федеральное государственное автономное образовательное учреждение высшего образования
«Московский политехнический университет»

Лекция №2 Основы CSS

Преподаватель: Шульга М.В

Москва 2021

Стиль - это набор параметров, задающий внешнее представление некоторого объекта в той или иной среде.

CSS - это сокращение от **Cascading Style Sheets** - в переводе Каскадные таблицы стилей. Уникальное изобретение человечества, значительно облегчающее создание веб-сайтов. CSS работает со шрифтами, полями, таблицами, отступами, картинками и др. и представляет значительно более широкие возможности, чем простой html.

Основные преимущества CSS:

- Управление дизайном любого количества документов с помощью одной таблицы стилей;
- более точный дизайн страниц, поддерживаемый всеми браузерами;
- разделение документа на две составляющие: структура и дизайн, благодаря чему исходный код становится чистым и легко читаемым
- новые расширенные возможности по сравнению с обычным html

Синтаксис и принцип работы CSS

Как и любой другой язык программирования, CSS имеет строго определенный синтаксис, т.е. правила по которым создаются таблицы стилей. Запомните, в CSS в отличие от HTML нет ни элементов, ни атрибутов, ни тегов. Основной структурной единицей здесь является правило, которое определяет, как будет выглядеть тот или иной элемент в документе.

Рассмотрим структуру правила:

Как видно из рисунка выше, сначала записывается так называемый **селектор**, показывающий к какому html тегу(тегам) применяется то или иное свойство.

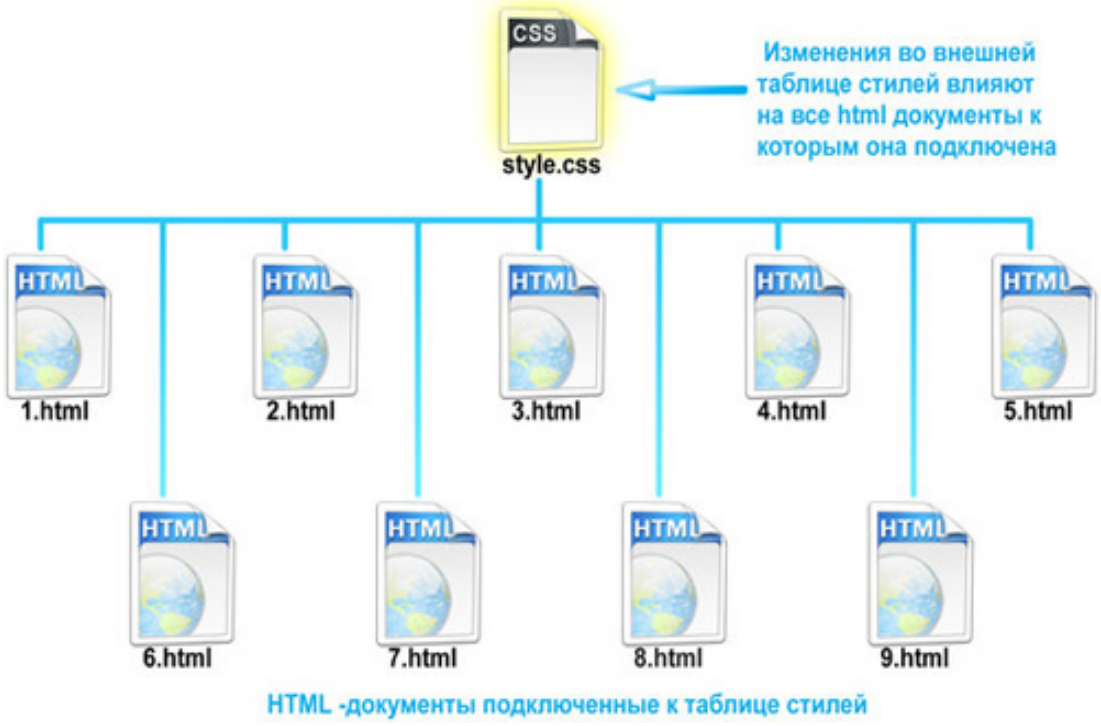
Далее, непосредственно за селектором, пишется **блок объявления стилей**, который обязательно заключается в фигурные скобки.

Каждое объявление в свою очередь состоит из **свойства** и его **значения**. После свойства ставится двоеточие. Правило может содержать в себе несколько объявлений. В таком случае они должны быть отделены друг от друга точкой с запятой(см.рисунок) причем после последнего объявления точку с запятой можно не ставить.

Показанное выше правило указывает на то, что все заголовки первого уровня в документе будут голубого цвета с размером шрифта 14 пикселей.

Как подключить CSS таблицу к HTML документу?

Как вы уже знаете, вся фишка css в том, что меняя стилевые правила во внешней таблице стилей, мы можем управлять дизайном сколь угодно большого количества страниц.



HTML -документы подключенные к таблице стилей

Но для этого нам нужно подключить внешнюю таблицу стилей ко всем страницам html, дизайном которых мы хотим управлять. Давайте по-порядку:

Внешняя таблица стилей это просто текстовый файл с расширением **.css**.

