

# 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")`

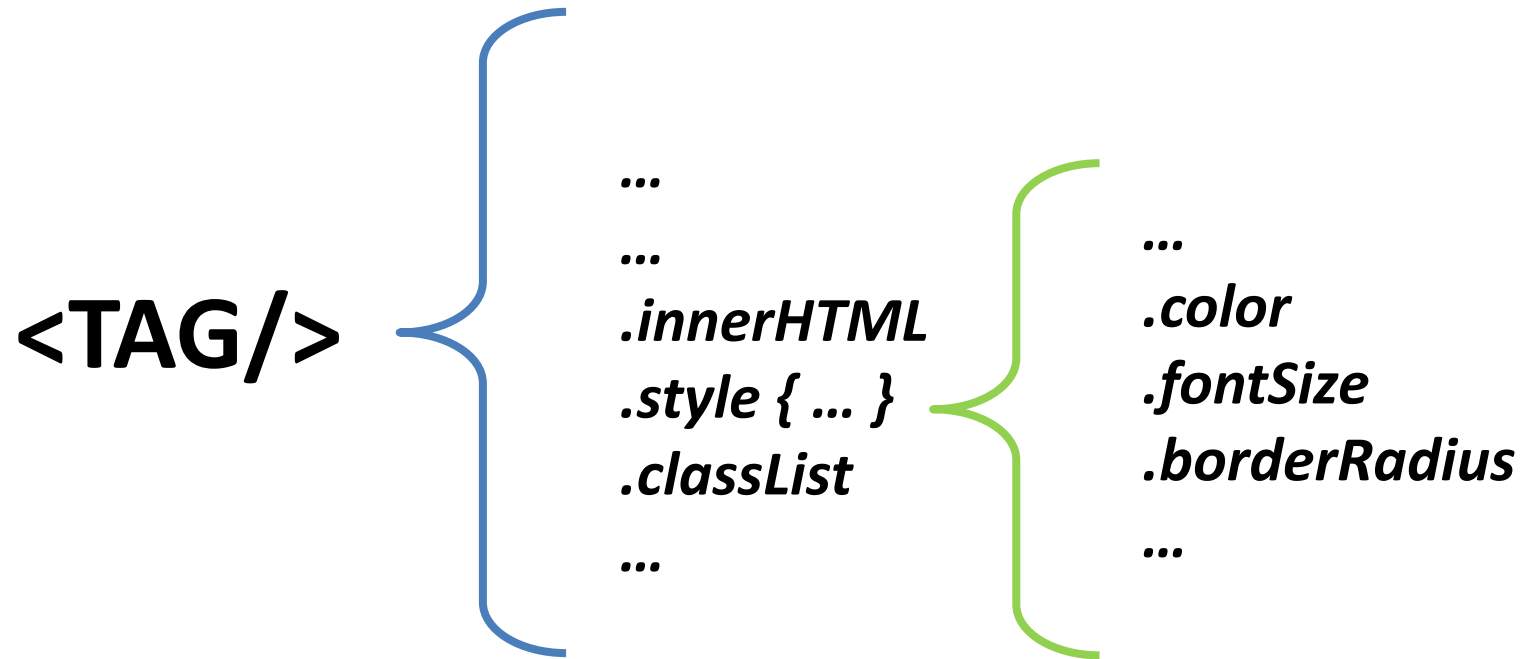
возвращает **псевдомассив объектов** (*тегов*) которые соответствуют **css-селектору** переданному в качестве **параметра** функции;

