

Профессиональная вёрстка

Урок 8

Grid layout, НОВЫЕ ВОЗМОЖНОСТИ CSS3

Понятие и применение адаптивного дизайна. Основы синтаксиса и примеры.

[CSS3 Grid Layout](#)

[Основные понятия](#)

[Начало работы с grid сеткой](#)

[Колонки и столбцы](#)

[Единицы измерения](#)

[Размещение элементов сетки](#)

[CSS3 рамки](#)

[Закругление углов с помощью border-radius](#)

[Рамки-изображения border-image](#)

[Ширина рамки-изображения border-image-width](#)

[Ресурс рамки-изображения border-image-source](#)

[Элементы рамки-изображения border-image-slice](#)

[Повтор рамки-изображения border-image-repeat](#)

[Смещение рамки-изображения border-image-outset](#)

[Смещение внешней рамки outline-offset](#)

[Градиентная рамка](#)

[Практическое задание](#)

[Дополнительные материалы](#)

[Используемая литература](#)

CSS3 Grid Layout

W3C описывает модуль CSS Grid Layout как систему двумерного макета, оптимизированного для дизайна пользовательского интерфейса. Главная идея, лежащая в основе макета сетки, заключается в разделении веб-страницы на столбцы и строки. В образовавшиеся области сетки можно помещать элементы сетки, а управлять их размерами и расположением можно с помощью специальных свойств модуля.

Кроме того, благодаря своей способности явно размещать элементы в сетке, Grid Layout позволяет кардинально преобразовывать структуру визуального макета (отображаемого на экране), не требуя соответствующих изменений разметки.

Хотя многие макеты могут быть отображены с помощью Grid или Flexbox, у каждого есть свои особенности. Grid обеспечивает двумерное выравнивание, использует нисходящий подход к макету, допускает явное перекрытие элементов и обладает более мощными связующими возможностями. Flexbox фокусируется на распределении пространства по оси, использует более простой восходящий подход к макету, может использовать систему переноса строк на основе размера контента для управления своей вторичной осью и опирается на базовую иерархию разметки для построения более сложных макетов.

Основные понятия

Сетка (grid) представляет собой набор пересекающихся горизонтальных и вертикальных линий, делящих пространство grid-контейнера на области, в которые могут быть помещены элементы сетки.

Линии сетки (grid lines) — это невидимые горизонтальные и вертикальные разделительные линии, они существуют по обе стороны от строки и столбца. На них можно ссылаться по числовому индексу (используя свойства `grid-column-start`, `grid-column-end`, `grid-row-start` и `grid-row-end`) или имени, заданному в CSS-коде. Числовые индексы сетки зависят от стиля языка, поэтому первым столбцом может быть как самый левый, так и самый правый столбец.

Дорожка сетки (grid track) — пространство между двумя соседними линиями сетки, используется для определения либо столбца, либо строки сетки. Дорожка идет от одного края контейнера к другому, размер зависит от расположения линий сетки, которые ее определяют. Дорожки сетки аналогичны столбцам и строкам таблицы. По умолчанию смежные дорожки плотно прилегают друг к другу, задать расстояние между ними можно с помощью свойств `row-gap`, `column-gap` и `gap`.

Ячейка сетки (grid cell) — пространство, ограниченное четырьмя линиями сетки, аналогично ячейке таблицы. Ячейка сетки — это область, в которой можно разместить контент. Это наименьшая единица сетки, на которую можно ссылаться при позиционировании элементов сетки. К ячейкам сетки нельзя обращаться напрямую с помощью CSS-свойств.

Область сетки (grid area) — прямоугольная область, ограниченная четырьмя линиями сетки и состоящая из одной или нескольких соседних ячеек. Область может быть такой же маленькой, как одна ячейка, или такой же большой, как все ячейки сетки. Область сетки может быть задана явно с помощью свойства `grid-template-areas`, по умолчанию на нее ссылаются ограничивающие линии сетки.

