Nth-child и nth-of-type

<http://css.yoksel.ru/nth-child/>

Как работает nth-child()?

В круглых скобках задается размер цикла: например (3n) выберет каждый третий элемент.

Также можно задать сдвиг вперед или назад: (3n+1) — найдет каждый третий и сделает один шаг вперед, а (3n-2) — два шага назад от найденого.

Если размер цикла не задан — (n) — выберутся все элементы списка.

Если размер цикла не задан, но задано конкретное число — (5) — выберется элемент списка с этим индексом.

nth-child в своем цикле считает все элементы, находящиеся на одном уровне вложенности относительно родительского элемента.

Селектор работает во всех современных браузерах с поддержкой CSS3.

Селекторы по атрибутам в CSS

обратиться к элементам html-документа по значению его атрибутов.

Атрибут точно соответствует заданному значению

html

<h2 id="first-title" class="magical" rel="friend">David Walsh</h2>

css

h2[rel=friend] { /\* woohoo! \*/ }

a[href=http://perishablepress.com] { color: red; }

input[type=text] { padding: 3px; }

input[type=radio] { float: left; }

Атрибут содержит заданное значение

<h1 rel="xxxexternalxxx">Attribute Contains</h1>

h1[rel\*=external] { color: red; }

Атрибут начинается с заданного значения

<h1 rel="external-link yep">Attribute Begins</h1>

h1[rel^=external] { color: red; }

Атрибут заканчивается на заданное значение

<h1 rel="friend external">Attribute Ends</h1>

h1[rel$=external] { color: red; }

Атрибут находится в списке, разделённом пробелами

<h1 rel="friend external sandwich">Attribute Space Separated</h1>

h1[rel~=external] { color: red; }

Атрибут находится в списке, разделённом дефисами

<h1 rel="friend-external-sandwich">Attribute Dash Separated</h1>

h1[rel|=external] { color: red; }

Соответствие по нескольким атрибутам

<h1 rel="handsome" title="Important note">Multiple Attributes</h1>

h1[rel=handsome][title^=Important] { color: red; }

GRID

Элемент, к которому применяется display: grid. Это прямой родитель для всех элементов сетки. Каждый дочерний элемент это ячейка таблицы. Родитель строит номера. Дети их заселяют.

Свойства для родительского элемента (Контейнера сетки)

display определяет элемент как контейнер и устанавливает новый контекст форматирования сетки для его содержимого. Обратите внимание на то, что column, float, clear и vertical-align не дадут никакого эффекта на контейнере.

* grid - формирует сетку как блок;
* inline-grid - формирует сетку как инлайновый блок ;
* subgrid - если ваш контейнер это ещё и элемент (вложенная сетка), то вы можете использовать это свойство для обозначения того, чтобы размеры строк/колонок были взяты из родительского элемента, а не определяли собственный;

Отличие grid и inline-grid:

<style>

body {font: 12px Verdana, sans-serif;}

.grid, .inline-grid {

border: 3px dashed #333;

grid-gap: 3px 5px;

margin: 5px 0;

}

.grid {

display: grid;

}

.inline-grid {

display: inline-grid;

}

.grid-item {

border: 3px solid #F2E205;

border-radius: 5px;

padding: 10px;

background: rgba(242,226,5,.8);

}

</style>

<div class="grid">

<div class="grid-item">Это display: grid</div>

</div>

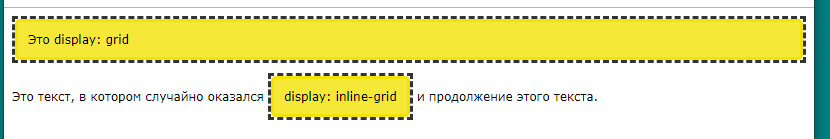
Это текст, в котором случайно оказался

<div class="inline-grid">

<div class="grid-item">display: inline-grid</div>

</div>

и продолжение этого текста.