Допустим у нас есть таблица стилей **style.css** и несколько страниц **html,** и причем все это расположено в одной папке. Тогда в каждом документе который мы хотим подключить, в голове документа(между тегами <head> и </head>) необходимо прописать строчку:

```
<link rel="stylesheet" type="text/css" href="style.css">
```

Эта строка указывает браузеру, что он должен использовать правила отображения HTML-файла из CSS-файла.

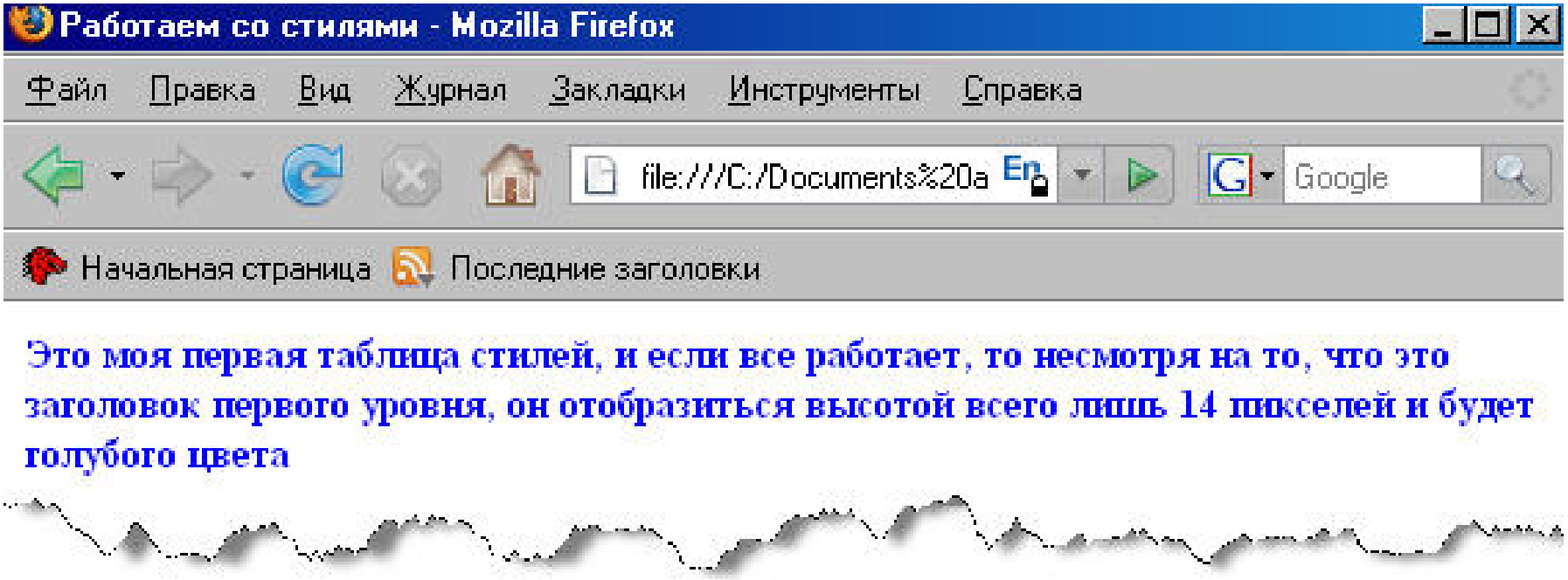
Пример: (index.htm)

```
<!DOCTYPE HTML>
<html>
<head>
<title>Работаем со стилями</title>
<link rel="stylesheet" type="text/css" href="style.css" >
</head>
<body>
<h1>Это моя первая таблица стилей, и если все работает, то несмотря на то, что это заголовок первого уровня, он отобразится высотой всего лишь 14 пикселей и будет голубого цвета</h1>
</body>
</html>
```

Пример: (style.css)

```
h1{color:blue;font-size:14px}
```

Если вы все сделали правильно, то запустив в браузере файл index.htm увидите :



Можно располагать правила стилей непосредственно в голове документа:

```
<!DOCTYPE HTML"
<html>
<head>
<title>Работаем со стилями</title>
<style type="text/css">
h1{color:blue;font-size:14px}
</style>
</head>
<body>
<h1>Это моя первая таблица стилей, и если все работает, то несмотря на то, что это заголовок первого уровня, он отобразиться высотой всего лишь 14 пикселей и будет голубого цвета</h1>
</body>
</html>
```

Также, можно задавать стиль с помощью атрибута style .
Например:

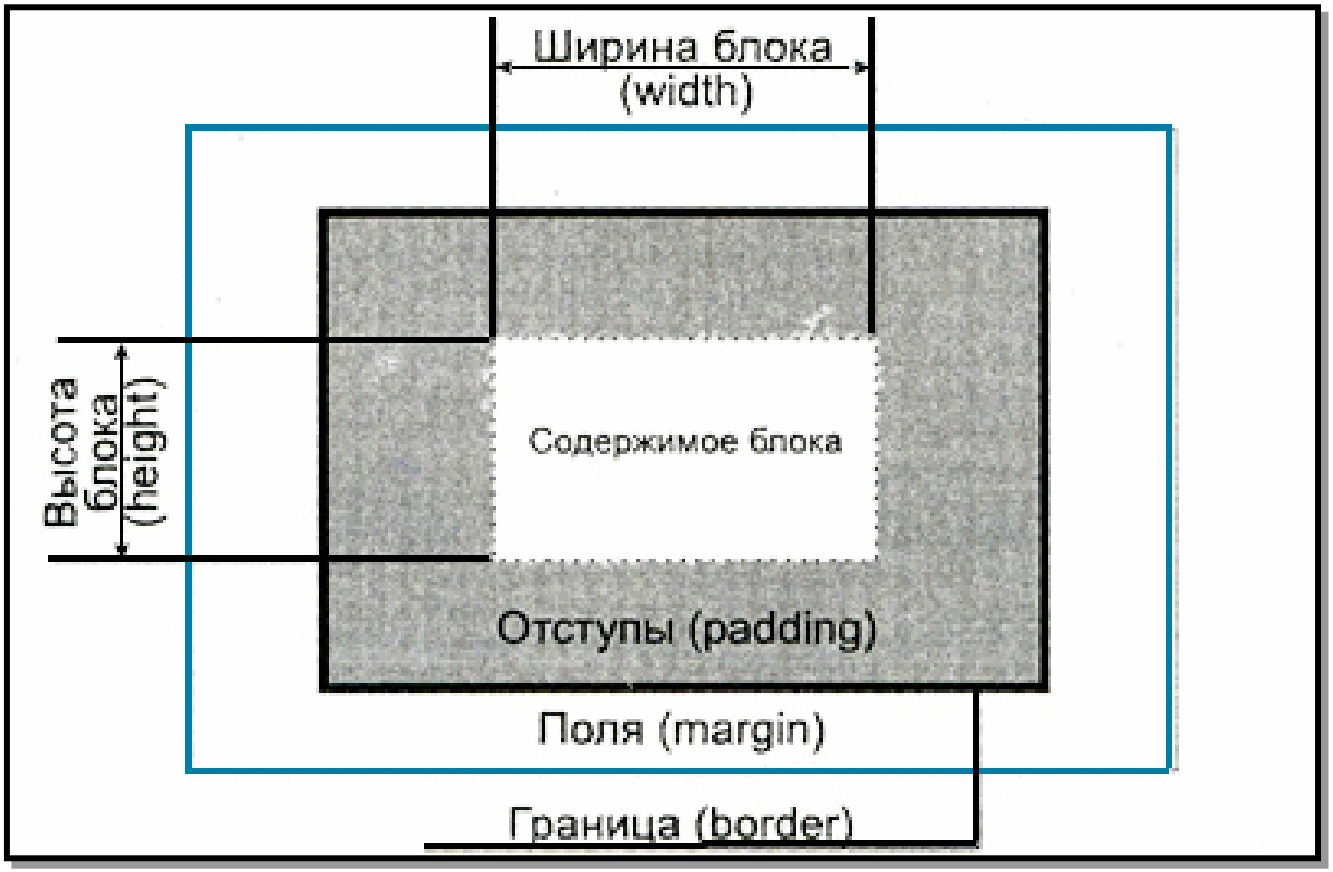
```
<!DOCTYPE HTML>
<html>
<head>
<title>Работаем со стилями</title>
</head>
<body>
<h1 style="color:blue;font-size:14px">Это моя первая таблица стилей, и если все работает, то несмотря на то, что это заголовок первого уровня, он отобразиться высотой всего лишь 14 пикселей и будет голубого цвета</h1>
</body>
</html>
```

Блоковая модель в CSS

В html различают элементы блочные и строчные. Причем блочные элементы это как бы отдельная структурная единица, которая всегда отделяется абзацными отступами. Т.е. нельзя расположить два блочных элемента на одной строке. Примером блоков в html могут служить элементы **H1-H6, P, DIV**. А строчные элементы не создают как-бы отдельной структурной единицы, не отделяются абзацными отступами, и например два строчных элемента могут без проблем расположиться на одной строке. Примером могут быть элементы **strong , EM , I.**

Технология стилей CSS также использует понятие блоков. Блоки в CSS представляют собой самостоятельную структурную единицу, имеющую форму прямоугольника.

Каждый элемент в дереве элементов документа – это самостоятельный блок. Из этого следует, что в CSS есть блоки, которые структурно отделены от остальных, а есть строчные блоки, которые могут находиться внутри структурных блоков. Но и те, и другие имеют одинаковую структуру:



Каждый такой блок обязательно имеет информационную часть, или содержимое, которым может быть текст, изображение или другая информация. Эта часть блока называется его контентом или содержимым. Например, для элемента **P** содержимым блока является текст абзаца.

Вокруг области контента могут быть пустые, не занятые содержимым области, называемые полями **(margin)**. С точки зрения дизайна поля обеспечивают для содержимого блока эстетически более привлекательный вид. При наличии полей определенного размера текст не примыкает вплотную к границам блока. Можно провести аналогию с полями, устанавливаемыми при печати документов на бумаге. Если полей не было бы, то текст начинался бы прямо у края листа.

Непосредственно за полями проходит граница блока (**border**), которая может иметь определенную толщину и стиль линий. Ширина блока может быть произвольной - от нулевой (граница в этом случае не видна) до произвольно заданной в единицах измерениях длины. Стиль линий может быть различным, от простой линии до объемных вариантов. Кроме того, граница может иметь произвольный цвет.

Также блок может иметь отступы (**padding**), это дополнительное свободное пространство вокруг границы блока. Согласно спецификации CSS, поля, границы и отступы не входят в ширину блока. Таким образом, указывая ширину блока, вы задаете ширину лишь той части блока, которая отведена для содержимого.