Элементы сетки (grid items) — отдельные элементы, которые назначаются области сетки (или ячейке сетки). Каждый контейнер-сетка включает ноль и более элементов сетки; каждый дочерний элемент контейнера-сетки автоматически становится элементом сетки. Дорожки, ячейки и области сетки построены из линий сетки. Тем не менее, не требуется, чтобы все области сетки были заполнены элементами, вполне возможно, что некоторые ячейки, или даже большинство, будут пустыми. Также

возможно, что элементы сетки будут перекрывать друг друга, либо определять перекрывающиеся области сетки.

Начало работы с grid сеткой

Как и в работе с flexbox, так и в grid-сетке необходимо создать блок родитель, который будет отвечать за расположение элементов внутри. Для этого родительскому блоку необходимо задать значение `display: grid;`. В отличии от flexbox-верстки, grid никак не меняет стандартное расположение элементов и они также размещаются в колонку.

Колонки и столбцы

Когда вы создаете контейнер-сетку, она по умолчанию имеет один столбец и одну строку, которые занимают полный размер контейнера. Для разделения контейнера-сетки на столбцы и строки используются свойства `grid-template-columns`, `grid-template-rows` и `grid-template-areas`. С помощью этих свойств можно определить сетку явно.

Окончательная сетка может оказаться больше из-за элементов, размещенных вне явной сетки; в этом случае будут созданы неявные дорожки, их размер будет определяться свойствами `grid-auto-rows` и `grid-auto-columns`.

Свойства `grid` и `grid-template` — это сокращенные обозначения, которые можно использовать для одновременной установки всех трех явных свойств сетки `grid-template-columns`, `grid-template-rows` и `grid-template-areas`. Grid сбрасывает свойства, управляющие неявной сеткой, тогда как свойство `grid-template` оставляет их без изменений.

Единицы измерения

Размеры дорожек сетки можно задавать с помощью положительных значений, используя относительные единицы длины — например, `em`, `vh`, `vw`; абсолютные единицы длины — `px`; и проценты `%`. Размеры в `%` вычисляются от ширины или высоты контейнера-сетки.

`fr` — единица длины, которая позволяет создавать гибкие дорожки. Не является единицей измерения в обычном ее понимании, поэтому не может быть представлена или объединена с другими типами единиц в выражениях `calc()`. Общий размер фиксированных строк или столбцов вычитается из доступного пространства контейнера-сетки. Оставшееся пространство делится между строками и столбцами с гибкими размерами пропорционально их коэффициенту

Еще одной очень интересной особенностью работы с grid является то, что данная технология уже работает с адаптивными размерами, такими как минимальные и максимальные размеры дорожек.

`max-content` устанавливает для дорожки размер, который занимает максимально необходимое пространство с учетом содержимого элемента сетки.

`min-content` позволяет занимать минимальное пространство, необходимое для этого содержимого, при этом ширина элемента ориентируется на самое длинное слово или на самое широкое изображение.

Функция `minmax(min,max)` определяет диапазон размеров, больше или равный `min` и меньше или равный `max`. Если `max < min`, то `max` игнорируется, а `minmax(min,max)` обрабатывается как `min`. Значения в `fr` можно устанавливать только как максимальное.

Если же размеры не известны у блока, тогда применяют автоматические размеры.

Значение `auto` ориентируется на содержимое элементов сетки одной дорожки. Как минимум, рассматривается как минимальный размер элемента сетки, как определено `min-width` или `min-height`. Как максимум, обрабатывается так же, как и `max-content`. Может растягиваться за счет свойств `align-content` и `justify-content`.

Как можно заметить, значения ширины или высоты элементов чаще всего повторяются, поэтому применяют значение `repeat()`.

Нотация `repeat()` представляет повторяющийся фрагмент списка дорожек, что позволяет записать в более компактной форме большое количество одинаковых по размерам столбцов или строк.

Первый аргумент задает количество повторений, которое может быть задано с помощью положительного целого числа или ключевых слов. Второй аргумент — размер повторяющейся дорожки.

Элементы сетки, которые не размещены явно, автоматически помещаются в незанятое пространство в контейнере-сетке с помощью алгоритма автоматического размещения. Свойство `grid-auto-flow` управляет автоматическим размещением элементов сетки без явного положения. После заполнения явной сетки (или если явной сетки нет) автоматическое размещение также приведет к генерации неявных дорожек сетки. Свойство не наследуется.

