МШП 20.35 УНИВЕРСИТЕТ КОД БУДУЩЕГО Минцифры России НАЦИОНАЛЬНЫЕ ПРОЕКТЫ РОССИИ ЦИФРОВАЯ ЭКОНОМИКА

Fullstack-разработка | Занятие №9

CSS
Состояние элементов

МШП

Цели урока:

1. Узнаем, какие бывают состояния элементов
2. Научимся делать простые переходы

МШП

Повторение

Разбор домашнего задания

МШП

Теория

Селектор псевдокласса

МШП

На данный момент вам знакомы следующие селекторы:

- по тегу `p { }`
- по классу `.active { }`

МШП

Псевдокласс —
это ключевое слово,
добавленное к селектору,
которое определяет его
особое состояние

```
p:hover {
    color: blue;
}
```

МШП

С помощью **псевдокласса** можно написать разные стили для разных состояний:

Пример:

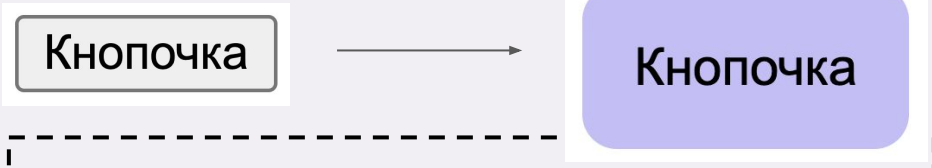
1. Базовое состояние
2. При наведении курсора

Псевдоклассы часто применяют, чтобы дать пользователю визуальную подсказку: *этот элемент интерактивный, нажми на него*

Давайте сделаем интерактивную кнопку!

МШП

Добавим кнопке немного базовых стилей




```
button {
  font-size: 30px;
  padding: 30px;
  border-radius: 20px;
  color: black;
  border: none;
  background-color: #c3bef7;
}
```

МШП

:hover

при наведении курсора

```
button:hover {
  background-color: #9c94f7;
  color: white;
  cursor: pointer;
}
```

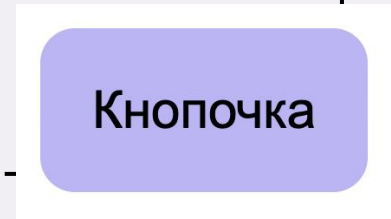
При наведении срабатывают сразу **оба селектора:**
button даёт базовые стили,
button:hover дополняет их

МШП

:hover

при наведении курсора

```
button:hover {
  background-color: #9c94f7;
  color: white;
  cursor: pointer;
}
```



Полезная визуальная подсказка, что элемент интерактивный – курсор в виде пальца

МШП

:active

при нажатии мыши

```
button:active {
  background-color: #8066f3;
}
```

Пока кнопка мыши зажата,
срабатывают сразу 3 селектора:
базовый, hover, active

МШП

:focus-visible

элемент в фокусе

Браузер добавляет активному элементу на странице **кольцо фокуса**

МШП

:focus-visible

элемент в фокусе

С помощью псевдокласса **focus-visible** можно поменять стили активного элемента

МШП

:focus-visible

элемент в фокусе

```
button:focus-visible {
  outline: 4px dotted #1281ff;
  outline-offset: 4px;
}
```

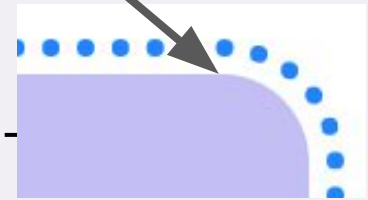
За линию вокруг элемента отвечает свойство **outline,** которое по своему синтаксису похоже на **border**

МШП

:focus-visible

элемент в фокусе

```
button:focus-visible {
  outline: 4px dotted #1281ff;
  outline-offset: 4px;
}
```



Свойство **outline-offset** задаёт расстояние между рамкой элемента и линией **outline**

Порядок важен!

```
button:active {
    color: yellow;
}
button:hover {
    color: red;
}
```

Какого цвета будет кнопка, если на неё нажать?

