



#ШПАРГАЛОЧКИ

# СОЗДАНИЕ САЙТОВ FRONT-END РАЗРАБОТКА

## Начальный уровень

Материалы подготовлены отделом методической разработки

Больше полезных материалов и общения в нашем комьюнити  
в Telegram: [https://t.me/hw\\_school](https://t.me/hw_school)





# Создание сайта My Games Hub

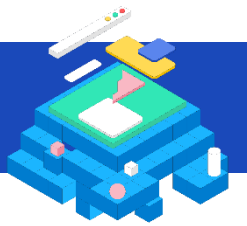




# transform

**transform** - CSS свойство, позволяющее подвинуть или наклонить элемент, а также изменить его масштаб. Принимает следующие значения:

- **translate(сдвиг\_по\_x, сдвиг\_по\_y)** - передвигает элемент, также можно использовать отдельные значения для сдвига по горизонтали и вертикали (**translateX()**, **translateY()**);
- **scale()** - изменяет масштаб элемента. Например, **scale(0.5)** уменьшит элемент в 2 раза. Если нужно изменить масштаб только по горизонтали или вертикали, можно использовать значения **scaleX()** и **scaleY()**.



# transform

**rotate()** - поворачивает элемент по часовой стрелке на заданное число градусов. Например, **rotate(45deg)** повернет элемент на 45 градусов. Чтобы элемент поворачивался против часовой стрелки, число должно быть отрицательным - **rotate(-90deg)**.

**skew()** - искажает стороны элемента относительно координатных осей. Принимает угол искажения по X и Y - **skew(10deg, 25deg)**. Также можно задать искажение отдельно для X (**skewX()**) и Y (**skewY()**).

Чтобы задать элементу сразу несколько трансформаций, их нужно указать в одном свойстве:

**transform: scale(2) skew(15deg) rotate(30deg) translate(100px 50px);**

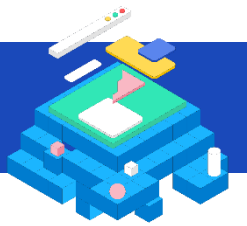


# Псевдоклассы

Помимо обычных классов, создаваемых в **html**, есть еще и **псевдоклассы**. Они создаются в **CSS** и служат для работы с элементами в каком-то состоянии. Например, для изменений стилей элемента при наведении на него курсора.

**:hover** - это **псевдокласс**, позволяющий изменить стили элемента, когда на него наведен курсор. А когда курсор будет убран с элемента, ему вернутся прежние стили.

```
a:hover {  
    color: green; /* при наведении на ссылку ее текст станет зеленым */  
}
```



# Задержка

**transition** - свойство, позволяющее настроить анимацию, в частности, ее время и задержку перед началом. transition объединяет в себе 4 свойства:

**transition-delay** (задержка перед анимацией, указывается в секундах - **s** или миллисекундах - **ms**)

**transition-duration** (сколько будет длиться анимация)

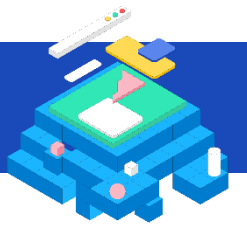
**transition-property** (свойство, к которому применится transition - например, transform или margin)

**transition-timing-function** (равномерность анимации)

Все эти значения (или часть из них) можно указать в одном свойстве transition:

**transition: 2s; /\* время анимации - 2 секунды \*/**

**transition: 5s 1s transform; /\* трансформация начнется с задержкой 1 секунду и продлится 5 секунд \*/**



# Анимация и keyframes

Чтобы создать анимацию, которую можно запустить в любом месте кода, нужно использовать ключевое слово **@keyframes** (принято писать его в конце CSS файла), а после него - имя анимации.

```
@keyframes pulsar {  
  
}
```

В фигурных скобках нужно указать так называемые **ключевые кадры** анимации (как минимум - первый и последний). Их можно обозначить словами **from** и **to**, но куда удобнее использовать проценты. **Начало** анимации обозначается как **0%**, **конец** - как **100%**:

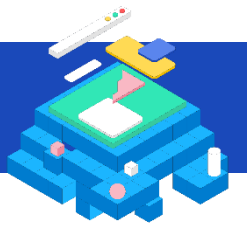


# Анимация и keyframes

Пример анимации, в которой элемент будет “пульсировать” - становиться то прозрачным, то нет.

```
@keyframes pulsar {  
  0% {  
    opacity: 1;  
  }  
  100% {  
    opacity: 0;  
  }  
}
```





# Анимация и keyframes

Чтобы применить созданную анимацию к элементу, в его свойствах нужно указать имя анимации, ее продолжительность и число повторов:

```
.my_class {
```

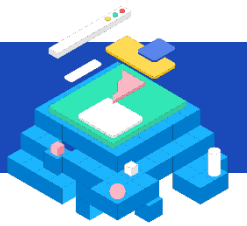
```
    animation-name: pulsar;
```

```
    animation-duration: 10s;
```

```
    animation-iteration-count: 10;
```

```
    /* повторится 10 раз, можно использовать infinite для бесконечной анимации */
```

```
}
```



# Другие свойства анимации

**animation-delay** - задержка анимации (в секундах или миллисекундах)

**animation-timing-functon** - равномерность анимации. Принимает следующие значения:

**linear** - равномерная анимация

**easy** - анимация начинается медленно, ускоряется, к концу снова замедляется

**easy-in-out** - то же, что и easy, но ускорение в начале быстрее

**easy-in** - анимация начинается медленно, затем ускоряется

**easy-out** - анимация начинается быстро, затем замедляется

**steps** - анимация резкая, по числу указанных в скобках шагов - **steps(10, end)**

**step-start** - стили сразу принимают конечное значение

**step-end** - стили находятся в начальном значении все время анимации, а потом переходят в конечное



# Фильтры

**filter** - CSS-свойство, добавляющее элементу разные графические эффекты. Принимает следующие значения (можно указать больше одного):

