

# Свойства в CSS3

# Типография

CSS предлагает много разных свойств для редактирования внешнего вида текста на странице. Эти свойства входят в две категории: свойства шрифта и свойства текста.



# Типография

## Свойства шрифта:

- font-family - для объявления, какой шрифт, а также какие резервные или заменяющие шрифты должны быть использованы для отображения текста

```
body {  
  font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;  
}
```

# Типография

При разработке убедитесь, что

- набор шрифтов подходит ко всем операционным системам
- набор шрифтов содержит либо все шрифты *sans serif* (без засечек) либо *serif* (с засечками)  
(для согласованности шрифтов)
- шрифты в наборе имеют схожую пропорциональность

# Типография

## Встраивание шрифтов

Есть возможность загружать шрифты с сервера и включать их на сайт через стилевое правило @font-face

1. Используем правило @font-face, чтобы определить имя шрифта через свойство font-family, а также исходник шрифта (путь к файлу, содержащему выбранный шрифт) через свойство src
2. Используем шрифт, включая его имя в качестве значения свойства font-family

```
@font-face {  
  font-family: "Lobster";  
  src: local("Lobster"), url("lobster.woff") format("woff");  
}  
body {  
  font-family: "Lobster", "Comic Sans", cursive;  
}
```

# Типография

- font-size - даёт возможность установить размер текста, используя типовые значения размера, включая пиксели, em, проценты, пункты или ключевые слова

Единицы измерения:

- абсолютные (явно указываем значение в определенных единицах типа пикселей, т.е. точные размеры)
- относительные (их размеры зависят от font-size, применённого к их родительским элементам. Возможные единицы: em, %)

# Типография

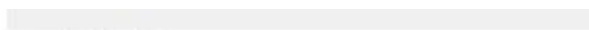
- font-style - задает стиль шрифта, возможные значения:
  - normal (по умолчанию) - обычный текст
  - italic - курсив
  - oblique - наклонное начертание
  - inherit - наследует значение от родительского элемента
- font-variant - определяет, как нужно представлять строчные буквы — оставить их без модификаций или делать их все прописными уменьшенного размера (капитель)
  - normal (по умолчанию) - обычный текст
  - small-caps - капитель
  - inherit - наследует значение от родительского элемента



# Типография

- font-weight - насыщенность (толщина) шрифта. Принимаемые значения:
  - normal (обычный текст), bold (жирный), bolder (жирнее), lighter (тоньше) и inherit (наследуется от родителя)
  - числовые значения 100, 200, 300, ..., 900 (указывают жирность). Значение normal соответствует числу 400, bold - 700

- line-height - межстрочный интервал. Значения в пикселях, %, числах



font-weight: 100;	Thin
font-weight: 200;	Extra-Light
font-weight: 300;	Light
font-weight: 400;	Regular
font-weight: 500;	Medium
font-weight: 600;	Semi-Bold
font-weight: 700;	<b>Bold</b>
font-weight: 800;	<b>Extra-Bold</b>
font-weight: 900;	<b>Black</b>



# Типография

- Универсальное свойство font:

Все свойства шрифта могут быть объединены в одном универсальном свойстве font.

Порядок значений слева направо:

font-style, font-variant, font-weight, font-size, line-height и font-family

```
html {  
  font: italic small-caps bold 14px/22px "Helvetica Neue", Helvetica, Arial,  
  sans-serif;  
}
```

# Цвета

На уровне элементов HTML, всему можно присвоить цвет.

Значения цвета:

- `color: rgb(255,255,255)`
- `color: #FFFFFF;`
- `color: rgba(255, 255, 255, 0.5)`

Свойство `color` определяет цвет текста HTML элемента

Свойство `background-color` определяет цвет фона элемента

# Базовые свойства для работы с текстом

- text-align - выравнивание текста. Возможные значения:
  - left (по левому краю)
  - right (по правому краю)
  - center (по центру)
  - justify (по ширине)
  - inherit (наследуется значение от родителя)

# Базовые свойства для работы с текстом

- text-decoration - декорация текста. Возможные значения:
  - none (убирает эффекты, декорацию)
  - underline (подчеркивание снизу)
  - overline (надчеркивание сверху)
  - line-through (зачеркнутый текст)
  - inherit (наследуется значение от родителя)

# Базовые свойства для работы с текстом

- `text-indent`- для создания красной строки внутри элемента. Положительные значения делают отступ текста внутрь, в то время как отрицательные значения делают отступ наружу.
- `text-shadow` - позволяет добавить тень или несколько теней к тексту. Свойство обычно принимает четыре значения, все они перечисляются друг за другом слева направо. Первые три значения — это размер, а последнее значение — цвет.

# Базовые свойства для работы с текстом

- text-transform - управляет преобразованием текста элемента в заглавные или прописные символы. Принимаемые значения:
  - capitalize (делает заглавной первую букву каждого слова)
  - uppercase (устанавливает заглавной каждую букву)
  - lowercase (делает каждую букву строчной)
  - none (отмена вышеперечисленных значений)
- letter-spacing - регулирует расстояние между буквами
- word-spacing - регулирует расстояние между словами в элементе



# Базовые свойства для работы с текстом

- color - цвет текста и любого оформления текста (подчёркивание, линии на текстом, перечёркивание и т.д.)
- background-color - цвет фона текста

# Блочная модель

Отображение элементов на странице:

## **Блочные элементы**

занимают всю доступную ширину  
независимо от их содержимого и  
начинаются с новой строки