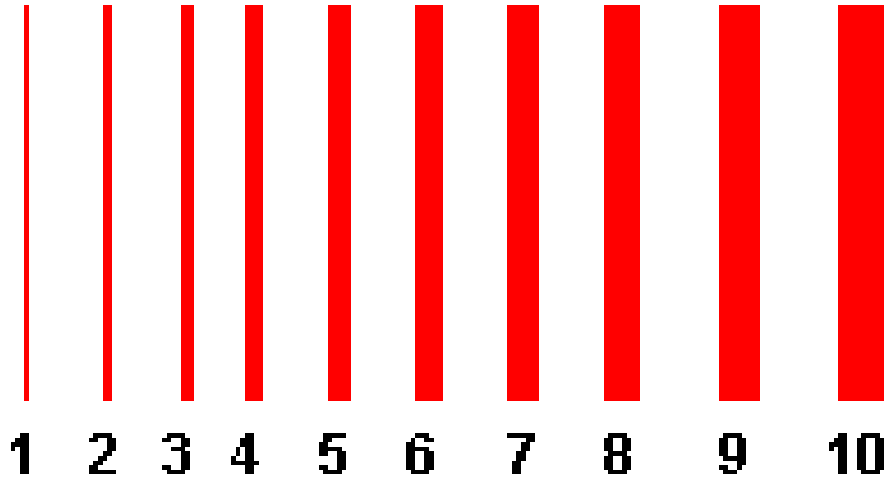
Рамки в CSS

Основные свойства рамок в CSS следующие:

- border-width
- border-color
- border-style
- Сокращенная форма - border
- Примеры

Свойство BORDER-WIDTH

Это свойство задает толщину рамки. Значение обычно указывается в пикселях, но также можно указывать ключевыми словами **thin** (2px) , **medium**(4px) и **thick**(6px). Для лучшего понимания, сколько это один пиксель, смотрите рисунок ниже:



На рисунке приведены значения ширины от 1 до 10 пикселей.

Свойство BORDER-COLOR

Как вы поняли данное свойство определяет цвет рамки. Значение цвета задается обычным образом, т.е. например: "#ff3344", или "gold".

Свойство BORDER-STYLE

Данное свойство определяет какого вида будет рамка. Ниже приведены 8 основных значений данного свойства. Все рамки в примере выполнены цветом gold и шириной 6 px .

SOLID Рамка состоит из сплошной линии

DOTTED Точечная рамка

DASHED	Пунктирная рамка
DOUBLE	Рамка из двойной сплошной линии
GROOVE	Рамка как бы из вдавленной линии, с имитацией объема
RIDGE	Рамка отображается выпуклой линией с имитацией объема
INSET	Рамка отображается так, что весь блок кажется вдавленным

OUTSET

Рамка отображается так, что весь блок кажется выпуклым

Примечание: минимальная ширина рамки типа **double** должна равняться **3 px**, иначе она будет отображаться некорректно.

ПРИМЕР 1:

```
h1 {
border-width: 4px;
border-style: dotted;
border-color: red;
}
h2 {
border-width: 18px;
border-style: inset;
border-color: red;
}
p {
border-width: 2px;
border-style: solid;
border-color: blue;
}
```

ПРИМЕР 1:

Вот такая рамка у заголовка 1-го уровня

Вот такая рамка у заголовка 2-го уровня

Такие рамки у параграфов. Такие рамки у параграфов. Такие рамки у параграфов. Такие рамки у параграфов. Такие рамки у параграфов. Такие рамки у параграфов. That doesn't mean they are easier to promote, build traffic and turn a profit.Both websites and blogs require work, patience, an understanding of Internet marketing and search engine optimization (the process of getting ranked high in Google, Yahoo, etc.) Both websites and blogs

That doesn't mean they are easier to promote, build traffic and turn a profit.Both websites and blogs require work, patience, an understanding of Internet marketing and search engine optimization (the process of getting ranked high in Google, Yahoo, etc.) Both websites and blogs require work, patience, an understanding of Internet marketing and search engine optimization (the process of getting ranked high in Google, Yahoo, etc.) Both websites and blogs require work, patience, an understanding of Internet marketing and search engine optimization (the process of getting ranked high in Google, Yahoo, etc.)

SThat doesn't mean they are easier to promote, build traffic and turn a profit.Both websites and blogs require work, patience, an understanding of Internet marketing and search engine optimization (the process of getting ranked high in Google, Yahoo, etc.) Both websites and blogs require work, patience, an

So please don't go into this thinking that creating a blog is an easier way to make money online. Even bloggers making a lot of money have put in their share of time and effort.So please don't go into this thinking that creating a blog is an easier way to make money online. Even bloggers making a lot of money have put in their share of time and effort.

hat doesn't mean they are easier to promote, build traffic and turn a profit.Both websites and blogs require work, patience, an understanding of Internet marketing and search engine optimization (the process of getting ranked high in Google, Yahoo, etc.) Both websites and blogs require work, patience, an understanding of Internet marketing and search engine optimization (the process of getting ranked high in Google, Yahoo, etc.) Both websites and blogs require work, patience, an understanding of Internet marketing and search engine optimization (the process of getting ranked high in Google, Yahoo, etc.)