`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 #1



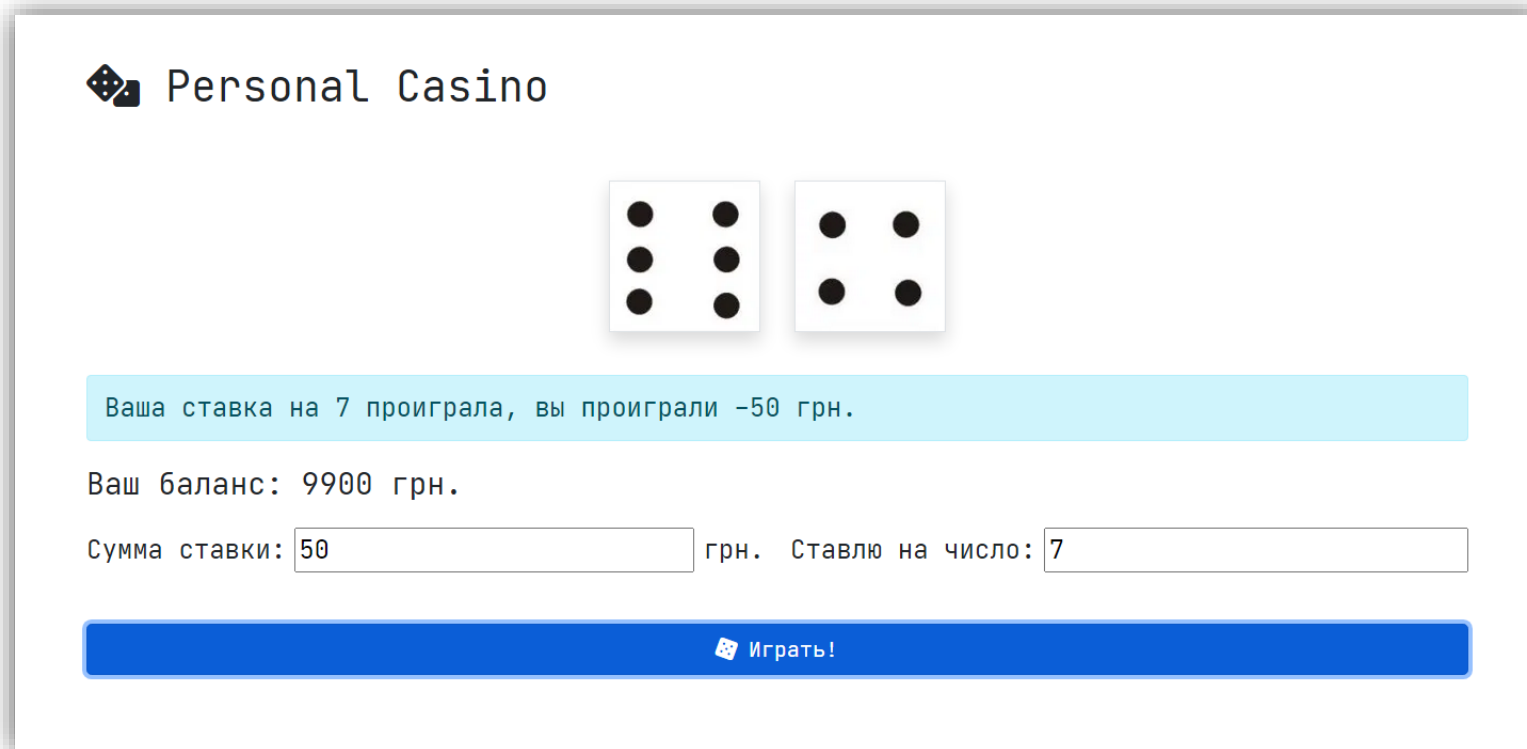
## Игра в кости




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**, если при этом оба выпавшие числа равны между собой то пользователь **получает утроенную 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:  грн. Ставлю на число:

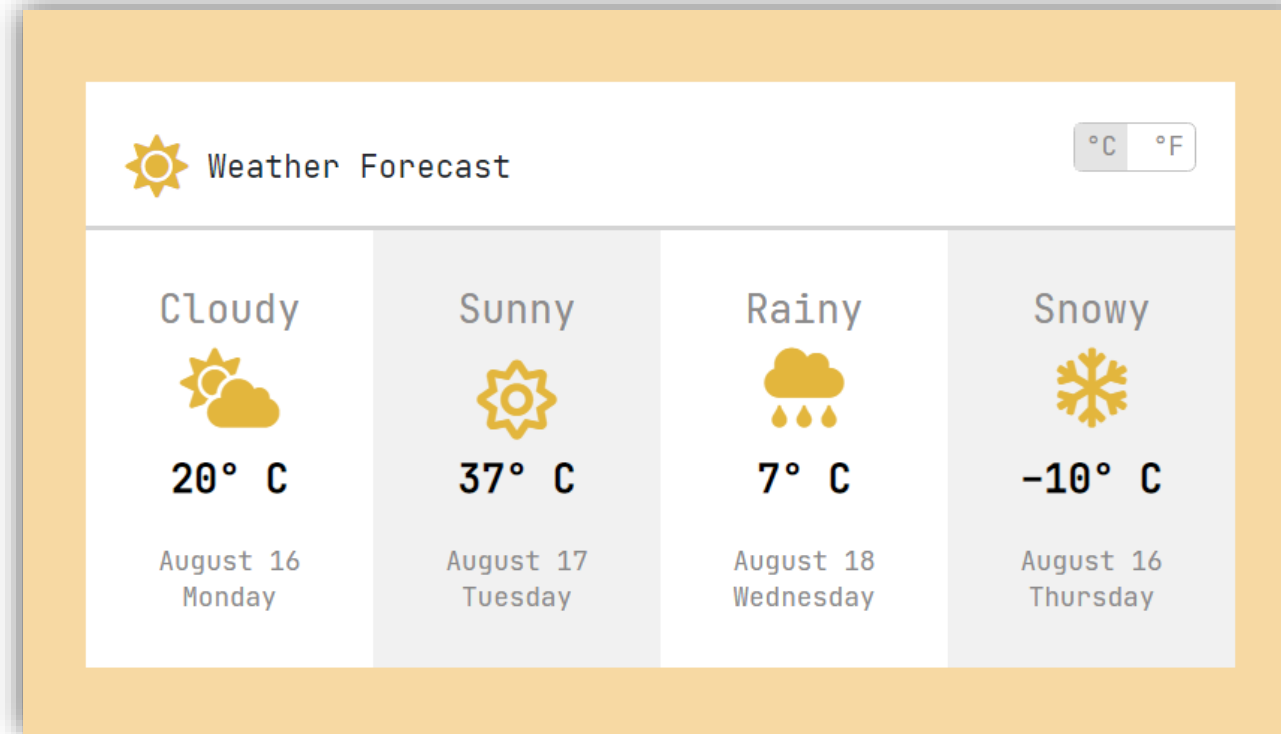
Играть!

