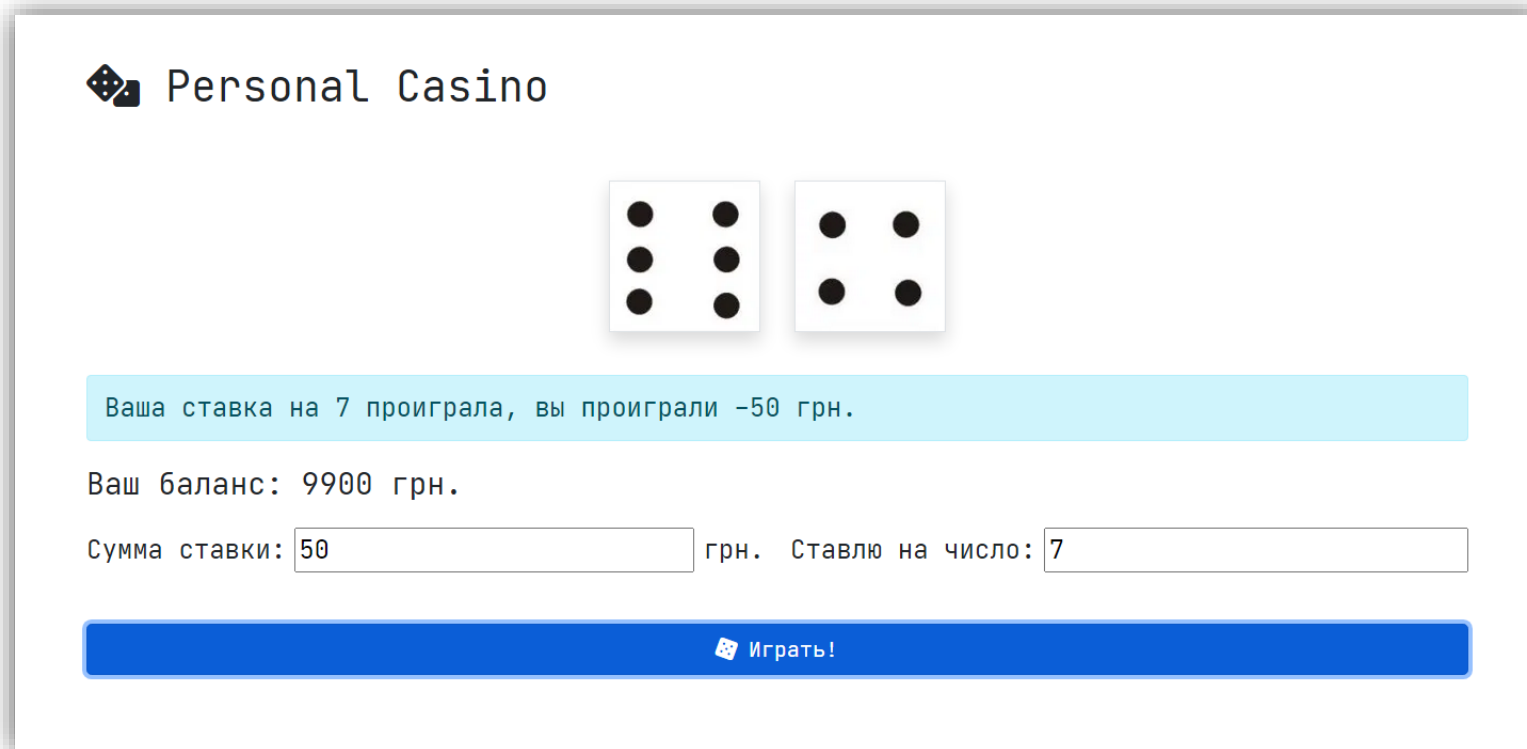
## Игра в кости



Игроку при старте даётся **10000 гривен**. У игрока спрашивается какую **сумму** он **ставит** (только целые числа), и на какой **результат** (от 2 до 12 включительно). После этого компьютер «**бросает кости**» генерирует два числа от 1 до 6 включительно. Если сумма чисел совпала с загаданным числом пользователя он **получает удвоенную ставку**. О результатах каждого «бросания» необходимо **уведомлять пользователя** и о сумме его выигрыша или проигрыша. Игра продолжается до тех пор пока пользователь не проиграет все деньги.