Размещение элементов сетки

Каждый элемент сетки связан с областью сетки, которая определяет содержащий блок для элемента сетки. Положение элементов сетки определяется расположением линий сетки и диапазоном сетки — количеством занимаемых дорожек сетки. По умолчанию элемент сетки занимает одну дорожку на каждой оси. Поэтому можно опустить значение `grid-column-end` или `grid-row-end`.

Свойства размещения на сетке — `grid-row-start`, `grid-row-end`, `grid-column-start` и `grid-column-end` и их краткая запись `grid-row`, `grid-column` и `grid-area` позволяют определить размещение элемента сетки

Свойства `grid-row` и `grid-column` являются сокращенными именами для свойств `grid-row-start/grid-row-end` и `grid-column-start/grid-column-end` соответственно. Если заданы два значения, первое (до косой черты) устанавливается для параметра `grid-row-start/grid-column-start`, второе — для `grid-row-end/grid-column-end`.

Если второе значение опущено, а первое указано в формате пользовательского идентификатора, то `grid-row-end/grid-column-end` также устанавливается в пользовательское имя сетки. В противном случае оно вычисляется в `auto`.

Если для свойства `grid-area` указано четыре значения, первое устанавливается для `grid-row-start`, второе — для `grid-column-start`, третье — для `grid-row-end`, четвертое — для `grid-column-end`. Если `grid-column-end / grid-row-end` не указан, а `grid-column-start / grid-row-start` указан в форме пользовательского имени, то для `grid-column-end/grid-row-end` также устанавливается значение пользовательского имени линии; в противном случае он будет установлен на `auto`.

Когда `grid-column-start` опущен, а значение `grid-row-start` указано в форме пользовательского имени, оно устанавливается для всех четырех значений. В противном случае оно устанавливается на `auto`.

Промежутки между элементами сетки

Свойства `row-gap` и `column-gap` (и их сокращенная запись `gap`), если они указаны в контейнере сетки, определяют промежутки между строками и столбцами сетки. При определении размера дорожки каждый промежуток рассматривается как дополнительная пустая дорожка указанного размера.

Дополнительный промежуток также может быть добавлен между дорожками за счет свойств `justify-content` и `align-content`. Промежутки добавляются только между двумя дорожками сетки, то есть не перед первой и не после последней дорожки.

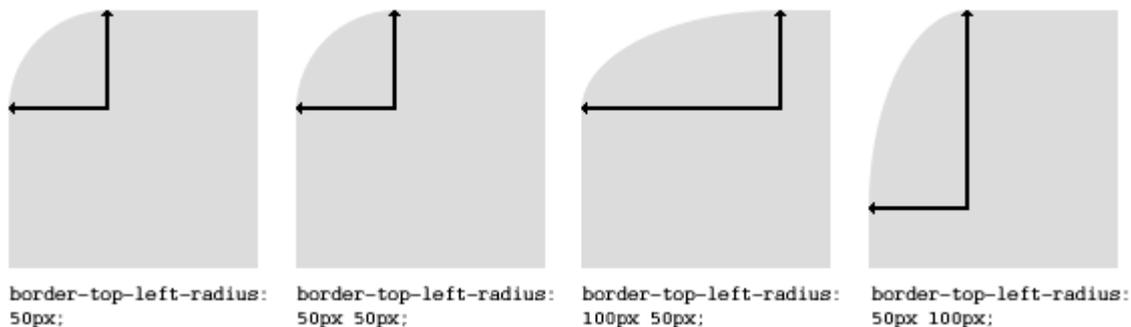
CSS3-рамки

CSS3-рамка дополняет возможности форматирования границ элементов с помощью свойств, позволяющих закруглить углы элемента, а также использовать изображения для оформления границ элемента.

Круглые углы и рамки-изображения:

1. Закругление углов с помощью `border-radius`.
2. Рамки-изображения `border-image`.
 - a. Ширина рамки-изображения `border-image-width`.
 - b. Ресурс рамки-изображения `border-image-source`.
 - c. Элементы рамки-изображения `border-image-slice`.
 - d. Повтор рамки-изображения `border-image-repeat`.
 - e. Смещение рамки-изображения `border-image-outset`.
3. Смещение внешней рамки `outline-offset`.
4. Градиентная рамка.