МШП

Порядок важен!

```
button:active {
    color: yellow;
}
button:hover {
    color: red;
}
```

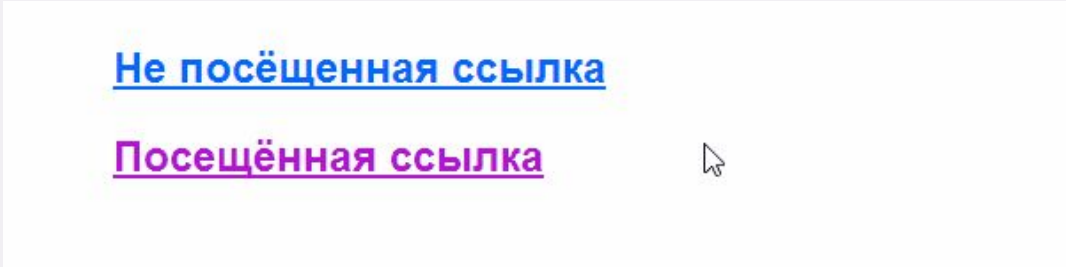
При изменении одинакового свойства браузер выбирает селектор псевдокласса, который указан **последним**

Порядок важен!

Чтобы состояния элементов не конфликтовали друг с другом, **всегда соблюдай эту последовательность**:

1. базовый селектор
2. :hover
3. :active
4. :focus-visible

МШП



У ссылок, как и тега button, тоже есть псевдоклассы **hover**, **active**, **focus-visible**

Но есть и собственный, секретный!

МШП

:visited

посещённая ссылка

```
a:visited {
  color: white;
  background-color: black;
}
```

mdn web docs

Браузеры накладывают серьёзный список ограничений на свойства, которые можно задавать ссылкам

МШП

Итоги

1. Псевдоклассы помогают сделать сайты более понятными для пользователей

2. Важно соблюдать порядок псевдоклассов в CSS файле

МШП

Тест

Псевдоклассы

МШП

Практика

Верстаем заготовку

МШП

Теория

Переход между состояниями

По умолчанию смена состояний происходит **мгновенно**

А мы, как пользователи, привыкли к **плавным анимациям**

МШП

transition

анимация перехода

```
button {
  /* прочие свойства */
  background-color: #c3bef7;
  transition: all 1s;
}
```

Чтобы переход между двумя состояниями происходил плавно, нужно добавить базовому селектору свойство **transition**

МШП

transition

анимация перехода

```
button {
  /* прочие свойства */
  background-color: #c3bef7;
  transition: background-color 1s,
              color 500ms;
}
```

Дополнительно можно указать, какие **свойства** должны изменяться плавно и за какое **время** они должны совершить переход к новому состоянию

МШП

Какие есть способы сделать плавный переход от меньшего размера к большему?

МШП

Изменение размера


```
button:hover {
  /* прочие свойства */
  width: 250px;
  height: 150px;
}
```

МШП

Изменение размера

Минусы решения:

1. Нужно просчитать размеры
2. Ширина и высота не анимируются :(

```
button:hover {
  /* прочие свойства */
  width: 250px;
  height: 150px;
}
```

МШП

Изменение размера


```
button:hover {
  /* прочие свойства */
  padding: 50px 80px;
}
```