## **Строчные элементы**

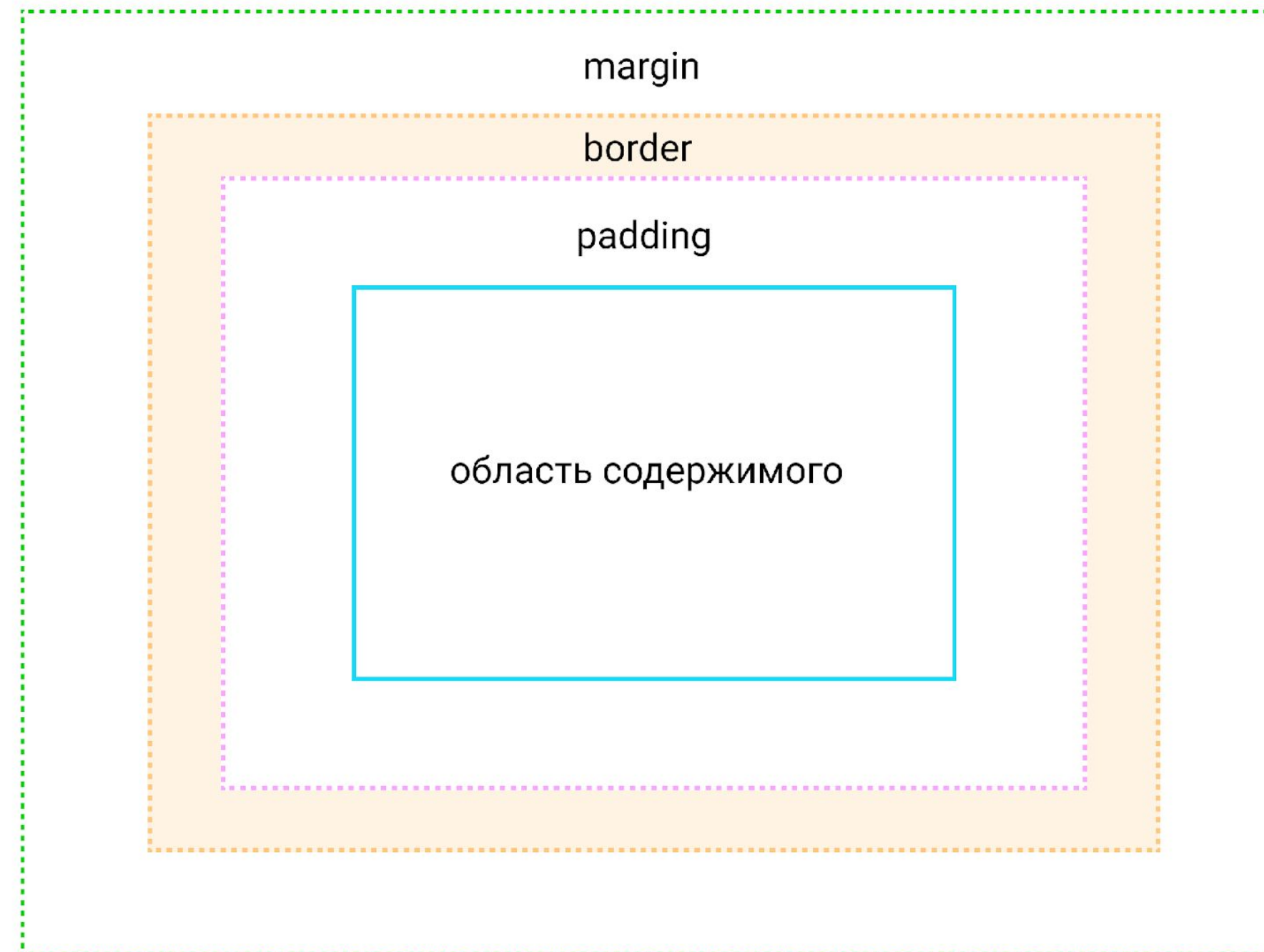
занимают ширину, которая требуется  
для содержимого и выстраиваются на  
одной строке, друг за другом

# Блочная модель

Отображение элементов на странице (блочные элемент/строчный элемент) задается свойством `display`:

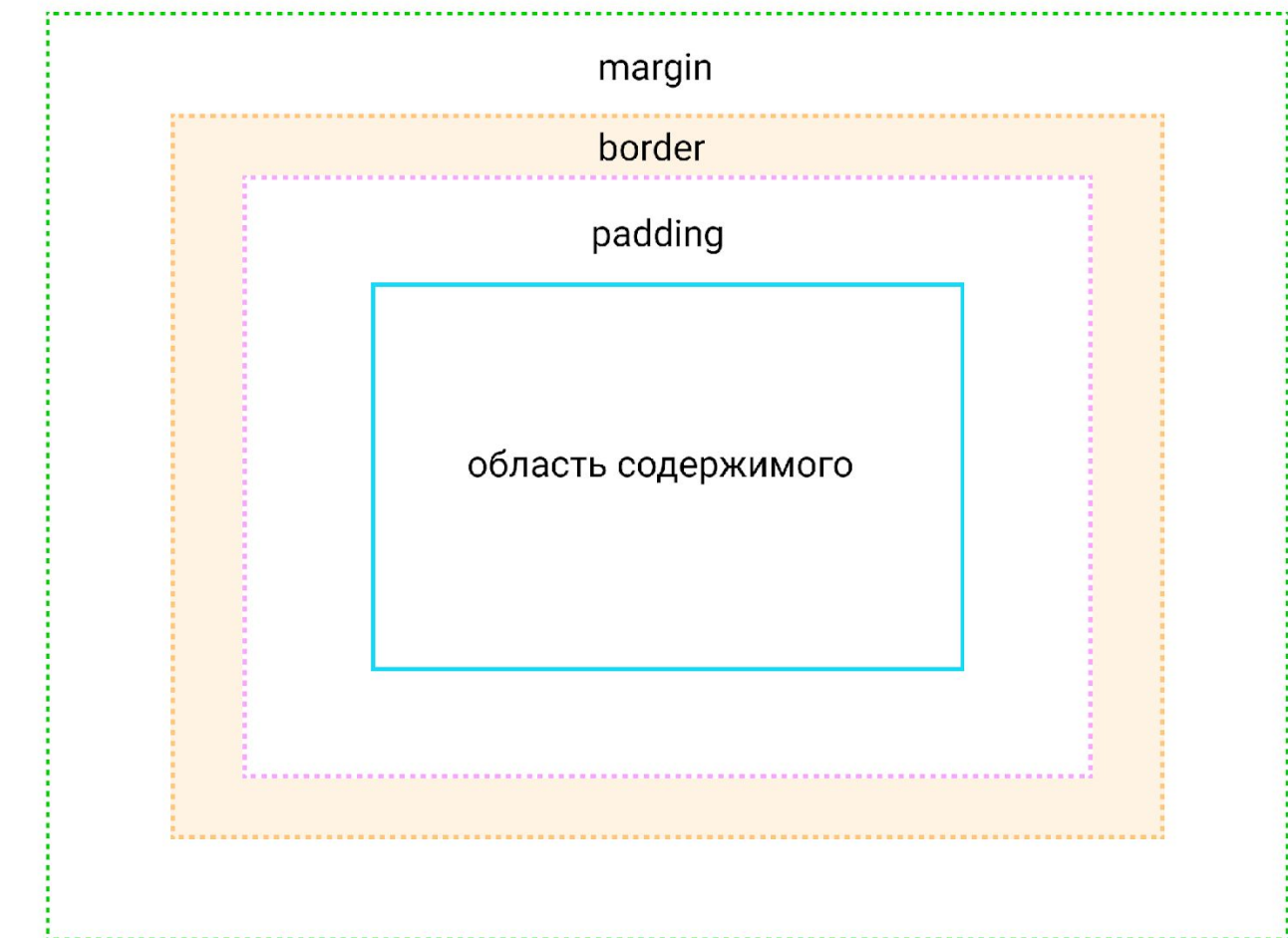
- `display: block` - элемент отображается как блочный
- `display: inline` - элемент отображается как строчный
- `display: inline-block` - элемент ведет себя как блочный, включая все свойства блочной модели (рассмотрим далее). Но элемент будет отображаться на строке с другими элементами, а не будет начинаться с новой строки по умолчанию
- `display: none` - элемент полностью скрывается и страница отображается так, словно этого элемента не существует