*Реализуем «Игру в кости»*

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

## **3. Немного практики #2**

# Преобразование данных в разметке

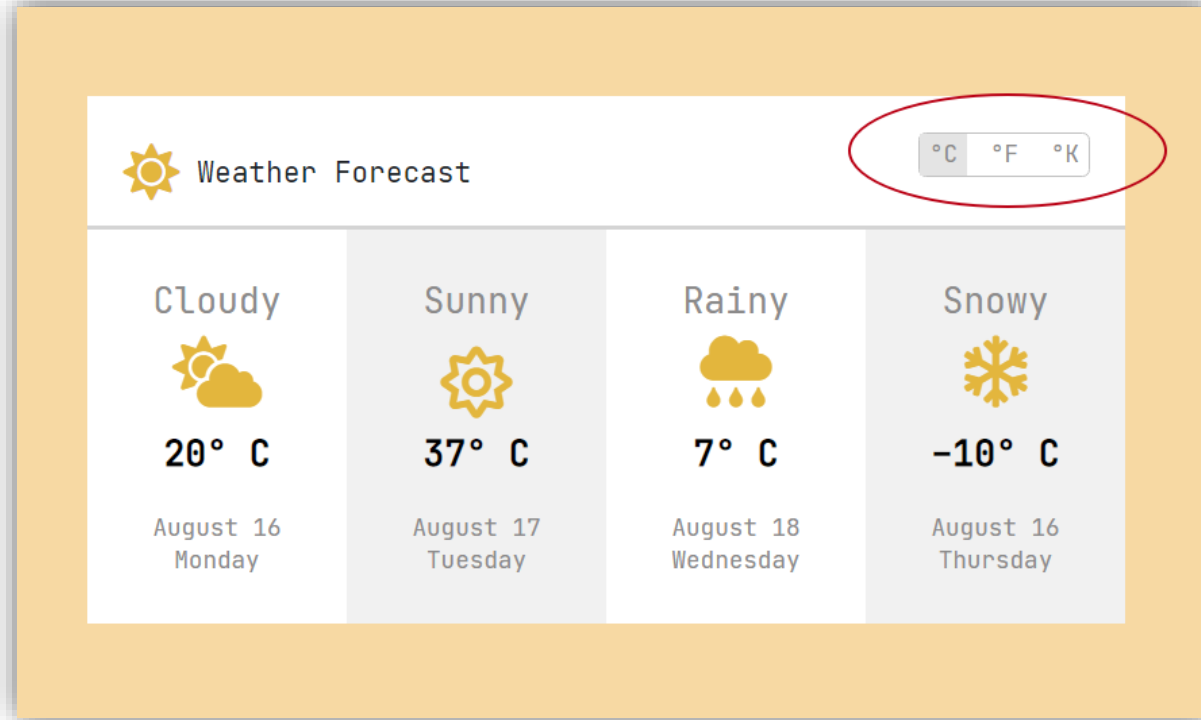


*Реализуем «Погодный виджет»*

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

# Домашнее задание

## Реализуйте переключение между **тремя шкалами** (Цельсию, Фаренгейту и Кельвину)



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