Закругление углов с помощью `border-radius`



Свойство позволяет закруглить углы строчных и блочных элементов. Кривая для каждого угла определяется с помощью одного или двух радиусов, определяющих его форму – круга или эллипса. Радиус распространяется на весь фон, даже если элемент не имеет границ, точное положение секущей определяется с помощью свойства `background-clip`.

Свойство `border-radius` позволяет закруглить все углы одновременно. С помощью свойств `border-top-left-radius`, `border-top-right-radius`, `border-bottom-right-radius`, `border-bottom-left-radius` можно закруглить каждый угол отдельно.

Если задать два значения для свойства `border-radius`, то первое закруглит верхний левый и нижний правый углы, а второе – верхний правый и нижний левый.

Значения, заданные через `/`, определяют горизонтальные и вертикальные радиусы. Свойство не наследуется.

border-radius

| (border-top-left-radius, border-bottom-left-radius) | border-top-right-radius, | border-bottom-right-radius, |
|--|---|-----------------------------|
| Значения | Описание | |
| длина | Позволяет закруглить углы блока с помощью значений единиц длины | |
| % | Значения, закругленные углы, задаются в процентах от длины и ширины сторон элемента | |
| initial | Устанавливает значение свойства в значение по умолчанию | |
| inherit | Наследует значение свойства от родительского элемента | |

```
div {width: 100px; height: 100px; border: 5px solid;}
.r1 {border-radius: 0 0 20px 20px;}
.r2 {border-radius: 0 10px 20px;}
.r3 {border-radius: 10px 20px;}
.r4 {border-radius: 10px/20px;}
.r5 {border-radius: 5px 10px 15px 30px/30px 15px 10px 5px;}
.r6 {border-radius: 10px 20px 30px 40px/30px;}
.r7 {border-radius: 50%;}
.r8 {border-top: none; border-bottom: none; border-radius: 30px/90px;}
.r9 {border-bottom-left-radius: 100px;}
.r10 {border-radius: 0 100%;}
.r11 {border-radius: 0 50% 50% 50%;}
.r12 {border-top-left-radius: 100% 20px; border-bottom-right-radius: 100% 20px;}
```

Пример закругления углов представлен на рис. 2.

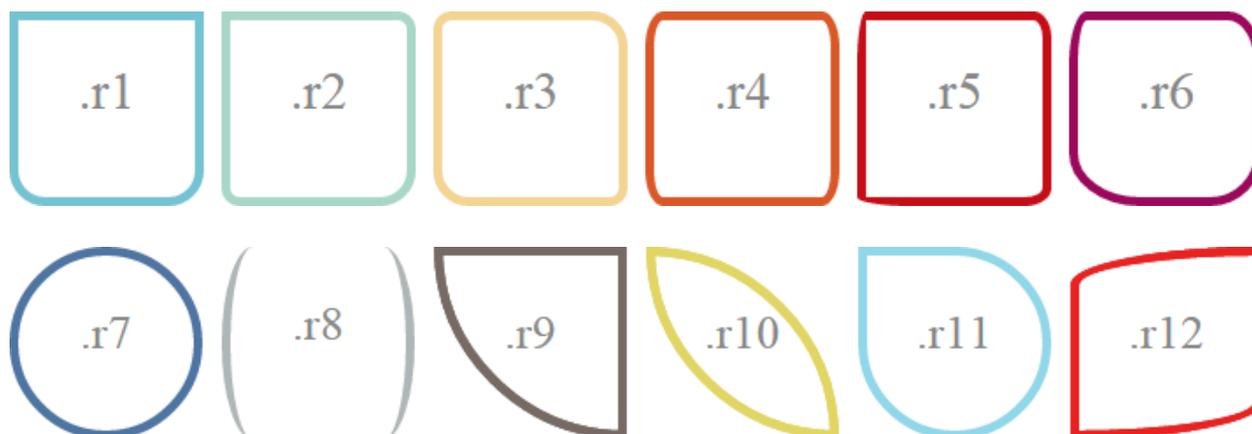
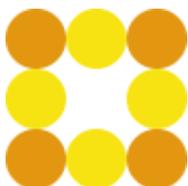