# Блочная модель



# Блочная модель

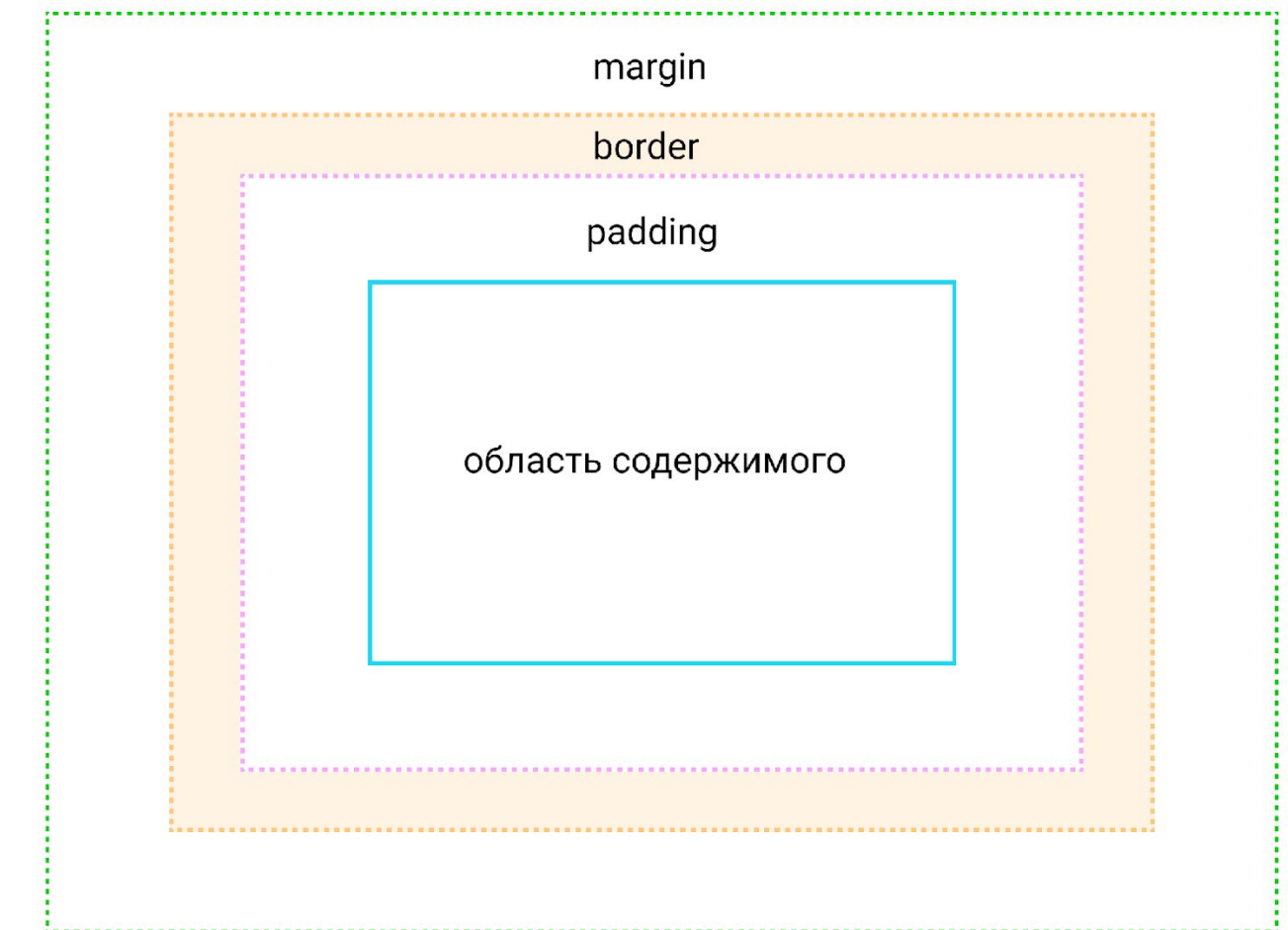
- каждый элемент на странице представляет собой прямоугольный блок и может иметь ширину, высоту, поля, границы и отступы
- у каждого блока есть 4 области: margin (внешние отступы), border (рамка), padding (внутренние поля), и content (контент или содержимое)



# Блочная модель

- размер каждой области определяется соответствующими свойствами и может быть нулевым, или, в случае margin, отрицательным
- Значение полей и отступов по умолчанию равно 0. Но некоторые браузеры добавляют этим свойствам положительные значения по умолчанию на основе своих таблиц стилей. Очистить стили можно при помощи универсального селектора:

```
* {  
  margin: 0;  
  padding: 0;  
}
```





# Отступы элемента

Отступы окружают край рамки элемента, обеспечивая расстояние между соседними блоками. Свойства отступов определяют их толщину. Применяются ко всем элементам, кроме внутренних элементов таблицы.

- `margin-top/margin-left/margin-right/margin-bottom` - задает верхний/левый/правый/нижний отступ блока элемента соответственно (допускаются отрицательные значения)
- `margin` - задает отступы для всех четырех сторон (если указано одно значение: применяется ко всем сторонам, если два значения: верхний и нижний отступы - первое значение, левый и правый - второе значение, если четыре значения - применяются сверху, справа, снизу и слева соответственно)

# Отступы элемента

Схлопывание отступов: смежные вертикальные отступы двух или более элементов в блочной модели схлопываются (перекрываются). При этом ширина общего отступа равна ширине большего из исходных.

Для предотвращения проблемы схлопывания рекомендуется задавать для всех элементов только верхний или нижний margin

Не схлопываются отступы:

- Между плавающим блоком и любым другим блоком
- У плавающих элементов и элементов со значением `overflow`, отличным от `visible`, со своими дочерними элементами в потоке
- У абсолютно позиционированных элементов, даже с их дочерними элементами
- У строчно-блочных элементов

# Поля элемента

Область полей представляет собой пространство между краем области содержимого и рамкой элемента. Свойства полей определяют толщину их области. Применяются ко всем элементам, кроме внутренних элементов таблицы.