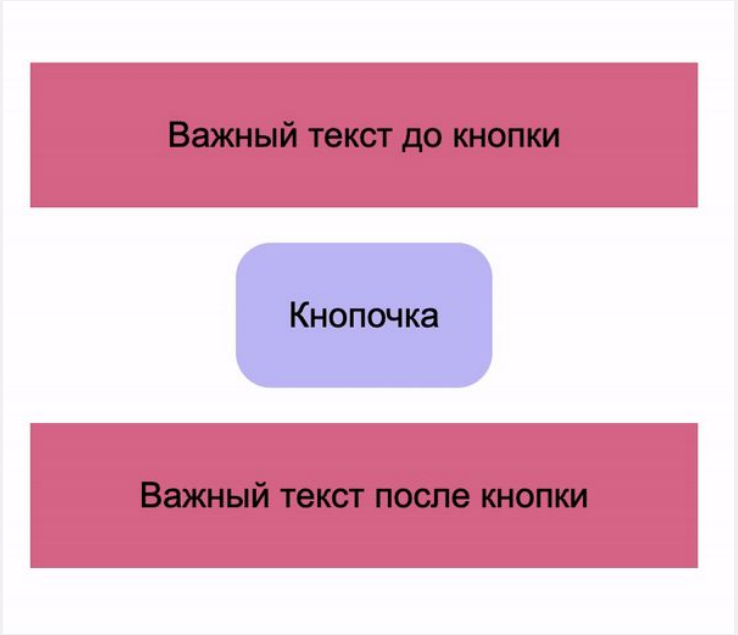
МШП

Изменение размера

Плюсы решения:

1. Указываем только новое значение отступа
2. Переход плавный

```
button:hover {
  /* прочие свойства */
  padding: 50px 80px;
}
```



Минусы решения:

Из-за изменения padding **размеры элементы меняются**, а это сдвигает соседние элементы

МШП

Изменение размера


```
button:hover {
  /* прочие свойства */
  transform: scale(1.2);
}
```

120%

Плюсы решения:

1. Указываем новый размер в процентах
2. Переход плавный
3. Соседей не толкаем!

МШП

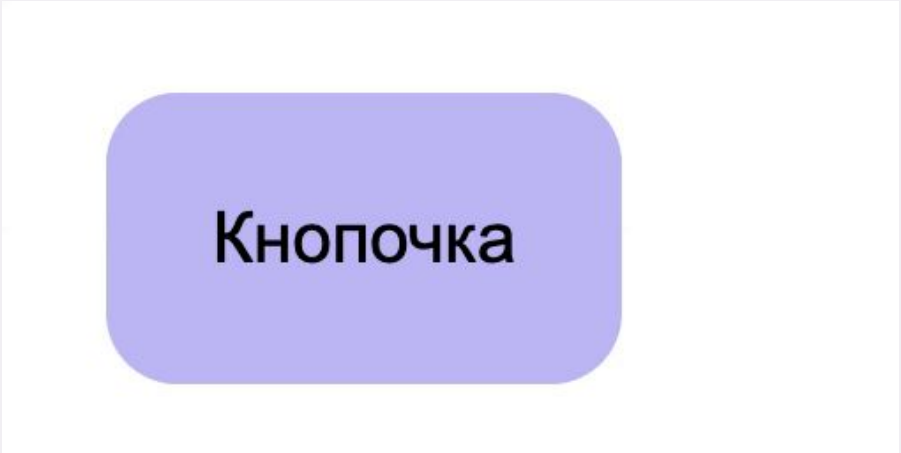
transform

scale

```
transform: scale(1.2);
увеличить на 20%

transform: scale(0.5);
уменьшить в 2 раза

transform: scaleY(1);
задать начальный размер по
вертикали

transform: scaleX(-1);
отразить зеркально по горизонтали
```

МШП

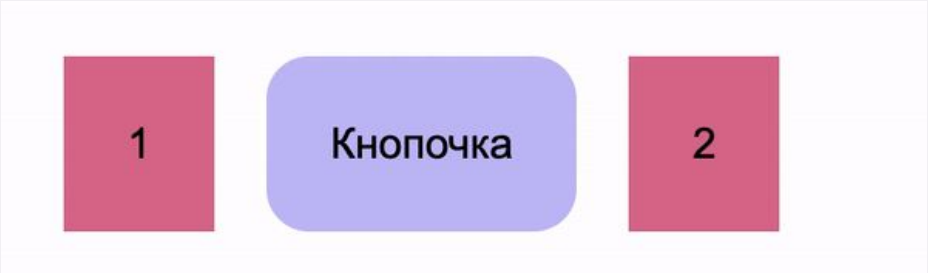
Какие есть способы переместить элемент правее?