That doesn't mean they are easier to promote, build traffic and turn a profit.Both websites and blogs require work, patience, an understanding of Internet marketing and search engine optimization (the process of getting ranked high in Google, Yahoo, etc.) Both websites and blogs require work, patience, an understanding of Internet marketing and search engine optimization (the process of getting ranked high in Google, Yahoo, etc.) Both websites and blogs require work, patience, an understanding of Internet marketing and search engine optimization (the process of getting ranked high in Google, Yahoo, etc.)

SThat doesn't mean they are easier to promote, build traffic and turn a profit.Both websites and blogs require work, patience, an understanding of Internet marketing and search engine optimization (the process of getting ranked high in Google, Yahoo, etc.) Both websites and blogs require work, patience, an

ПРИМЕР 2:

```
h1 {border-width: 30px;
border-style:outset ;
border-color: red;
}
h2 {border-width: 20px;
border-style: dashed;
border-color: gold;
}
h3 {border-width: 16px;
border-style: double;
border-color: green;
}
p {border-width: 1px;
border-style: dotted;
border-color: blue;
}
```

ПРИМЕР 2:

Вот такая рамка у заголовка 1-го уровня

Вот такая рамка у заголовка второго уровня

Такая у заголовка третьего уровня

Такие рамки у параграфов. Такие рамки у параграфов. Такие рамки у параграфов. Такие рамки у параграфов. Такие рамки у параграфов. Такие рамки у параграфов. Такие рамки у параграфов. That doesn't mean they are easier to promote, build traffic and turn a profit.Both websites and blogs require work, patience, an understanding of Internet marketing and search engine optimization (the process of getting ranked high in Google, Yahoo, etc.) Both websites and blogs

Такие рамки у параграфов. Такие рамки у параграфов. Такие рамки у параграфов. Такие рамки у параграфов. Такие рамки у параграфов. Такие рамки у параграфов. Такие рамки у параграфов. That doesn't mean they are easier to promote, build traffic and turn a profit.Both websites and blogs require work, patience, an understanding of Internet marketing and search engine optimization (the process of getting ranked high in Google, Yahoo, etc.) Both websites and blogs

Такие рамки у параграфов. Такие рамки у параграфов. Такие рамки у параграфов. Такие рамки у параграфов. Такие рамки у параграфов. Такие рамки у параграфов. Такие рамки у параграфов. That doesn't mean they are easier to promote, build traffic and turn a profit.Both websites and blogs require work, patience, an understanding of Internet marketing and search engine optimization (the process of getting ranked high in Google, Yahoo, etc.) Both websites and blogs

Такие рамки у параграфов. Такие рамки у параграфов. Такие рамки у параграфов. Такие рамки у параграфов. Такие рамки у параграфов. Такие рамки у параграфов. Такие рамки у параграфов. That doesn't mean they are easier to promote, build traffic and turn a profit.Both websites and blogs require work, patience, an understanding of Internet marketing and search engine optimization (the process of getting ranked high in Google, Yahoo, etc.) Both websites and blogs

Такие рамки у параграфов. Такие рамки у параграфов. Такие рамки у параграфов. Такие рамки у параграфов. Такие рамки у параграфов. Такие рамки у параграфов. Такие рамки у параграфов. That doesn't mean they are easier to promote, build traffic and turn a profit.Both websites and blogs require work, patience, an understanding of Internet marketing and search engine optimization (the process of getting ranked

Сокращенная форма - border

Как и в других свойствах, у рамки есть сокращенная форма border . В начале пишется толщина, затем после пробела стиль, а затем после пробела цвет. Предыдущий пример можно записать так:

```
h1 {border:30px outset red;}
h2 {border:20px dashed gold;}
h3 {border: 16px double green;}
P {border:1px dotted
```

Примеры:

Во всех перечисленных выше примерах, если добавить после слова **border** одно из ключевых слов (top, right, bottom, left) можно регулировать параметры рамки с разных сторон соответственно (верх, право, низ, лево). Ну вот например можно сделать так:

```
h1 {
border-top-width: 30px;
border-top-style:solid ;
border-top-color: red;

border-right-width: 20px;
border-right-style:dashed ;
border-right-color: gold;

border-bottom-width: 10px;
border-bottom-style:dashed;
border-bottom-color: green;

border-left-width: 40px;
border-left-style:solid ;
border-left-color: blue;
}
```

Результат:

Естественно намного красивее код будет выглядеть в сокращенном виде:

```
h1 {
border-top: 30px solid red;
border-right: 20px dashed gold;
border-bottom: 10px dashed green;
border-left: 40px solid blue;
}
```

Можно также комбинировать перечисленные выше свойства, ну например так:

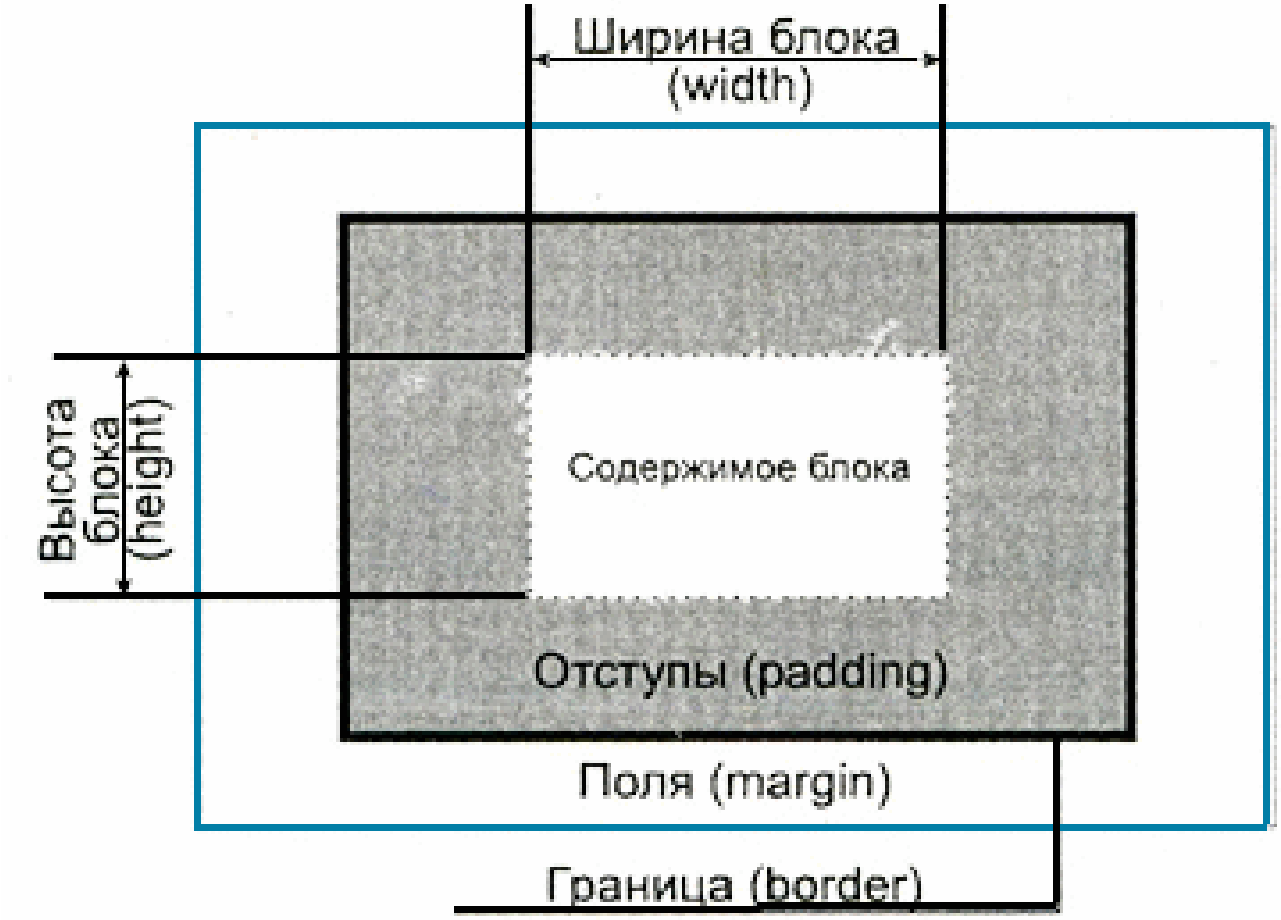
```
h1 {
border: 30px solid red;
border-bottom: 10px solid gold;
}

h2 {
border: 30px solid red;
border-bottom-color: green
}
```

Результат:

Поля (margin) и отступы (padding)

Давайте посмотрим, в чем отличие margin от padding. Для этого, еще разок вспомним блоковую модель в CSS.



MARGIN (Поля) - это расстояние от границы(рамки) блока, до ближайших элементов, или, если их нет, до краев документа.
PADDING (Отступы) - как бы внутреннее расстояние, между границей(рамкой) и содержимым блока.
Пример: создадим три стиля для трех разных параграфов, с различными значениями margin и padding и посмотрим на результат:

```
.p1 {background-color : #FFE446;border:1px solid red;margin:70px;}
.p2 {background-color : #FFE446;border:1px solid red;padding:70px;}
.p3 {background-color : #FFE446;border:1px solid red;margin:50px;padding:20px;}
```

Результат: Поля (margin) и отступы (padding)

Добавляя уже знакомые нам ключевые слова: top, right, bottom, left можно регулировать отступы и поля соответственно сверху, справа, снизу, слева.

```
p {
margin-top:50px;
margin-right:50px;
margin-bottom:50px;
margin-left:150px;
}
```

Результат: Поля (margin) и отступы (padding) 2

Тоже самое, только в более сокращенной записи:

```
p {margin:50px;margin-left:150px;}
```

Т.е. так как только левое поле отличается от остальных, мы просто записали общее поле, а потом дописали значение левого поля, и получили тот же результат что и в первом примере.

Возможен также такой вариант записи:

```
p {margin: 50px 50px 50px 150px;}
```

Т.е. значения записываются по часовой стрелке: верхнее, правое, нижнее, левое.

В каких случаях лучше пользоваться отступами, а в каких полями?

Несколько принципиальных отличий:

Отступы(padding) располагаются внутри блока, а поля(margin) - за их пределами;

Фон блока и фоновое изображение распространяются только на отступы, но не на поля. Поля всегда прозрачны, и сквозь них просвечивает фон родительского элемента.