- padding-top/padding-left/padding-right/padding-bottom - задает верхнее/левое/правое/нижнее поле блока элемента соответственно (отрицательные значения недопустимы)
- padding - задает поле для всех четырех сторон (если указано одно значение: применяется ко всем сторонам, если два значения: верхнее и нижнее поле - первое значение, левое и правое - второе значение, если четыре значения - применяются сверху, справа, снизу и слева соответственно)

# Границы элемента

Границы располагаются между отступами и полями, создавая рамку вокруг элемента.

Для задания границ используется свойство `border`, которому передаются значения: ширина, стиль и цвет

Значения ширина, стиль и цвет могут быть разбиты на 3 отдельных свойства:

- `border-width` - ширина границ (в абсолютных или относительных единицах)
- `border-style` - внешний вид границ (`solid`, `dashed`, `dotted`, `none`)
- `border-color` - цвет границ

Можно отдельно задавать границы для сторон:

`border-top`/`border-bottom`/`border-left`/`border-right` (верхняя/нижняя/левая/правая граница)

# Границы элемента

Свойство border-radius позволяет закруглять углы элемента. Значение - единицы размера (px, %), которые определяют радиус скругления углов элемента.

Свойство border-radius также может быть разбито на ряд свойств, которые позволяют нам изменить радиусы отдельных углов элемента. Эти свойства начинаются с border, продолжаются с положения угла по вертикали (top или bottom) и горизонтали (left или right) и завершаются radius. Например, для изменения правого верхнего угла <div> может быть использовано свойство border-top-right-radius

# Размеры элементов

- width - ширина элемента (px, %)
- height - высота элемента (px, %)

По умолчанию реальная ширина элемента = width + padding + border + margin

Свойство box-sizing позволяет менять то, как вычисляются размеры элемента. Значения:

- content-box (по умолчанию) - к высоте/ширине элемента прибавляются значения margin, padding и border
- padding-box - значение padding включается внутри width и height
- border-box (лучшее значение) - значения padding и border включены внутри width и height



# Фон элемента

Каждый блок html-элемента имеет фоновый слой, который может быть полностью прозрачным (по умолчанию) или заполнен цветом и/или одним или несколькими изображениями.

Свойства:

- background-color - устанавливает цвет фона элемента
- background-image - устанавливает фоновое изображение (одно или несколько) элемента.

Значения:

- none - есть слой изображения, но ничего не рисуется
- url(image.png) - указывается абсолютный или относительный путь к изображению
- linear-gradient(white, gray) - в качестве фонового изображения устанавливается градиент

# Фон элемента

- background-repeat - определяет, как фоновые изображения укладываются в области фона после того, как для них установлены размеры и позиционирование. Если значение свойства имеет два ключевых слова, первое используется для горизонтального направления, второе — для вертикального. Значения:

repeat-x	Изображение повторяется в горизонтальном направлении.	
repeat-y	Изображение повторяется в вертикальном направлении.	
repeat	Изображение повторяется в обоих направлениях так часто, чтобы покрыть область отрисовки фона. Если изображение не помещается, то обрезается.	
space	Изображение повторяется столько раз, сколько оно помещается в области фона, не обрезаясь, изображения расположены на равном расстоянии друг от друга.	
round	Изображение повторяется так часто, чтобы заполнить область фона, масштабируясь и не обрезаясь.	
no-repeat	Изображение размещается один раз и не повторяется.	

# Фон элемента

- background-attachment - указывает, является ли фоновое изображение фиксированным относительно области просмотра или прокручивается вместе с элементом или его содержимым. Значения:

scroll	Фоновое изображение прокручивается вместе с текстом и другим содержимым (значение по умолчанию)
fixed	Предотвращает перемещение, фиксирует фоновое изображение на заднем плане.
local	Фоновое изображение прокручивается вместе с содержимым элемента.

- background-position - указывает начальные значения фоновых изображений. Значения - top/left/right/center/bottom/ пиксели / %

# Фон элемента

- background-clip - определяет область рисования фона. Значения:

border-box	Фон распространяется до внешнего края границы
padding-box	Фон распространяется до внешнего края отступа. Под границей фон не рисуется.
content-box	Фон закрашивается внутри (обрезается) поля содержимого.
text	Фон закрашивается внутри (обрезается) текста переднего плана.

- background-size - устанавливает размер фоновых изображений. Значения:

длина	Размер задается парой значений, первое значение устанавливает ширину изображения, второе — высоту
%	Задаёт размер фонового изображения в процентах от ширины или высоты элемента, которое заполняется фоном.
cover	Масштабирует изображение с сохранением пропорций так, чтобы его ширина или высота равнялась ширине или высоте блока.
contain	Масштабирует изображение с сохранением пропорций таким образом, чтобы оно целиком поместилось внутри блока.

# Фон элемента

Фон блока элемента может иметь несколько слоев в CSS3. Количество слоев определяется количеством значений, разделенных запятыми, указанных в свойстве `background-image`. Значение `none` также создает слой.

Первое изображение в списке — это слой, отображаемый ближайший к пользователю, следующий отрисовывается за первым, и так далее. Цвет фона, если он есть, закрашивается под всеми остальными слоями.

# Фон элемента

Установка прозрачности:

- Использование функции `rgba()`, в которой последний параметр задает значение прозрачности, которое может быть в диапазоне от 0 (полная прозрачность) до 1 (полная непрозрачность).
- Использовании свойства `opacity`, значение которого устанавливается в диапазоне от 0 до 1, позволяя сделать любой элемент полупрозрачным.

