**Псевдоклассы** – это инструмент, который позволяет элементам реагировать на действия пользователя и давать обратную связь. Например, псевдокласс ***:hover*** выбирает элемент, на который наведен курсор мыши. Селектор псевдокласса ***.article-link:hover*** выберет на странице все элементы с классом ***.article-link*** и применит к ним стили из правила, когда на них будет наведен курсор.

*button {*

*border: 2px solid red;*

*color: white;*

*background-color: black;*

*}*

*button:hover { // перекрашивает кнопку при наведении*

*border: 2px solid black;*

*background-color: red;*

*}*

Рассмотрим пример, когда при наведении на элемент одного класса будет меняться прозрачность элементов другого класса. Этого можно достичь с помощью свойства ***hover***.

*.error {*

*font-size: 120px;*

*}*

*.message {*

*opacity: 0; // сделать прозрачным*

*}*

*.error:hover ~ .message {*

*opacity: 1; // сделать видимым*

*}*

Псевдокласс ***:link*** стилизует все ссылки, по которым еще не переходили. Его редко используют, потому что селекторы ссылки ***a*** и ***a:link*** абсолютно идентичны, и стили применяются к одним и тем же элементам.

Псевдокласс ***:visited*** стилизует все ссылки, по которым уже переходили. В целях безопасности возможности стилизации посещенных ссылок ограничены следующими свойствами: ***color***, ***background-color***, ***border-color***, ***outline-color***, ***column-rule-color***, ***fill***, ***stroke***. Также браузер будет игнорировать прозрачные (***transparent***) и полупрозрачные цвета (***rgb* с альфа-каналом**).

Псевдокласс ***:active*** стилизует элементы в тот миг, когда на них уже нажали, но еще не отпустили клавишу мыши. Это промежуточное состояние называется **активным**.

Для исключения конфликтов рекомендовано писать псевдоклассы для элемента в следующем порядке:

*link -> visited -> hover -> active*

Псевдокласс ***:focus*** стилизует элементы, на которые навелись любым способом (например, с помощью *Tab* клавиатуры или после клика по ним). Как правило он изменяет обводку (***outline***) элемента.

*button:focus {*

*outline: 15px solid lightpink;*

*}*

Псевдокласс ***:focus-visible*** стилизует элементы, на которые навелись только с клавиатуры. Потребуется сбросить стили для ***:focus***.

*button:focus {*

*outline: none;*

*}*

*button:focus-visible {*

*border-color: rebeccapurple;*

*color: palevioletred;*

*}*

Псевдокласс ***:focus-within*** стилизует элементы, которые либо сами находятся в фокусе (на которые навелись любым способом), либо имеют дочерние элементы в фокусе.

*form:focus-within { // при фокусировки любого элемента формы, форма сменит цвет*

*background-color: #c4c4c4;*

*}*

Псевдоклассы ***:nth-child()*** и ***:nth-last-child()*** стилизуют каждый ***n***-й элемент с начала и конца соответственно. Внутри скобок указывается либо формула, либо номер, либо ключевые слова (например, ***even*** для четных и ***odd*** для нечетных). Подсчет элементов ведется внутри одного блока родителя. Визуализировать формулы можно [тут](https://css-tricks.com/examples/nth-child-tester/).

*li:nth-child(3n) { // перекрасим каждый 3й элемент списка*

*color: #fd24af;*

*}*

*li:nth-child(5) { // подчеркнем 5й элемент списка*

*text-decoration: underline;*

*}*

*.card:nth-child(even) { //* заменить правый отступ на левый у каждого четного элемента

*margin-right: 0;*

*margin-left: 80px;*

*}*

Псевдоклассы ***:first-child*** и ***:last-child*** являются производными ***:nth-child()*** и стилизуют первый и последний элементы соответственно внутри одного блока родителя.

Псевдокласс ***:nth-of-type()*** и его производные работают аналогично ***:nth-child()***, но считают только однотипные элементы.