Концепция сетки и основные понятия

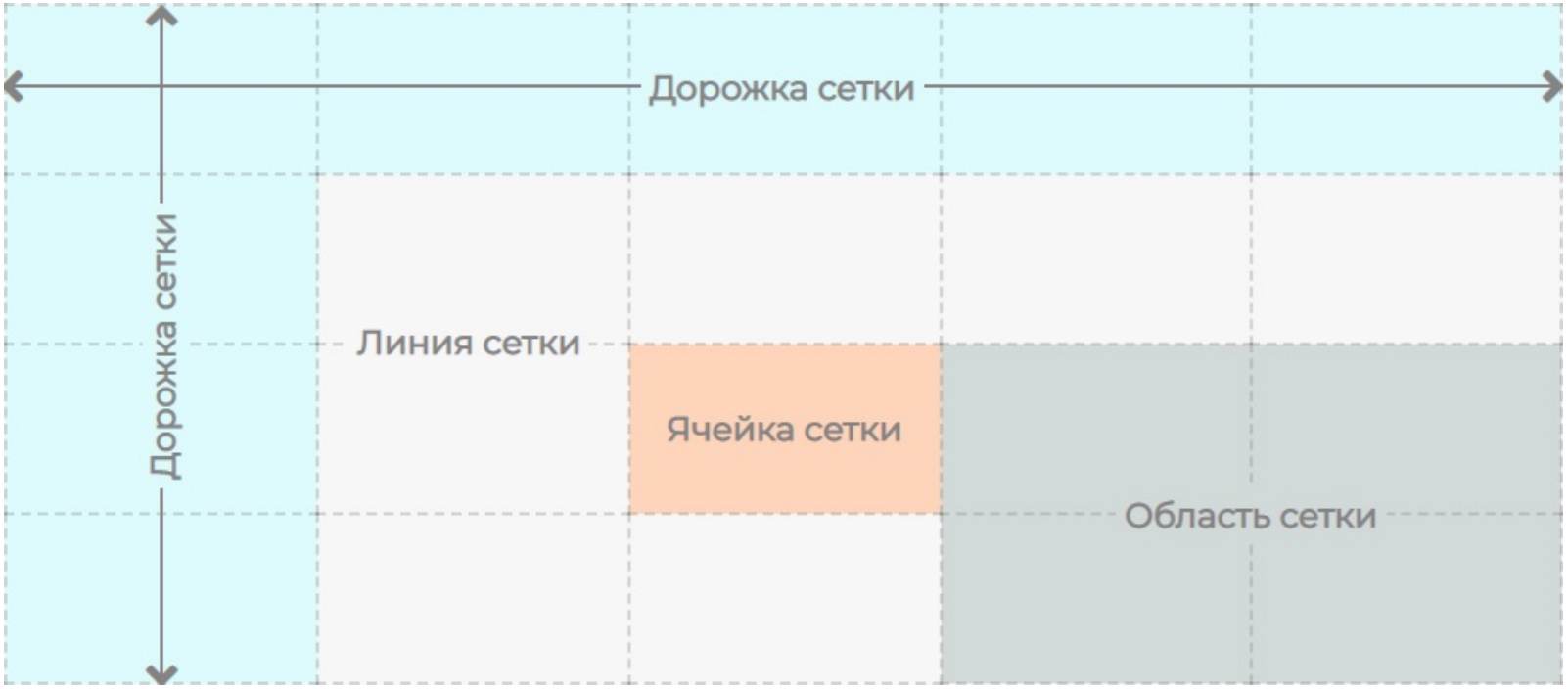


РИС. КОМПОНЕНТЫ СЕТКИ

Сетка (grid) представляет собой набор пересекающихся горизонтальных и вертикальных линий, делящих пространство grid-контейнера на области сетки, 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.

Выделяют две группы линий сетки: одна группа определяет столбцы, которые проходят вдоль оси блока (ось столбцов), и перпендикулярная группа, определяющая строки, простирающиеся вдоль линейной оси (ось строк), в соответствии с CSS3 режимом записи.

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 или даже большинство ячеек сетки будут пустыми от любого содержимого. Также возможно, что элементы сетки будут перекрывать друг друга, либо определять перекрывающиеся области сетки.

Контейнер-сетка

Контейнер-сетка (grid container) — это блок, который устанавливает контекст форматирования по типу сетки. Создает область с сеткой, а дочерние элементы располагаются в соответствии с правилами компоновки сетки.

Когда вы определяете контейнер сетки с помощью display: grid или display: inline-grid, создаете новый контекст форматирования для содержимого этого контейнера, который влияет только на дочерние элементы сетки.

Существует 2 вида: обычный display: grid и встроенный display: inline-grid. Первый генерирует grid-контейнер уровня блока, второй — grid-контейнер уровня строки. Не являются блочными контейнерами, поэтому некоторые CSS-свойства не работают в контексте макета сетки:

float и clear игнорируются элементами сетки, но не самим контейнером-сеткой.

vertical-align не влияет на элементы сетки.