# Управление документом и игровой процесс



Personal Casino

Ваша ставка на 7 проиграла, вы проиграли -50 грн.

Ваш баланс: 9900 грн.

Сумма ставки: 50 грн. Ставлю на число: 7

Играть!

*Реализуем «Игру в кости»*

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

# Цикл разработки сайта

# Маркетинговый инструмент



## Сайт



## Программный продукт

# Прежде всего!

## Какой у вас бюджет?

**Бюджет определяет всё:** какие инструменты и технологии будут применяться, каких специалистов необходимо привлечь и т.п.

# 1. Маркетинговое планирование

Какую задачу мы решаем?

Каковы цели и задачи?

Зачем вам сайт?

## 2. Техническое проектирование

~~Техническое задание~~

**Требования / Requirements**

1. Структура и функциональность разделов сайта;
2. Применяемые технологии и инструменты;
3. Планирование рабочего времени.

**Кто пишет «ТЗ» и нужно ли его писать?**

**Как быть с переменами у заказчика?**

# 3. Дизайн

Дизайн ≠ «Наведение красоты»

**Вайрфреймы, Мокапы, Макеты...**

Проектирование структуры и внешнего вида страниц сайта

Соответствует ли дизайн фирменному стилю?

*У вас ведь есть фирменный стиль?*

## 4.1 Верстка

Тестирование в различных браузерах

**Набор шаблонов страниц**

**Адаптивность**

**SEO/SMO оптимизация**



## 4.2 Программирование

# Реализация функционала компонентов сайта

# 5. Наполнение

## Составление текстов

Согласование текстов с заказчиком

Составление МЕТА-данных (SEO/SMO)

# Контент

**Контент – будет продавать!**

## 6. Выгрузка сайта и тестирование

**А хостинг? А домен?**

**Что нужно проверить проверить?**

- *Во всех ли современных браузерах работает сайт?*
- *Все ли необходимые **материалы размещены?***
- *Все ли программные компоненты работают слаженно и четко?*
- *Нет ли **битых ссылок** и/или пустых страниц?*
- *Доступна ли **карта сайта** и др. файлы (sitemap.xml, robots.txt)*

# Поддержка и обслуживание

# 7.1. Продление домена, хостинга

*(кто должен этим заниматься?)*

- непрерывная работа сайта;
- репутация заказчика и разработчика;
- контекстная реклама;
- потеря домена;

...