Рис. 2. Примеры закруглений углов блока

Рамки-изображения border-image

Свойство позволяет устанавливать изображение в качестве рамки элемента. Основное требование, предъявляемое к изображению, – оно должно быть симметричным. Свойство включает в себя следующие значения: {border-image: width source slice repeat outset;}

| border-image | |
|----------------|--|
| Значения | Описание |
| краткая запись | Устанавливает рамку-изображение с помощью одного свойства, являющегося краткой записью свойств border-image-source, border-image-slice, border-image-width, border-image-outset и border-image-repeat. Значения по умолчанию: {border-image: 1 none 100% stretch 0;} |
| initial | Устанавливает это свойство в значение по умолчанию |
| inherit | Наследует значение этого свойства от родительского элемента |



При помощи данного изображения можно получить рамки для элемента.

```
/* пример 1 */
div {
width: 260px; height: 100px;
border-image-width: 15px;
border-image-source: url(http://html5book.ru/images/border_round.png);
border-image-slice: 30;
border-image-repeat: stretch;
}
/* Пример 2 */
div {
width: 260px; height: 100px;
border-image-width: 15px;
border-image-source: url(http://html5book.ru/images/border_round.png);
border-image-slice: 30;
border-image-repeat: round;
}
```

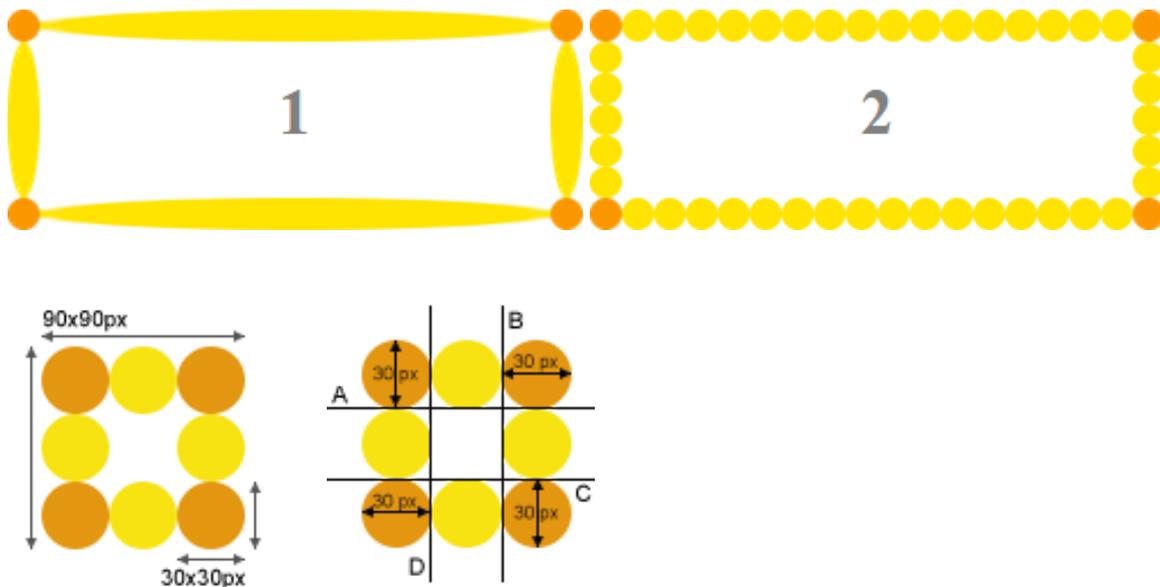


Рис. 3. Пример оформления блоков с помощью изображения

Срезы А – В – С – D образуют углы рамки, а часть рисунка, расположенная между ними, заполняет оставшееся пространство рамки в соответствии с заданным значением свойства `border-image-repeat`.

Размер угловой части (в данном примере это число 30), задается с помощью значения свойства `border-image-slice`.

