JAVARUSH

Путь к карьере Frontend Fullstack разработчика

Модуль 1. WEB CORE

Уровень 15. Псевдоклассы и псевдоэлементы

JAVARUSH

Меняем стиль элементов в зависимости от состояния

С помощью псевдоклассов мы можем контролировать динамические параметры селекторов. Эти свойства сработают, когда селектор подходит под дополнительный признак.

Селектор может и отсутствовать. Тогда правило применится ко всем элементам, которые могут иметь признак этого псевдокласса.

Зачем нужны псевдоклассы:

- **Создание интерактивных элементов:** Кнопки, меню, ссылки.
- **Визуальная обратная связь:** Пользователь понимает, что произошло с элементом.
- **Улучшение пользовательского опыта:** Делает интерфейс более отзывчивым и понятным.

JAVARUSH

Псевдоклассы состояния

Позволяют изменять внешний вид элементов HTML в зависимости от того, что с ними делает пользователь

Основные псевдоклассы состояния:

- :hover - при наведении курсора мыши
- :focus - когда элемент получает фокус (например, при нажатии на поле ввода)
- :active - во время активного действия (например, при нажатии кнопки мыши)
- :visited - для уже посещенных ссылок

```
/* Кнопка, которая меняет цвет при
наведении и при нажатии */
.button {
   background-color: #4CAF50;
   color: white;
   padding: 15px 32px;
   text-align: center;
   text-decoration: none;
   display: inline-block;
   font-size: 16px;
}

.button:hover {
   background-color: #3e8e41;
}

.button:active {
   background-color: #3e8e41;
   box-shadow: 0 5px #666;
   transform: translateY(4px);
}
```

JAVARUSH

Комбинирование псевдоклассов

Мы можем комбинировать псевдоклассы для создания более сложных эффектов:

```
a:link { /* Все ссылки */
   color: blue;
}

a:visited { /* Посещенные ссылки */
   color: purple;
}

a:hover { /* При наведении курсора на любую ссылку */
   text-decoration: underline;
}

a:active { /* При клике на ссылку */
   color: red;
}
```

Дополнительные псевдоклассы:

- :enabled, :disabled - для элементов форм, которые могут быть включены или отключены.
- :checked - для элементов формы, которые выбраны.
- :not() - для выбора элементов, которые не соответствуют определенному селектору.

JAVARUSH

Несколько полезных моментов

1. **Приоритет псевдоклассов:**

   - **L-V-F-H-A:** Для ссылок рекомендуется следующий порядок объявления стилей: :link, :visited, :focus, :hover, :active. Это обеспечит правильное наследование стилей и предотвратит нежелательные эффекты.
   - **Понимание каскадности:** Помните о каскадности стилей. Более специфичные селекторы переопределят менее специфичные.

2. **Сброс стилей для тач-устройств:**

   - **@media:** Используйте медиа-запросы для сброса стилей :hover на мобильных устройствах.

3. **Сочетание с другими свойствами:**

   - **transition** - создавайте плавные переходы между состояниями.
   - **transform** - анимируйте элементы при переходе между состояниями.

JAVARUSH

Псевдоклассы группы :child

Псевдоклассы группы :child позволяют выбирать элементы в зависимости от их позиции среди других дочерних элементов внутри родительского элемента.

Основные псевдоклассы:

- **:first-child** - выбирает первый дочерний элемент.
- **:last-child** - выбирает последний дочерний элемент.
- **:only-child** - выбирает элемент, который является единственным дочерним элементом.
- **:nth-child(n)** - выбирает элементы, которые являются n-ым дочерним элементом.
  - n может быть числом (например, :nth-child(3)), ключевым словом (odd, even) или формулой (3n+1).
- **:nth-last-child(n)** - аналогично :nth-child, но отсчет ведется с конца.

- **Используйте :not()** для исключения элементов из выбора.
- **Понимание контекста:** Помните, что :nth-child учитывает все дочерние элементы, включая элементы, которые не соответствуют селектору.

```
/* Каждый третий элемент списка будет
красным */
ul li:nth-child(3n) {
    color: red;
}

/* Все элементы кроме первого и
последнего будут серыми */
ul li:not(:first-child):not(:last-
child) {
    color: gray;
}

/* Только элементы с четными
индексами будут жирными */
ul li:nth-child(even) {
    font-weight: bold;
}
```

JAVARUSH

Псевдокласс :not(): Исключаем лишнее

Этот псевдокласс позволяет исключить из выборки элементы, которые соответствуют определенному селектору.
Слева от :not необязательно должен быть селектор. Можно написать :not(.hidden), и браузер выберет вообще все элементы на странице, кроме тех, у которых есть класс .hidden.

Исключаем все ссылки
Все элементы на странице, кроме ссылок, будут иметь шрифт 16px.

```
body :not(a) {
   font-size: 16px;
}
```

Исключаем первый и последний элементы списка:
Все элементы списка, кроме первого и последнего, будут иметь отступы слева и справа.

```
ul li:not(:first-child):not(:last-child) {
   margin: 0 10px;
}
```

JAVARUSH

Псевдоклассы группы type

Позволяют выбирать элементы определенного типа в зависимости от их позиции среди других элементов такого же типа внутри общего родительского элемента.