**Сопровождение сайта!**

**Завершение  
работы сайта**

## 8. Завершение работы сайта

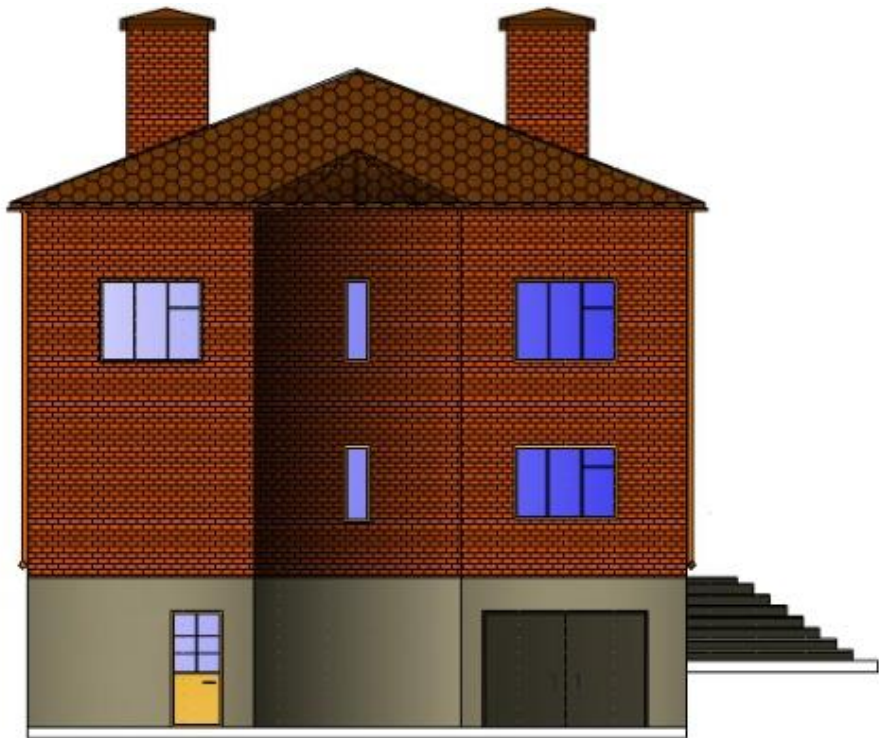
**Чистка истории / Ликвидация сайта**

**Хороший сайт отличается  
от плохого  
проработкой деталей**

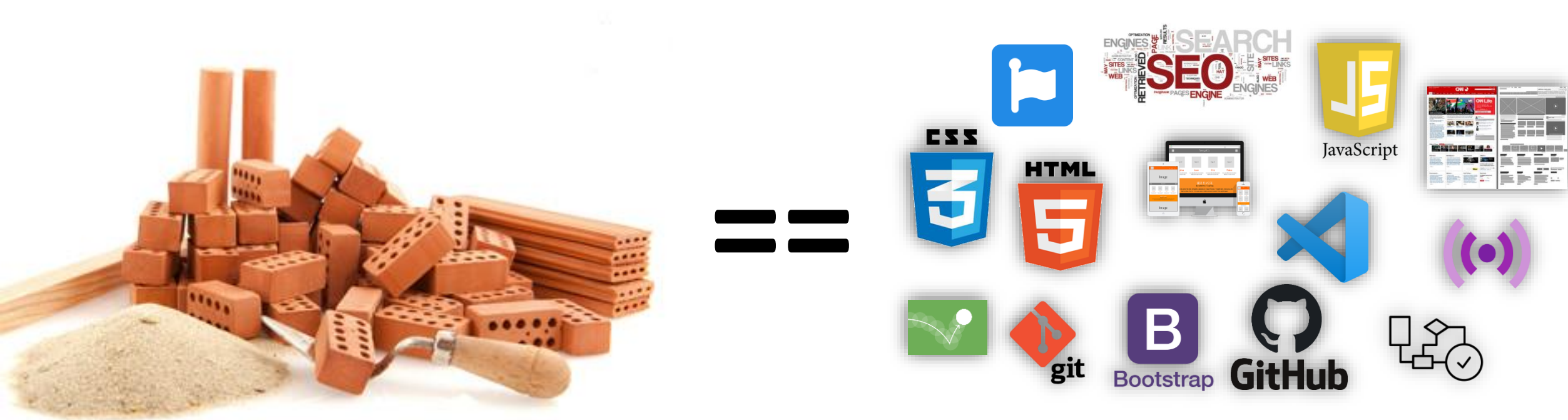


**Время подводить  
итоги**

# Задача специалиста решить задачу имеющимися инструментами



**Задача специалиста, любой профессии,  
решить задачу имеющимися инструментами**



Что дальше?

# 1. Совершенствовать навыки и умения



<http://prometheus.org.ua/>

<https://tproger.ru/>

<https://habrahabr.ru/>

<https://dou.ua/>



## 2. Практиковать



Например заняться **волонтерством**, применять знания в работе, в своём бизнесе.

### 3. Поддерживать связь



**А кого не испугало  
программировании то...**



---

# JS

COURSE  
ORT DNIPRO

---