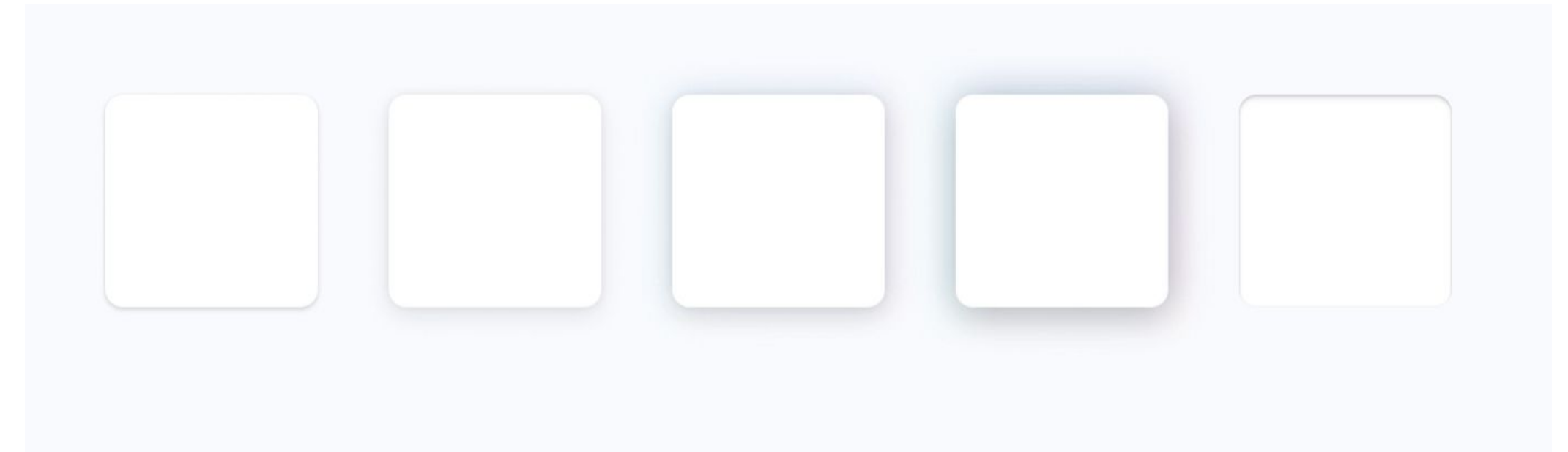
Когда использовать свойство `opacity` предпочтительнее?

- нужно сделать полупрозрачными несколько цветов. Свойство `opacity` позволяет сделать полупрозрачными цвет фона, текста и рамки элемента.
- нужно сделать что-то полупрозрачным, даже не зная его цвет, например, потому, что цвет может устанавливаться другой таблицей стилей
- нужно сделать полупрозрачным изображение
- нужно использовать переход, т.е. эффект анимации, который делает элемент постепенно появляющимся или исчезающим



# Тени элемента

Тени элементов в CSS3:



Блочные тени, свойство

`box-shadow`

(высокий уровень поддержки браузерами)

Текстовые тени, свойство

`text-shadow`

(Не работает в Internet Explorer)

# Тени элемента

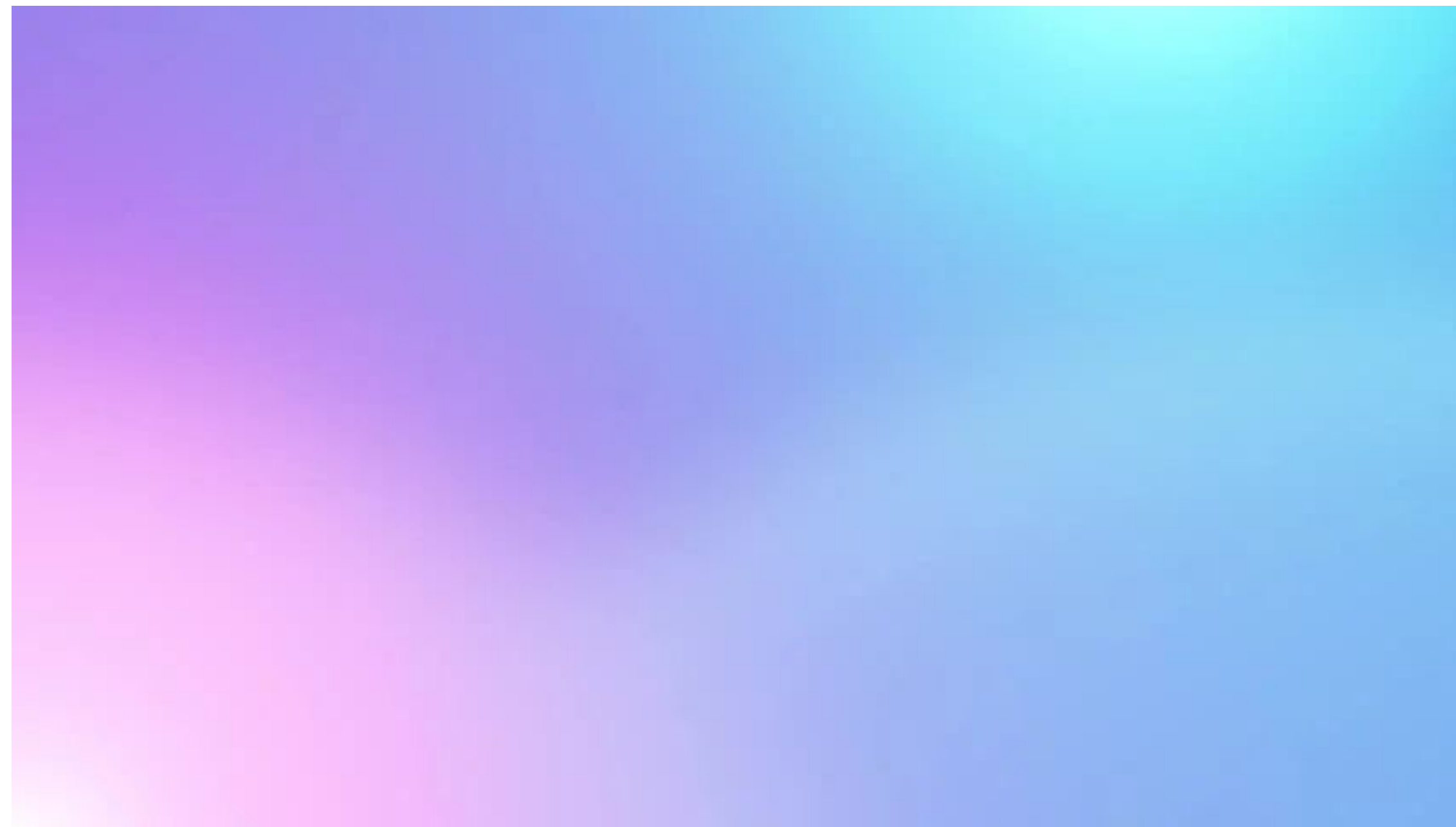
Свойство box-shadow добавляет элементу одну или более теней. Тень представляет собой копию элемента, смещенную на указанное расстояние.

box-shadow: x-offset y-offset blur spread color

Смещение по ширине      Смещение по высоте      Радиус размытия      Растяжение      Цвет тени

# Градиент

CSS-градиент представляет собой переходы от одного цвета к другому. Градиентный фон можно устанавливать в свойствах `background`, `background-image`, `border-image` и `list-style-image`.



# Градиент

Линейный градиент создается с помощью двух и более цветов, для которых задано направление, или линия градиента.