*button:first-of-type { // перекрасить цвет первой кнопки внутри родителя*

*color: #fd24af;*

*}*

Псевдокласс ***:not()*** используется для исключения селектора из выборки. Внутри скобок могут быть и другие псевдоклассы, перечисленные через запятую.

*.text:not(h2) { // изменить цвет всех элементов, кроме заголовков h2*

*color: #fd24af;*

*}*

*p:not(:last-of-type) { // задать отступы всем абзацам, кроме последнего*

*margin-bottom: 25px;*

*}*

Псевдокласс ***:has()*** позволяет изменить родительский элемент указанного селектора. Сработает для любого уровня вложенности.

*.card:has(.label) { // если у card есть пометка label, то card меняет цвет рамки*

*border-color: #5f0dee;*

*}*

Пример выпадающего списка через псевдоклассы <https://codepen.io/spacehaz/pen/poYdWXz>

**Псевдоэлементы** – это инструмент, который позволяет выбрать только часть какого-то элемента. Например, псевдоэлемент ***::first-line*** выберет только первую строку текста, какой бы длины она ни была и как бы ни менялась. Селектор псевдоэлемента ***.article-text::first-line*** выберет на странице все элементы с классом ***.article-text*** и применит стили из правила только к их первой строке текста.

Псевдоэлементы ***::before*** и ***::after*** позволяют встраивать блоки перед и после элемента соответственно с помощью свойства ***content***. Не забываем сбрасывать ***box-sizing*** для них через общий селектор ***\****. Скринридеры не увидят эти изменения. Основная задача этих псевдоэлементов – избавиться от пустых или бессмысленных блоков. Не работает с тегами, вместо которых что-то рисуется.

*p::before { // добавим текст перед абзацем*

*content: "Заметка! ";*

*color: #fd24af;*

*}*

*p::after { // добавим текст после абзаца*

*content: "!";*

*color: #fd24af;*

*}*

*.header::before { // добавим затемняющий слой поверх картинки и перед заголовком*

*content: ‘ ’;*

*position: absolute;*

*z-index: -1;*

*inset: 0;*

*background-color: rgb(0 0 0 / 0.5);*

*}*

Свойство ***white-space*** позволяет редактировать текст внутри свойства ***content***. Например, переносить на новую строку.

*.page\_\_title:hover::before { // всплывающая подсказка при наведении на текст*

*content: 'На самом деле нет.\AВажнее &mdash; научиться ошибаться.';*

*white-space: pre;*

*}*

Псевдоэлемент ***::marker*** используется для создания самодельного маркера в списках. На текущий момент поддерживает свойства ***content***, ***white-space***, ***text-transform***, ***letter-spacing***, ***color*** и все свойства семейства ***font***. Раньше эту задачу решали при помощи ***::before***.

*li {*

*display: flex;*

*align-items: center;*

*}*

*li::before { // создать кастомный маркер и поместить перед элементом списка*

*content: “”;*

*display: inline-block;*

*width: 15px;*

*height: 15px;*

*margin-inline-end: 10px;*

*border-radius: 50%;*

*background-color: red;*

*}*

*li:nth-child(2)::before { // сменить цвет маркера*

*background-color: orange;*

*}*

*li::marker { // аналогично через marker*

*color: #23054a;*

*font-weight: bold;*

*font-size: 2rem;*

*}*

Псевдоэлемент ***::selection*** позволяет менять стили текста при выделении. На текущий момент поддерживает свойства ***color***, ***background-color***, ***text-decoration*** и ***text-shadow***.

*p::selection { // покрасить цвет и фон при выделении текста*

*color: #6fc1fd;*

*background-color: black;*

*}*

Псевдоэлемент ***::placeholder*** позволяет менять стили текста подсказок в полях ввода.

*input::placeholder {*

*color: rgb(0 0 0 / 60%);*

*}*