- **blur** - размытие, указывается в % или других единицах измерения;
- **brightness** - яркость, указывается в % или дробным числом. 0 - черный цвет, 100% или 1 - исходное состояние;
- **contrast** - контрастность, 0% - черный цвет, 100% - исходное состояние;
- **drop-shadow** - тень элемента, принимает отступы по X и Y, размытие, растяжение и цвет;
- **grayscale** - 0% - исходное состояние, 100% - все цвета переводятся в оттенки серого ;
- **hue-rotate** - принимает угол поворота по цветовому кругу и меняет цвет;
- **invert** - инвертирует цвета, принимает значения от 0% (исходное состояние) до 100% (полностью инвертированные цвета);
- **opacity** - непрозрачность, 0% - полностью прозрачный элемент, 100% - полностью непрозрачный;
- **saturate** - насыщенность цвета, 100% - исходное состояние, 0% - черно-белое изображение;
- **sepia** - превращает изображение в сепию, 0% - исходное состояние, 100% - сепия.



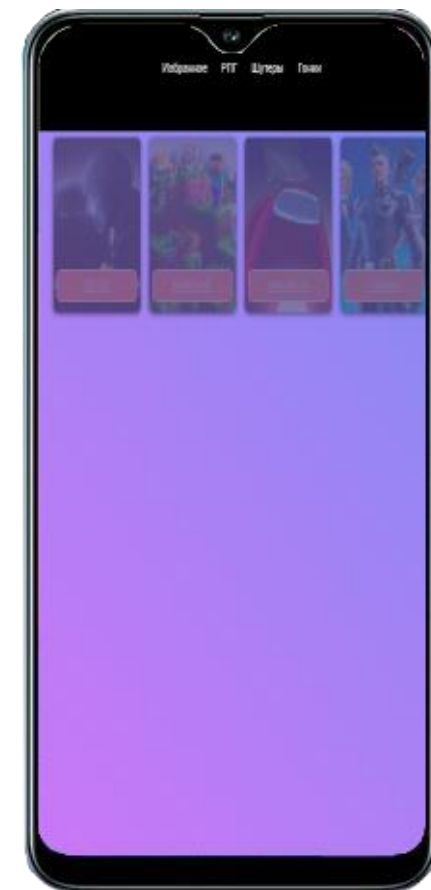
# Viewport

По умолчанию на всех устройствах браузер пытается открыть сайт так, чтобы он был виден полностью. Но тогда пользователи будут видеть слишком маленькие буквы и картинки, как на примере 1. Для того, чтобы пользователю было удобно просматривать страницу, её необходимо представлять в натуральном масштабе. В таком случае, пользователь будет видеть только определенную область, как представлено на примере 2, это и называется viewport.

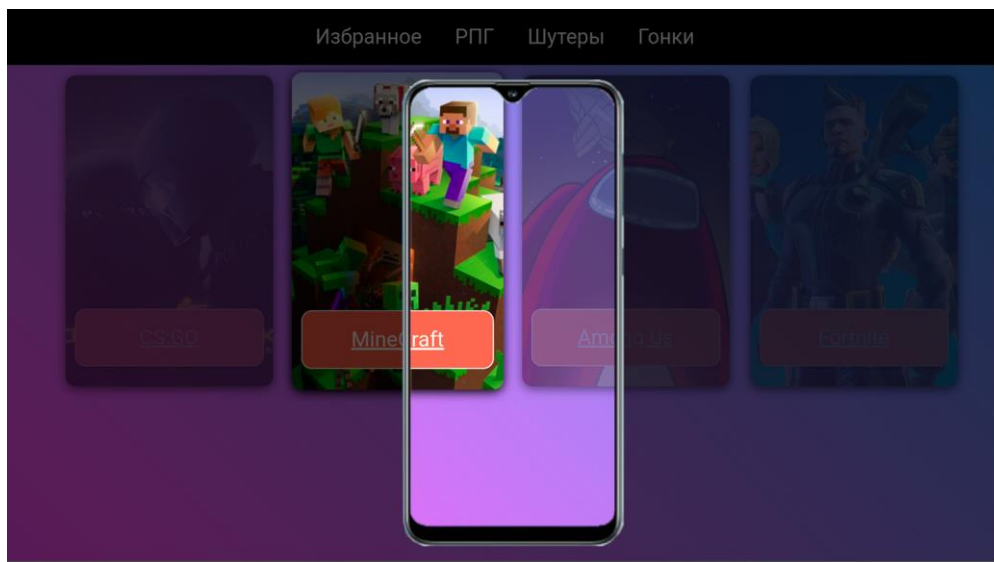
Для того, чтобы включить такой режим, необходимо в теге Head прописать `viewport`.

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Пример №1



Пример №2





# Атрибуты мета-тега viewport

- **name="viewport"** - атрибут указывающий на название мета-тега, т.к. они бывают разные;
- **content** - атрибут, который задает то, как будет отображаться страница. Включает в себя следующие значения:
  - **width** – ширина области видимости можно указать в пикселях, но удобнее указать значение device-width, которое задает размер экрана устройства;
  - **initial-scale** - позволяет указать какой будет начальный масштаб страницы, до того как пользователь начнет её масштабировать. Другими словами, какой масштаб страницы должен быть при загрузке. Здесь значение вещественное, 1 означает, что это масштаб один к одному.