CSS-свойство `linear-gradient()`

Цвета градиента по умолчанию распределяются равномерно в направлении, перпендикулярном линии градиента.

```
background: linear-gradient(направление, цвет1, цвет2, ...)
```

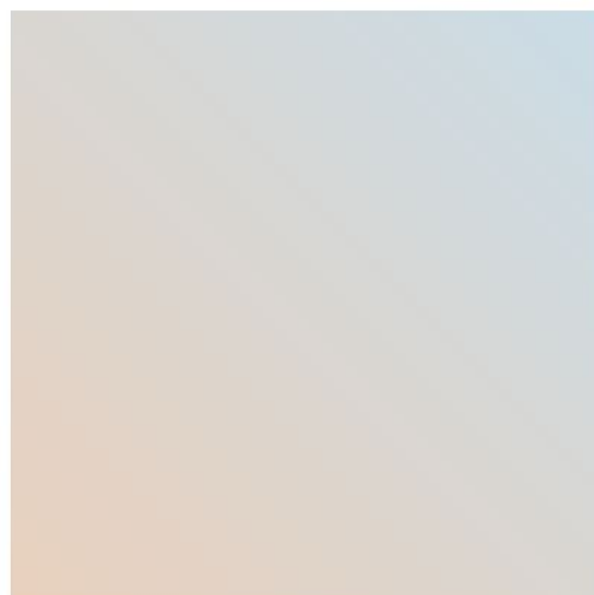
# Градиент

linear-gradient()

Направление:

- Угол наклона в градусах deg, значение которого определяет угол наклона линии внутри элемента.
- Ключевые слова 'to top', 'to right', 'to bottom', 'to left', которые соответствуют углу градиента, равному 0deg, 90deg, 180deg и 270deg соответственно.
- Направление можно задавать парой ключевых слов (например, to top left)

```
div {  
  height: 200px;  
  width: 200px;  
  background: linear-gradient(45deg, #EECFBA, #C5DDE8);  
}
```



```
div {  
  height: 200px;  
  width: 200px;  
  background: linear-gradient(to right, #F6EFD2, #CEAD78);  
}
```



# Градиент

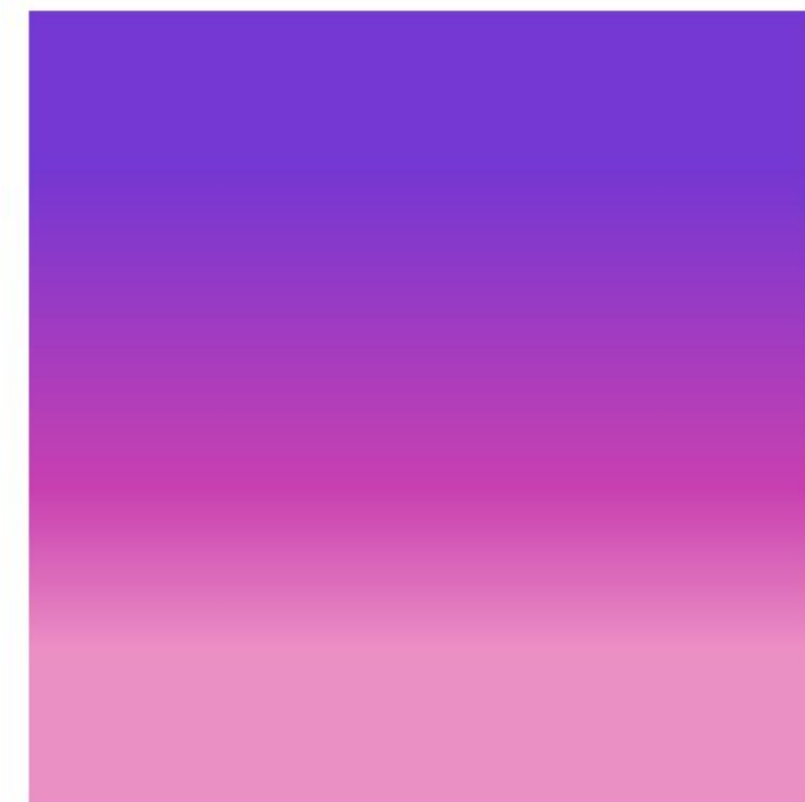
linear-gradient()

Цвет:

Для неравномерного распределения цветов указывается начальная позиция каждого цвета через точки остановки градиента.

Точки остановки задаются в %, где 0% — начальная точка, 100% — конечная точка

```
div {  
  height: 200px;  
  width: 200px;  
  background: linear-gradient(to top, #FA89C8 20%, #DA2BB5 40%, #802BDA 80%);  
}
```



# Градиент

Радиальный градиент - цвета выходят из одной точки (центра градиента) и равномерно распределяются наружу, рисуя форму круга или эллипса

CSS-свойство `radial-gradient()`

`background: radial-gradient(форма/размер/позиция центра, цвет1, цвет2, ...)`



# Градиент

radial-gradient()

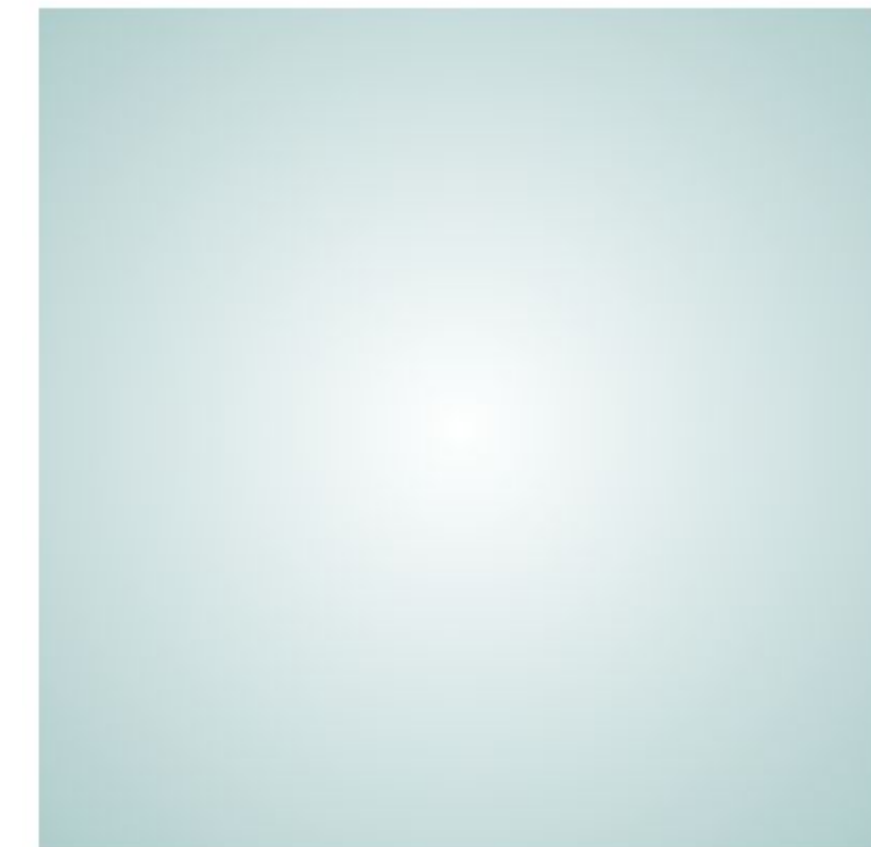
Форма градиента:

- circle - круг
- ellipse (по умолчанию) - эллипс

Позиция центра:

- at top/at center (по умолчанию)/at bottom/at left/at right

```
div {  
  height: 200px;  
  width: 200px;  
  background: radial-gradient(at center, #FFFFFF, #A7CECC);  
}
```



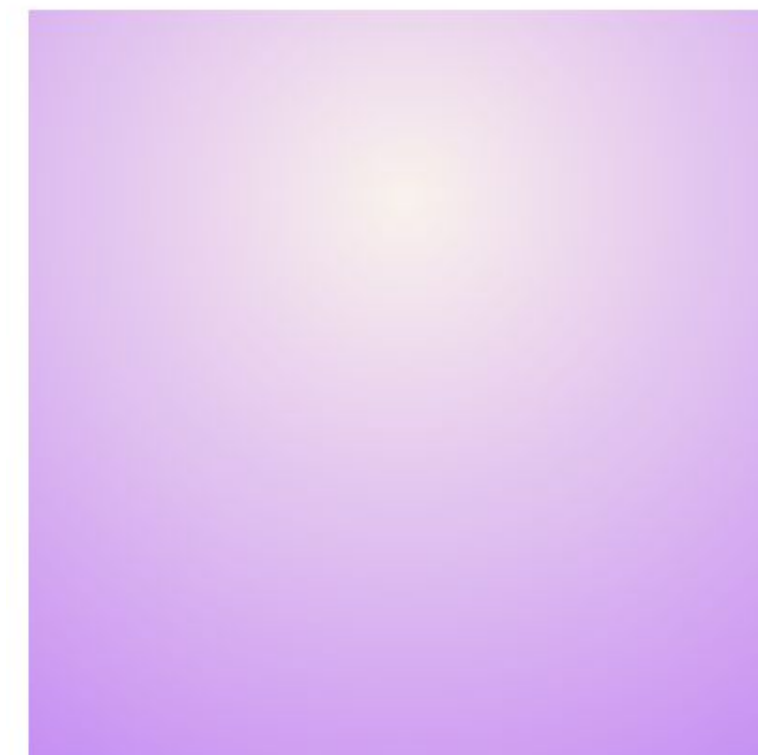
# Градиент

radial-gradient()

Размер градиента:

- closest-side - Размер градиента рассчитывается из расстояния до любой ближней стороны блока для circle или до ближних сторон по X и по Y для ellipse
- farthest-side - Размер рассчитывается из расстояния до дальних сторон
- closest-corner - Размер рассчитывается из расстояния до ближних углов
- farthest-corner - Размер рассчитывается из расстояния до дальних углов

```
div {  
  height: 200px;  
  width: 200px;  
  background: radial-gradient(circle farthest-corner at 100px 50px, #FBF2EB, #D189FA);  
}
```



# Градиент

Создание фигуры мяч

```
div {  
  height: 200px;  
  width: 200px;  
  border-radius: 50%;  
  margin: 0 auto;  
  background: radial-gradient(circle at 65% 15%, #5766D8, #9334C9);  
}
```



# Градиент

Повтор градиента:

Создается функций `repeating-linear-gradient()` и `repeating-radial-gradient()`

```
div {  
  height: 200px;  
  width: 200px;  
  background: repeating-radial-gradient(circle, #B9ECFE, #B9ECFE 10px, #82DDFF 10px, #82DDFF 20px);  
}
```





# Градиент

Комбинация градиента и фонового изображения:

```
div {  
  height: 500px;  
  width: 500px;  
  background: linear-gradient(45deg, ■ rgba(103, 0, 31, .8), ■ rgba(34, 101, 163, .5)), url(example.png);  
  background-size: cover;  
}
```



Без градиента:



# Позиционировани е

CSS-позиционирование описывает, как любой из элементов может быть размещен независимо от порядка документа, т.е. извлечен из общего «потока» элементов.

Каждый элемент дерева документа генерирует в соответствии с блочной моделью 0 или более блоков.

Расположение этих блоков регулируется:

- размерами и типом элемента,
- схемой позиционирования (нормальный поток, обтекание и абсолютное позиционирование),
- отношениями между элементами в дереве документа
- внешней информацией (например, размер области просмотра, внутренними размерами изображений и т.д.)

# Позиционировани е

Схемы позиционирования:

- Нормальный поток - блочный контекст форматирования (элементы с `display block`, `list-item` или `table`), строчный (встроенный) контекст форматирования (элементы с `display inline`, `inline-block` или `inline-table`), и относительное и «липкое» позиционирование элементов уровня блока и строки.
- Обтекание - блок удаляется из нормального потока и позиционируется влево или вправо. Содержимое обтекает правую сторону элемента с `float: left` и левую сторону элемента с `float: right`
- Абсолютное позиционирование - блок полностью удаляется из нормального потока и ему присваивается позиция относительно содержащего блока. Абсолютное позиционирование реализуется с помощью значений `position: absolute;` и `position: fixed`



# Позиционировани е

Позиционирование позволяет переопределять поведение базового потока документа.

Позиционирование задается свойством `position`.

Виды позиционирования:

- `static` - статическое (по умолчанию).
- `relative` - относительное. Можно модифицировать окончательное положение позиционируемого объекта (задавать смещение с помощью свойств: `top`, `bottom`, `left`, `right`)
- `absolute` - абсолютное. Элемент не существует в нормальном потоке макета документа.

Вместо этого он располагается на своём собственном слое отдельно от всего остального и ведет себя так, будто других элементов вокруг не существует.

# Позиционировани е

Позиционирование позволяет переопределять поведение базового потока документа.

Позиционирование задается свойством `position`.

