ДЕТСКАЯ ОНЛАЙН-ШКОЛА ПРОГРАММИРОВАНИЯ
HELLO WORLD

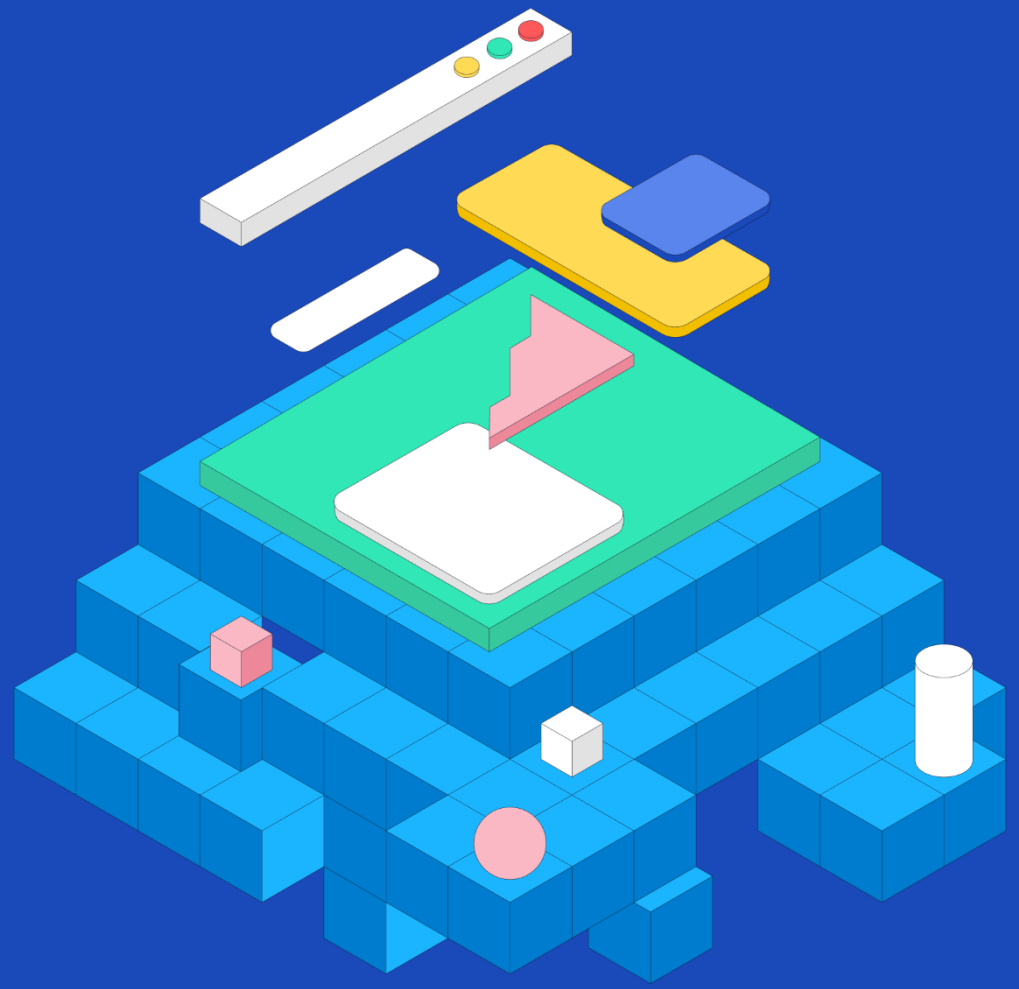
#ШПАРГАЛОЧКИ

СОЗДАНИЕ САЙТОВ FRONT-END РАЗРАБОТКА

Средний уровень

Материалы подготовлены отделом методической разработки

Больше полезных материалов и общения в нашем комьюнити в Telegram: https://t.me/hw_school



ООО «Онлайн-школа программирования» ОГРН 1186952015645 Лицензия №10/21-69 hwschool.online

Найди пару

calc

calc - **CSS-функция**, позволяющая выполнить какие-то вычисления для значений свойств. Например, можно задать размеры в процентах и вычесть из них сколько-то пикселей:

```
.element {
        width: calc(100% - 20px);
}
```

Псевдокласс active

active - псевдокласс, который отвечает за состояние элемента в момент нажатия на него. Может пригодиться, например, для создания анимации:

```
element:active {

    /* стили при нажатии */

}
```

backface-visibility

backface-visibility - свойство, определяющее, будет ли видна задняя часть элемента после его переворота. **backface-visibility** принимает 2 значения: **visible** и **hidden**.

Это свойство нужно при работе с 3D-трансформациями.

```
.element {
        backface-visibility: hidden;
}
```

transform-style

Свойство **transform-style** отвечает за стиль анимации (она может происходить как в двумерной, так и в трехмерной плоскости):

```
transform-style: preserve-3d; /* переходим в 3D */
```

У этого свойства есть и второе значение - **flat**. Это значение по умолчанию, при его использовании элемент остается в той же плоскости, что и родитель.

perspective

perspective - свойство, устанавливающее перспективу, то есть расстояние от плоскости экрана до центра настраиваемого элемента (на самом деле не всегда именно до центра, но по умолчанию - так).

Принимает расстояние, чем оно больше, тем “дальше” элемент окажется от экрана:

perspective: 380px;

data-атрибуты

data-атрибуты - это специальные атрибуты для тегов, которым можно задавать собственные имена. Общая конструкция - **data-***, где вместо * может стоять любое имя:

```
<div class="first" data-number="1"></div>

<div class="second" data-number="1"></div>
```

data-атрибуты можно использовать в JavaScript и CSS. Чтобы получить к ним доступ в JavaScript, потребуется обратиться к свойству **dataset** элемента:

```
element = event.target.parentElement;

console.log(element.dataset.number);
```

addEventListener и removeEventListener

Основной недостаток событий вроде onclick - одному элементу нельзя назначить 2 события, так как второе перезапишет первое. Чтобы решить эту проблему, нужно использовать **слушатель событий**. Он добавляется через метод **addEventListener**, принимает событие и функцию, которая должна сработать:

document.querySelector(“button”).addEventListener(“click”, func);

Тут по клику на кнопку сработает функция **func**.

А чтобы удалить слушатель, когда он больше не нужен, есть метод **removeEventListener**:

document.querySelector(“button”).removeEventListener(“click”, func);

null и undefined

null и **undefined** - 2 значения, которые могут обозначать отсутствие какого-либо значения (например, в переменной).

Разница между ними в том, что **null присваивается** программистом, то есть мы явно говорим, что в переменной ничего не хранится.

А **undefined** - это обозначение незаданного значения. Оно значит, что переменная **не определена**. Например, в нее просто ничего не записали.