subgrid используется если нужно привязаться к линиям сетки родителя. Примеры можно посмотреть в <https://codepen.io/rachelandrew/pen/mgpqBv> и <https://codepen.io/rachelandrew/pen/YMYEVO?editors=1100> и <https://codepen.io/rachelandrew/pen/axLzYv?editors=1100> разные способы привязки.

grid-template-columns grid-tempate-rows

Определяет колонки и строки сетки с помощью списка значений. Когда вы оставляете пустое пространство между значениями треков, линиям сетки автоматически присваиваются числовые имена. При этом дочерние элементы описываются с помощью номеров столбцов и строк. Или по именам, если определены. Всякие заморочки с двойными именами и т.д. Разберёмся по ходу дела.

Фреймы вычисляются после выделении всех фиксированных значений.

grid-template-areas

Определяет шаблон сетки ссылаясь на имена областей, которые заданы с помощью свойства grid-area. Повторение названия области приводит к тому, что содержимое охватывает эти ячейки. Точка означает пустую ячейку.

Пример:

.item-a {grid-area: header;}

.item-b {grid-area: main;}

.item-c {grid-area: sidebar;}

.item-d {grid-area: footer;}

.container {

grid-template-columns: 50px 50px 50px 50px;

grid-template-rows: auto;

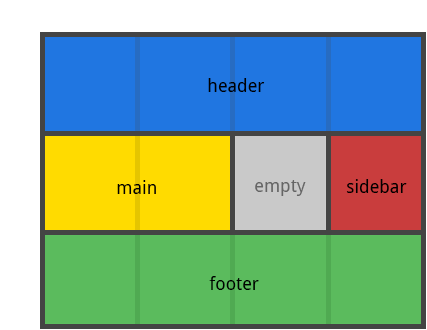
grid-template-areas:

"header header header header"

"main main . sidebar"

"footer footer footer footer";

}



grid-template

Сокращение для grid-template-rows, grid-template-columns, и grid-template-areas.

Вот пример:

.container {

grid-template:

[row1-start] 25px "header header header" [row1-end]

[row2-start] "footer footer footer" 25px [row2-end]

/ auto 50px auto;

}

Что эквивалентно следующему:

.container {

grid-template-rows: [row1-start] 25px [row1-end row2-start] 25px [row2-end];

grid-template-columns: auto 50px auto;

grid-template-areas:

"header header header"

"footer footer footer";

}

grid-column-gap grid-row-gap

Определяет размер ширины линий. Отступы создаются только между колонками и строками, но не для внешних краев сетки.

grid-gap

Сокращение для grid-row-gap и grid-column-gap.

justify-items

Выравнивает содержимое внутри сетки вдоль оси строки (в отличии от align-items который выравнивает элементы вдоль оси столбца). Это значение применяется ко всем элементам сетки внутри контейнера.

* start - выравнивает содержимое по левой стороне области;
* end - выравнивает содержимое по правой стороне области;
* center - выравнивает содержимое по центру области;
* stretch - заполняет всю ширину области (по умолчанию);

align-items

Выравнивает содержимое внутри сетки вдоль оси столбца. Смотреть выше.

justify-content

Выравнивает саму сетку внутри контейнера

* start - выравнивает сетку по левой стороне контейнера;
* end - выравнивает сетку по правой стороне контейнера;
* center - выравнивает сетку по центру контейнера;
* stretch - масштабирует элементы чтобы сетка могла заполнить всю ширину контейнера;
* space-around - одинаковое пространство между элементами, и полуразмерные отступы по краям;
* space-between - одинаковое пространство между элементами, без отступов по краям;
* space-evenly - одинаковое пространство между элементами, и полноразмерные отступы по краям;

align-content

Свойства для дочерних элементов (Grid элементы)

grid-area - даёт название элементу чтобы можно было ссылаться на него с помощью шаблона созданного через grid-template-areas свойство.

.item-d {

grid-area: header

}

min-content и max-content

Работают для нескольких слов

align-self

Это свойство помогает позиционировать содержимое по вертикали.

justify-self

Это свойство помогает позиционировать содержимое по горизонтали.