Виды позиционирования:

- `sticky` - липкое. Позволяет позиционируемому элементу вести себя как будто он относительно позиционирован, до тех пор пока он не будет прокручен до определённой пороговой точки, после чего становится фиксированным.
- `fixed` - фиксированное. Элемент всегда располагается относительно окна браузера, родительские элементы при этом игнорируются, элемент не смещается при прокручивании страницы.

# Позиционировани е

Контекст позиционирования - относительно какого элемента позиционируется позиционируемый элемент.

Например, чтобы абсолютно позиционировать какой-то элемент не относительно окна браузера, а относительно какого-нибудь элемента, то этому элементу добавляют `position: relative`

# Позиционировани е

Обтекание - позволяет блокам смещаться влево или вправо на текущей строке.

Свойство float. Значения:

- left - элемент перемещается влево, содержимое обтекает плавающий блок по правому краю.
- right - элемент перемещается вправо, содержимое обтекает плавающий блок по левому краю.

# Позиционировани е

Плавающий блок принимает размеры своего содержимого с учетом внутренних отступов и рамок.

Верхние и нижние отступы `margin` плавающих элементов не схлопываются.

Плавающие элементы могут использовать отрицательные отступы `margin`, чтобы перемещаться за пределы области содержимого их родительского элемента.

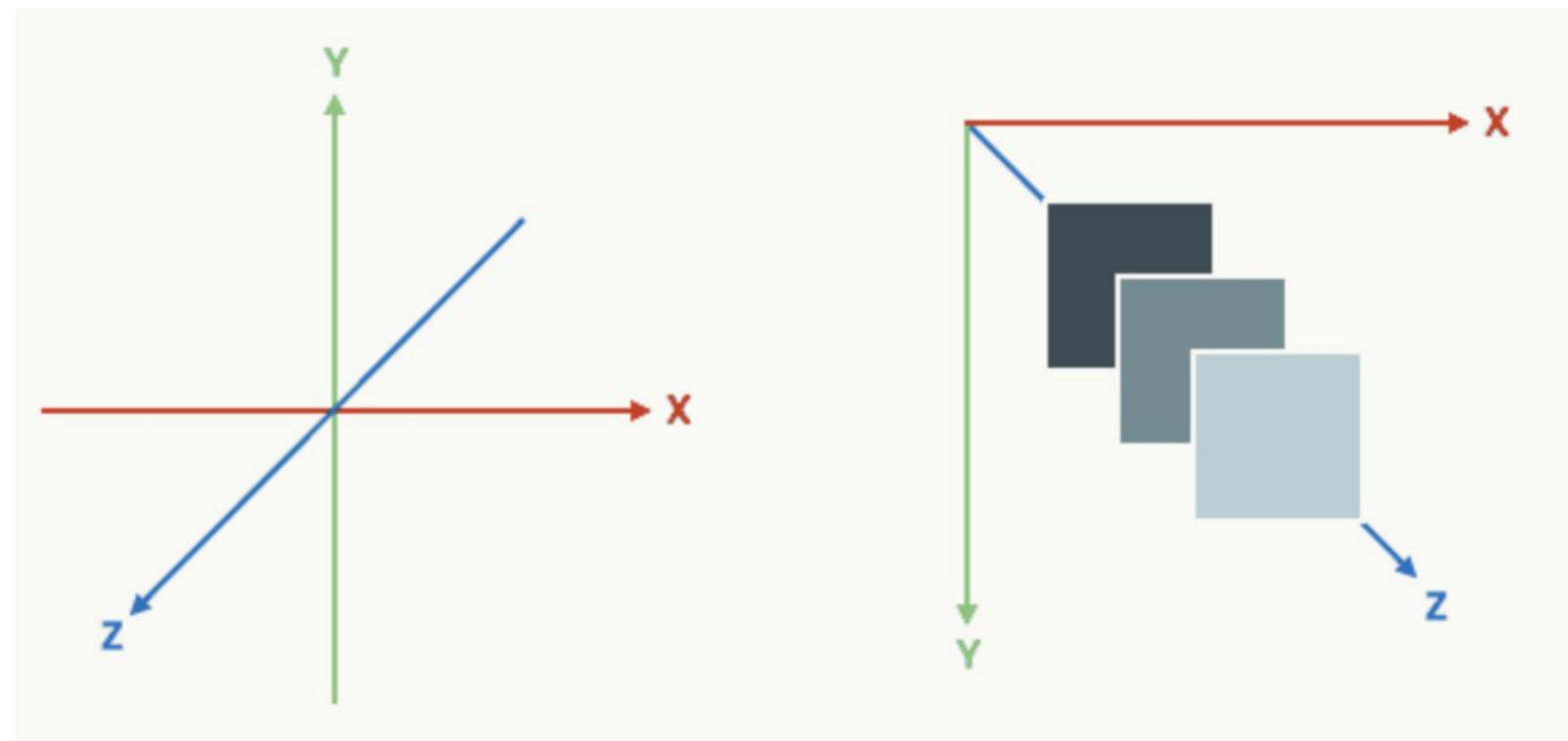
Свойство не оказывает влияние на элементы с `display: flex` и `display: inline-flex`. Не применяется к абсолютно позиционированным элементам.

# Позиционирование

Свойство z-index позволяет изменить порядок наложения элементов

Каждый блок имеет позицию в трех измерениях. В дополнение к горизонтальному и вертикальному положению, блоки выкладываются вдоль оси Z друг над другом.

Порядок, в котором дерево документа отрисовывается на экране, описывается с помощью контекста наложения.



# Позиционировани е

## Контекст наложения

Если для элементов свойства `z-index` и `position` не заданы явно, контекст наложения равен порядку их расположения в исходном коде и браузер отображает элементы на странице в следующем порядке:

- Корневой элемент , который содержит все элементы веб-странице.
- Блочные элементы, неплавающие и непозиционированные
- Плавающие непозиционированные элементы в порядке их расположения в исходном коде
- Строковые непозиционированные элементы (текст, изображения).
- Позиционированные элементы в порядке их следования в исходном коде



# Позиционировани е

Свойство `z-index` создает новый контекст наложения. Оно позволяет изменить порядок наложения позиционированных элементов. Элементы будут отображаться на странице в следующем порядке:

- Корневой элемент , который содержит все элементы веб-странице
- Позиционированные элементы с отрицательным значением `z-index`
- Блочные элементы, неплавающие и непозиционированные
- Плавающие непозиционированные элементы в порядке их расположения в исходном коде
- Строковые непозиционированные элементы (текст, изображения)
- Позиционированные элементы со значениями `z-index: 0` и `z-index: auto`
- чем больше `z-index` элемента, тем выше над остальными элементами находится