Ширина рамки-изображения `border-image-width`

Свойство задает ширину изображения для границы элемента. Если ширина не задана, то по умолчанию она равна 1.

| border-image-width | |
|---------------------------|---|
| Значения | Описание |
| длина | Устанавливает ширину рамки в единицах длины – px / em. Можно задавать от одного до четырех значений одновременно. Если задано одно значение, то ширина всех сторон рамки одинакова, два значения задают ширину верхней-нижней и правой-левой и т.д. |
| число | Числовое значение, на которое умножается значение <code>border-width</code> |
| % | Ширина рамки элемента вычисляется относительно размера изображения: горизонтальные – относительно ширины, вертикальные – относительно высоты |
| auto | Соответствует значению <code>border-image-slice</code> |
| initial | Устанавливает это свойство в значение по умолчанию |
| inherit | Наследует значение этого свойства от родительского элемента |

Синтаксис:

```
div {  
border-image-source: url(border.png);  
border-image-width: 30px;  
}
```

Ресурс рамки-изображения border-image-source

Свойство задает путь к изображению, которое будет использоваться для границы блока.

| border-image-source | |
|---------------------|---|
| Значения | Описание |
| none | Отсутствие изображения для рамки. Значение по умолчанию |
| url(url) | Относительный или абсолютный путь к изображению |
| initial | Устанавливает это свойство в значение по умолчанию |
| inherit | Наследует значение этого свойства от родительского элемента |

Синтаксис:

```
div {border-image-source: url(border.png);}
```

Элементы рамки-изображения border-image-slice

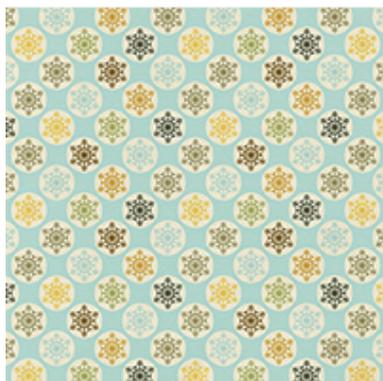
Свойство определяет размер частей изображения, используемых для оформления границ элемента, и делит изображение на девять частей: четыре угла, четыре края между углами и центральную часть.

| border-image-slice | |
|--------------------|--|
| Значения | Описание |
| число | Размер частей рамки можно задавать с помощью одного, двух, трех или четырех значений Одно значение устанавливает границы одинакового размера для каждой стороны элемента Два значения: первое определяет размер верхней и нижней границы, второе – правой и левой. Три значения: первое определяет размер верхней границы, второе – правой и левой, а третье – нижней границы. Четыре значения: определяют размеры верхней, правой, нижней и левой границы. Числовое значение представляет количество px. |
| % | Размеры границ рассчитываются относительно размера изображения: горизонтальные – относительно ширины, вертикальные – относительно высоты |

| | |
|---------|---|
| fill | Значение указывается вместе с числом или процентным значением. Если оно задано, изображение не обрезается внутренним краем рамки, а заполняет также область внутри рамки. |
| initial | Устанавливает это свойство в значение по умолчанию. |
| inherit | Наследует значение этого свойства от родительского элемента. |

Синтаксис:

```
div {border-image-slice: 50 20;}
```



border.png
размер 200x200px

```
div {
width: 200px;
height: 200px;
border-image-width: 40px;
border-image-source:
url(border.png);
border-image-slice: 40;
border-image-repeat: round;
}
```



<div></div>

Рис. 4. Пример задания срезов изображения

Повтор рамки-изображения border-image-repeat

Свойство управляет заполнением фоновым изображением пространства между углами рамки. Можно задавать как с помощью одного значения, так и с помощью пары значений.

| border-image-repeat | |
|---------------------|---|
| Значения | Описание |
| stretch | Растягивает изображение на все пространство. Значение по умолчанию |
| repeat | Повторяет заполняющую часть изображения, при этом видны места стыков с угловой частью, и если длины образца не хватает, то он растягивается |
| round | Наиболее точно заполняет промежуток между углами рамки, дублируя заполняющую часть изображения, при этом может образовывать стыки по середине стороны рамки |
| space | Действует аналогично с repeat |
| initial | Устанавливает это свойство в значение по умолчанию |
| inherit | Наследует значение этого свойства от родительского элемента |

Синтаксис:

```
div {  
border-image-source: url(border.png);  
border-image-repeat: repeat;  
}
```