Основные псевдоклассы:

- :first-of-type - выбирает первый элемент данного типа среди своих братьев.
- :nth-of-type(n) - выбирает каждый n-ый элемент данного типа.
- :last-of-type - выбирает последний элемент данного типа.
- :nth-last-of-type(n) - выбирает каждый n-ый элемент данного типа, начиная с конца.

Дополнительные возможности:

- :nth-of-type(even) - выбирает все четные элементы данного типа.
- :nth-of-type(odd) - выбирает все нечетные элементы данного типа.
- :only-of-type - выбирает элемент данного типа, если он единственный в своем роде внутри родительского элемента.

```
/* Выделим жирным шрифтом первый параграф */
p:first-of-type {
    font-weight: bold;
}

/* Выделим курсивом каждый второй параграф */
p:nth-of-type(2n) {
    font-style: italic;
}

/* Выделим подчеркиванием последний span */
span:last-of-type {
    text-decoration: underline;
}
```

JAVARUSH

Псевдоклассы форм

Позволяют стилизовать элементы форм в зависимости от их текущего состояния.

Основные псевдоклассы форм:

- :checked - применяется к выбранным элементам (чекбоксы, радиокнопки, опции).
- :disabled - применяется к отключенным элементам (неактивным).
- :enabled - применяется к включенным элементам (активным).
- :required - применяется к элементам, для которых значение обязательно.
- :optional - применяется к элементам, для которых значение не обязательно.
- :valid - применяется к элементам, которые прошли валидацию.
- :invalid: Применяется к элементам, которые не прошли валидацию.
- :in-range - выбирает элемент, если его значение находится в определенном диапазоне (для элементов типа ползунка)
- :out-of-range - выбирает элемент, если его значение не находится в определенном диапазоне

```
/* Стиль для выбранного чекбокса */
input[type="checkbox"]:checked + label {
    color: green;
}

/* Стиль для отключенной кнопки */
button:disabled {
    background-color: gray;
    cursor: not-allowed;
}

/* Стиль для поля ввода, которое не
прошло валидацию */
input:invalid {
    border: 2px solid red;
}

/* Стиль для поля ввода, когда оно
получает фокус */
input:focus {
    outline: 2px solid blue;
}
```

JAVARUSH

Псевдоэлементы

Псевдоэлементы — это специальные элементы, которые не определяются в HTML-коде, а создаются и стилизуются исключительно с помощью CSS. Они позволяют нам добавлять декоративные элементы, создавать сложные макеты и улучшать пользовательский интерфейс без необходимости изменять структуру HTML-документа.

Зачем нужны псевдоэлементы?

- **Создание декоративных элементов:** Добавление теней, границ, фона и других эффектов к элементам.
- **Создание оверлеев:** Наложение полупрозрачных слоев поверх элементов.
- **Стилизация первых букв или строк:** Выделение важных частей текста.
- **Создание кастомных элементов управления:** Создание кнопок, индикаторов и других элементов пользовательского интерфейса.

Основные псевдоэлементы:

- ::before - добавляет контент перед содержимым выбранного элемента.
- ::after - добавляет контент после содержимого выбранного элемента.
- ::first-letter - стилизует первую букву элемента.
- ::first-line - стилизует первую строку элемента.
- ::selection - стилизует выделенный пользователем текст.

JAVARUSH

::before

Позволяет добавлять новый элемент перед содержимым выбранного элемента, не трогая HTML-код. По сути, это как вставить элемент в самое начало выбранного тега, но без необходимости писать дополнительный HTML.

Зачем использовать ::before?

- **Добавление декоративных элементов:** Иконки, значки, границы, фоны и многое другое.
- **Создание кастомных элементов управления:** Кнопки, индикаторы, чекбоксы и т.д.
- **Создание сложных макетов:** Создание треугольников, стрелок, других геометрических фигур.
- **Улучшение читаемости:** Выделение важных частей текста, создание оглавлений.

Как работает ::before?

1. **Выбираем элемент:** С помощью селектора указываем, к какому элементу мы хотим применить псевдоэлемент.
2. **Добавляем ::before:** После селектора пишем ::before.
3. **Определяем стили:** В фигурных скобках указываем стили для нового псевдоэлемента.

```
p::before {
    content: "▶ "; /* Добавляем символ "play" */
    color: blue;
}
```

JAVARUSH

Основные свойства для работы с ::before

- **content:** определяет содержимое псевдоэлемента. Может быть текстом, эмодзи, URL для фонового изображения и т.д.
- **display:** определяет тип отображения псевдоэлемента (block, inline, inline-block и т.д.).
- **position:** позволяет позиционировать псевдоэлемент относительно родительского элемента (absolute, relative, fixed).
- **top, right, bottom, left:** используются для точного позиционирования.