МШП

Изменение положения

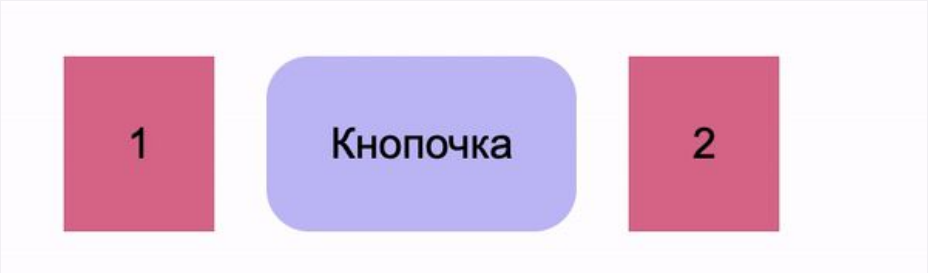
```
button:hover {
   /* прочие свойства */
   margin-left: 40px;
}
```



МШП

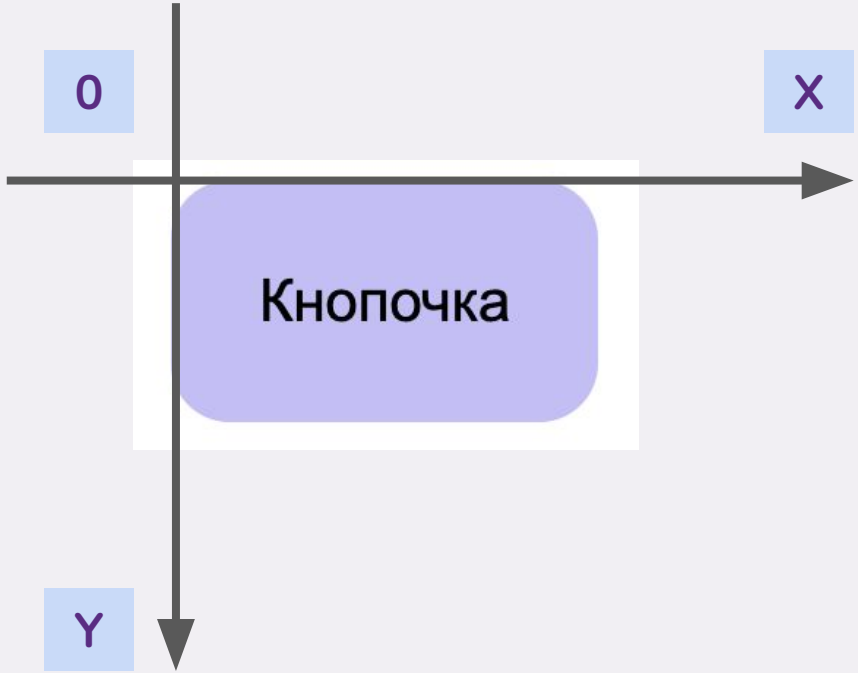
Изменение положения

```
button:hover {
  /* прочие свойства */
  transform: translateX(40px);
}
```



МШП

Координаты перемещения



МШП

transform

translate

```
transform: translateX(10px);
сдвинуть вправо на 10px

transform: translateX(-10px);
сдвинуть влево на 10px

transform: translateY(-10px);
сдвинуть вверх на 10px

transform: translateY(10px);
сдвинуть вниз на 10px
```

МШП

transform

```
button:hover {
  transform: translateX(10px)
             scale(1.2)
             translateY(-5px);
}
```

Если нужно задать несколько трансформаций для одного элемента, то это можно легко сделать через пробел

МШП

Практика

Переходы

МШП

Теория

Позиционирование

МШП

Цели урока:

- ✓ Узнали, какие бывают состояния элементов
- ✓ Научились делать простые переходы