Рис. 5. Пример повтора центральной рамки изображения

Смещение рамки-изображения border-image-outset

Свойство позволяет переместить изображение-рамку за пределы границ элемента на указанную длину. Может задаваться с помощью одного и четырех значений.

| border-image-outset | |
|---------------------|--|
| Значения | Описание |
| длина | Отступ рамки-изображения, задается с помощью любого положительного числа, указанного в px или em |
| число | Числовое значение, на которое умножается border-width |
| initial | Устанавливает это свойство в значение по умолчанию |
| inherit | Наследует значение этого свойства от родительского элемента |

Синтаксис:

```
div {  
border-image-source: url(border.png);  
border-image-outset: 10px;  
}
```

Смещение внешней рамки outline-offset

Свойство задает расстояние между границей элемента border и внешней границей, созданной с помощью свойства outline.

| outline-offset | |
|----------------|---|
| Значения | Описание |
| длина | Задаёт расстояние с помощью единиц длины – px / em. Отрицательное значение отображает рамку внутри элемента, положительное – снаружи элемента |
| initial | Устанавливает это свойство в значение по умолчанию |
| inherit | Наследует значение этого свойства от родительского элемента |

Синтаксис:

```

/*Рисунок 1:*/
img {
border: 1px solid pink;
outline: 1px dashed grey;
outline-offset: 3px;
}
/*Рисунок 2:*/
img {
border-width: 1px 10px;
border-style: solid;
border-color: pink;
outline: 1px dashed grey;
outline-offset: 3px;
}
/*Рисунок 3:*/
img {
border: 3px inset pink;
outline: 1px dashed grey;
outline-offset: 1px;
}

```

Градиентная рамка

Значением border-image может выступать не только изображение, но и градиентная заливка.

В качестве одного из цветов выступает transparent. Таким способом можно задавать границы сразу для всех сторон элемента или для каждой по отдельности. Толщина рамки регулируется свойством border-width.

Синтаксис:

```
<div class="wrap"><div class="gradient"></div></div>
```

Синтаксис:

```
.wrap {
height: 200px;
padding: 25px;
}
```

```
background: #00E4F6;
}
.gradient {
height: 150px;
width: 50%;
margin: 0 auto;
border-width: 10px;
border-image: linear-gradient(to right, transparent 0%, #ADF2F7 100%);
border-image-slice: 1;
}
```

Другой вариант задания градиента.

Синтаксис:

```
<div class="wrap"><div class="gradient"></div></div>
```

Синтаксис:

```
{box-sizing:border-box;}
.wrap {
height: 200px;
padding: 25px;
}
.gradient {
height: 150px;
width: 50%;
margin: 0 auto;
border: 10px solid transparent;
border-image: 10 repeating-linear-gradient(45deg, #A7CECC, #A7CECC 10px,
transparent 10px, transparent 20px, #F8463F 20px, #F8463F 30px,transparent
30px, transparent 40px);
}
```



Практическое задание

1. Доделать то, что не получилось на предыдущих уроках.
2. Проверить сайт на кроссбраузерность.
3. Проверить сайт на валидность.

4. * Создать адаптивную главную страницу index_992.psd.

Дополнительные материалы

1. @media – <https://developer.mozilla.org/ru/docs/Web/CSS/@media>.
2. 14 инструментов для проверки кроссбраузерности сайта – <http://ruseller.com/lessons.php?rub=28&id=1209>.
3. CSS3 Generator – <http://css3generator.com/CSS3>.
4. Animation Generator – <http://webcodetools.com/css-generator/animation>.
5. Кривые Безье – <https://matthewlein.com/ceaser/>.

Используемая литература

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. <http://www.wisdomweb.ru/>.
2. <http://html5book.ru/css3-transform/>.
3. Гоше Х. HTML5. Для профессионалов. – СПб.: Питер, 2013. – 496 с.: ил.
4. Хоган Б. HTML5 и CSS3. Веб-разработка по стандартам нового поколения. – СПб.: Питер, 2012. – 272 с.
5. Макфарланд Д. Большая книга CSS3. – СПб.: Питер, 2016. – 608 с.