1. **Псевдоклассы состояния**

:hover – наведение мышкой, при задании свойств для :hover надо учитывать влияние этих свойств на соседние элементы(пример с border для пункта меню)

Если задать :hover для div. Вынести ссылку за пределы дива (position: absolute;). При наведении на а сработает :hover для div. Ссылка является частью div, она в него вложена. Разбор работы :hover при вложенности на примере.

:has – Стили, указанные для элемента с псевдоклассом :has(), будут применены, если хотя бы один из относительных селекторов, переданных в качестве аргумента, соответствует хотя бы одному элементу. С помощью данного псевдокласса можно указать стили для родительского или предшествующего элементов.

:focus - перемещение по странице с помощью клавиатуры, можно прописать для любого тега HTML, но если это не элемент формы, то нужно указывать атрибут tabindex - регулирует порядок элементов при нажатии TAB или contenteditable – в зависимости от значения, запрещает (false)или разрешает (true) редактирование пользователем содержимого тега в браузере 🡪 <http://shpargalkablog.ru/2014/04/focus-css.html>

outline - отличие от border только для всех сторон, внешний, для чего нужен , сравнение border и outline при использовании на строчном элементе, использование outline-offset (расстояние между outline и объектом), outline-color (отдельно цвет для outline), border-radius работает с outline

:focus-within - соответствует элементу, который либо сам находится в фокусе, либо содержит элемент, который находится в фокусе. Другими словами, он представляет элементу, который соответствует псевдоклассу :focus либо имеет потомка, который соответствует :focus.

:focus-visible - применяется, в то время как элемент соответствует псевдоклассу :focus, и UA (User Agent) определяет с помощью эвристики. Проще говоря браузер определяет на этом элементе :focus будет работать только с клавиатурой или и с мышкой тоже

:active - состояния при активации (для ссылки и кнопки момент нажатой кнопки мыши), пример на нажатой кнопке, прописываем так как в дальнейшем будет соответствовать событию “click”

:visited - посещенные ссылки

Порядок записи псевдоклассов:

**L** — :link, **V** — :visited, **H** — :hover, **F** — :focus, **A** — :active

:target при нажатии на ссылку переходит к указанному объекту(используя #ID) и меняет его CSS свойства 🡪 [http](http://shpargalkablog.ru/2016/08/target-css.html#dvaGreen)://shpargalkablog.ru/2016/08/target-css.html#dvaGreen Присваивается тому элементу куда переходим, чтобы отменить надо сделать переход по другой ссылке. Описывает состояния элемента, на который переходим. (для отмены действия :target должна сработать другая ссылка, перезагрузка не поможет)

О псевдоклассах 🡪 [http](http://shpargalkablog.ru/2012/02/psevdoklassy-css.html#target)://shpargalkablog.ru/2012/02/psevdoklassy-css.html#target

настройка внешнего вида курсора: cursor: значение; <https://developer.mozilla.org/ru/docs/Web/CSS/cursor>

cursor: auto; - определяется браузером

cursor: none; - исчезает при наведении мыши на данный элемент

cursor: url(‘test.png’), auto; - своя картинка, можно svg, png размер не более 32рх, через запятую в качестве запасного значения любое стандартное

pointer-events – определяет может ли пользователь нажимать на элемент, наследуется всеми элементами

по умолчанию auto, если поставить none, то на элемент и его дочерние элементы нажать нельзя

1. **Псевдоклассы положения**

:first-child - обращение к первому элементу в блоке

.text p:first-child { color: red;}

:first-of-type - обращение к первому элементу в блоке данного типа

.text p:first-of-type { color: red; }

Аналогично для последнего элемента в блоке

:last-child

:last-of-type

Использование \* 🡪 Любой первый объект внутри родителя

.text \*:first-child { color: red; }

:nth-child(…)

:nth-of-type(…) - обращение к конкретному элементу или группе элементов (использование формул) в блоке 🡪 <https://wp-kama.ru/wp-content/uploads/2015/06/nth-child-posobie.html>

odd - нечет

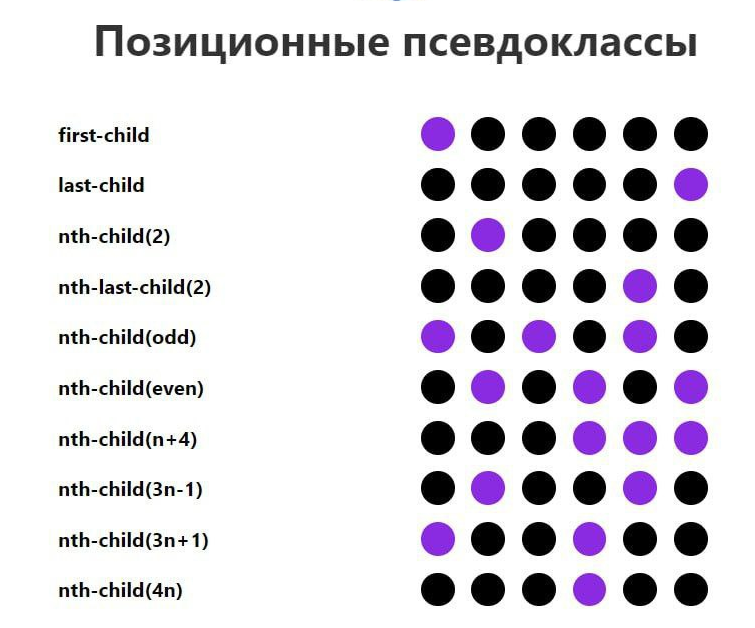
even - чет

применение :not(:first-child) - не первый

:only-child - находит любой элемент, являющийся единственным потомком родителя(описывает состояние элемента который является единственным потомком родителя)

:only-of-type - аналогично

:empty - обращение к пустому элементу (нет контента, нет дочерних элементов) Важно!! Элемент должен быть ПУСТЫМ без переносов строки и пробелов 🡪 <div class=”block”></div>



:is() – позволяет проще записывать комбинированные селекторы, перечисляя вложенные селекторы для указанного элемента.

.class:is(.class2) – аналогично .class.class2

.block :is(h2,h3,p) – аналогично .book h2, .book h3, .book p

Аналогично можно применять для состояний

.link:is(:hover, :focus) – для класса link при hover и focus одинаковые свойства, аналогично .link:hover, .link:focus

Также можно использовать для родительского элемента например :is(.head, .card, main) div – аналогично .head div, .card div, main div

:where() принимает один или несколько селекторов в качестве аргумента. Браузер при чтении применяет стили к любому из селекторов-аргументов.

:where() похож на [:is()](https://doka.guide/css/is/) по своей механике и также полностью поддерживается во всех стабильных браузерах с [января 2021 года](https://caniuse.com/mdn-css_selectors_where). Различие заключается в [специфичности](https://doka.guide/css/specificity/): у :where() она нулевая, а у :is() равна наиболее специфичному из переданных селекторов.