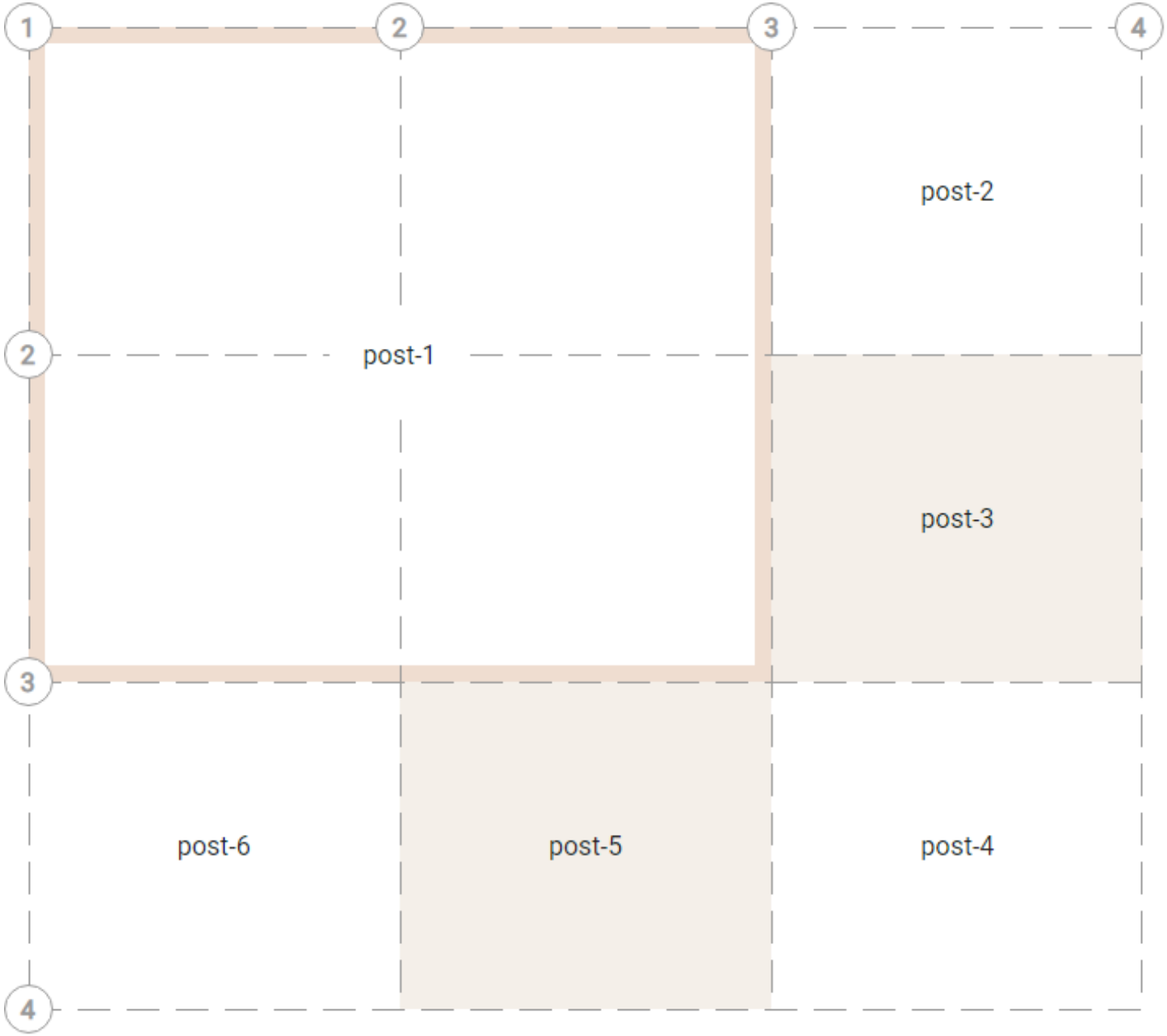
Псевдоэлементы ::first-line и ::first-letter не применяются к контейнеру-сетке и его потомкам.

Вычисляемое значение свойства display будет grid.
Сетка по умолчанию имеет 1 столбец и строку, которые занимают полный размер контейнера. Чтобы разделить контейнер-сетку на столбцы или строки используются свойства grid-template-columns, grid-template-rows и grid-template-areas.

Если сетка больше из-за элементов сетки, размещенных вне явной сетки, то будут созданы неявные дорожки, размер этих неявных дорожек 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 свойств сетки grid-template-columns, grid-template-rows и grid-template-areas. grid сбрасывает свойства, управляющие неявной сеткой, а grid-template оставляет их без изменений.

Чтобы определить количество строк/столбцов имеется свойства grid-template-rows игrid-template-columns. Свойства не наследуется.


1
2
3
4
post-2
2
post-1
post-3
3
post-6
post-5
post-4
4

Размещение элементов сетки с помощью числовых индексов

```
.grid-container {
 display: grid;
 grid-template-rows: 200px 200px 200px;
 grid-template-columns: 1fr 1fr 1fr;
 }
 .post-1 {
 grid-row-start: 1; grid-row-end: 3;
 grid-column-start: 1;
 grid-column-end: 3;
 }
 .post-2 {
 grid-row-start: 1;
 grid-column-start: 3;
 }
 .post-3 {
 grid-row-start: 2;
 grid-column-start: 3;
 }
```

```
.post-4 {
 grid-row-start: 3;
 grid-column-start: 3;
 }
 .post-5 {
 grid-row-start: 3;
 grid-column-start: 2;
 }
 .post-6 {
 grid-row-start: 3;
 grid-column-start: 1;
 }
```