Создание кастомных чекбоксов:

```
input[type="checkbox"]::before {
    content: "";
    display: inline-block;
    width: 20px;
    height: 20px;
    border: 1px solid #ccc;
    border-radius: 3px;
    background-color: #fff;
    margin-right: 5px;
}
input[type="checkbox"]:checked::before {
    background-color: #007bff;
}
```

JAVARUSH

::after

Псевдоэлемент ::after позволяет добавлять новый элемент после содержимого выбранного элемента. Подходит для создания различных эффектов, таких как:

- **Декоративные элементы:** стрелки, иконки, линии.
- **Оформление текста:** выделение ключевых слов, создание цитат.
- **Создание кастомных элементов управления:** кнопки, индикаторы.

Как работает ::after?

1. **Выбираем элемент:** С помощью селектора указываем, к какому элементу мы хотим применить псевдоэлемент.
2. **Добавляем ::after:** После селектора пишем ::after.
3. **Определяем стили:** В фигурных скобках указываем стили для нового псевдоэлемента.

Создание кастомных списков:

```
li::after {
   content: "•";
   color: green;
   margin-left: 5px;
}
```

JAVARUSH

Свойство content

Свойство content позволяет задать содержимое псевдоэлемента.
Это может быть:

- **Текст:** Любая строка символов, включая специальные символы и эмодзи.
- **Значения счетчиков:** Результат работы функций counter() и counters().
- **Значения атрибутов:** Результат работы функции attr().
- **URL изображения:** Для создания фонового изображения.

Как работает content с псевдоэлементами ::before и ::after?

- ::before - добавляет содержимое перед содержимым элемента.
- ::after - добавляет содержимое после содержимого элемента.

JAVARUSH

::first-letter

Псевдоэлемент ::first-letter позволяет стилизовать первую букву первого слова в блочном элементе. Это отличный способ добавить визуальный акцент и улучшить читаемость текста.

Как работает ::first-letter?

1. **Выбираем элемент:** С помощью селектора указываем, к какому элементу мы хотим применить псевдоэлемент.
2. **Добавляем ::first-letter:** После селектора пишем ::first-letter.
3. **Определяем стили:** В фигурных скобках указываем стили для первой буквы.

Основные свойства для работы с ::first-letter:

- font: шрифт, размер, стиль и другие свойства шрифта.
- color: цвет текста.
- background: фон первой буквы.
- border: граница вокруг первой буквы.
- margin: внешние отступы.
- padding: внутренние отступы.
- float: позволяет разместить первую букву сбоку от текста.
- vertical-align: выравнивание первой буквы по вертикали.

```
p::first-letter {
    font-size: 2em;
    color: blue;
    float: left;
}
```

::selection

Псевдоэлемент ::selection позволяет изменять внешний вид текста, который пользователь выделяет на странице.

Как работает ::selection?

1. **Выбираем элемент (необязательно):** Можно применить стили к выделению текста на всей странице или только к определенным элементам.
2. **Добавляем ::selection:** После селектора (или без него) пишем ::selection.
3. **Определяем стили:** В фигурных скобках указываем стили для выделенного текста.

Основные свойства для работы с ::selection:

- color: - цвет текста.
- background-color: - цвет фона.
- text-shadow: - тень текста.
- text-decoration: - подчеркивание, зачеркивание и другие декоративные эффекты.

```
p::selection {
    background-color: #e0e0e0;
    color: #333;
}
```

Этот псевдоэлемент не поддерживается в Safari на iOS

JAVARUSH

::backdrop

Псевдоэлемент ::backdrop позволяет нам стилизовать фон, который появляется за элементами, выдвигающимися на передний план (например, модальные окна, поповеры).

Этот фон обычно затемняет основное содержимое страницы, чтобы сосредоточить внимание пользователя на всплывающем элементе.

```
<dialog open>
</dialog>

dialog::backdrop {
   background-color: rgba(0, 0, 0, 0.5); /* Полупрозрачный черный фон */
}
```

Основные свойства для работы с ::backdrop:

- **background:** - определяет фон подложки (цвет, изображение, градиент).
- **backdrop-filter:** - позволяет применять фильтры к фону (размытие, сепия и т.д.).
- **opacity:** - устанавливает прозрачность фона.

JAVARUSH

Комбинирование псевдоклассов

Псевдоклассы и псевдоэлементы можно комбинировать для создания сложных селекторов, которые позволяют применять стили к элементам в зависимости от их состояния и структуры.

Синтаксис:

```
селектор:псевдокласс::псевдоэлемент {
   свойства: значения;
}
```

Стилизация первой буквы при наведении

В этом примере первая буква абзаца увеличивается и окрашивается в красный цвет при наведении курсора на абзац:

```
/* Стилизация первой буквы абзаца при наведении курсора */

p:hover::first-letter {
   font-size: 2em;
   color: #e74c3c;
}

<p>Hover over this paragraph to see the first letter change.</p>
```

Рефакторинг — это весело

JAVARUSH

Домашнее задание

Уровень 15. Псевдоклассы и псевдоэлементы