

# «Fullstack» разработка

## Лекция 4

### Использование блочной верстки

# Кроссбраузерная верстка

**Хак** — это набор приемов, когда определённому браузеру «подсовывают» код, который понимается только этим браузером, а остальными игнорируется.

Основной недостаток хаков - снижается универсальность кода.

# Слой

В HTML4, HTML5 и XHTML слой — это элемент веб-страницы, созданный с помощью тега `<div>`, к которому применяется стилевое оформление.

# Использование тега <div>

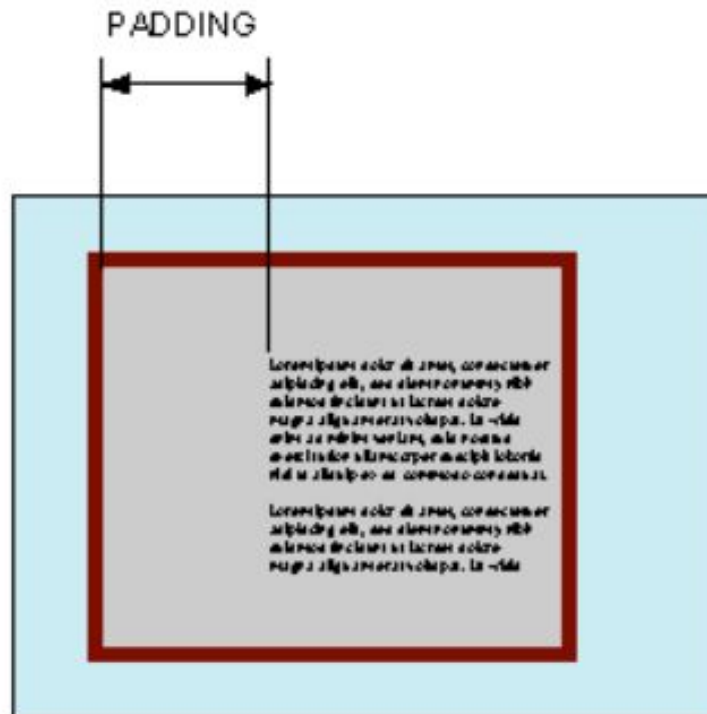
Тег <div> – это основа блочной вёрстки. Данный элемент выполняет роль фундамента, общий вид которого можно задавать широким набором свойств.

# Свойства, формирующие блочный элемент



# Поля (свойство padding)

Полем называется расстояние от внутреннего края рамки элемента до воображаемого прямоугольника, ограничивающего его содержимое.

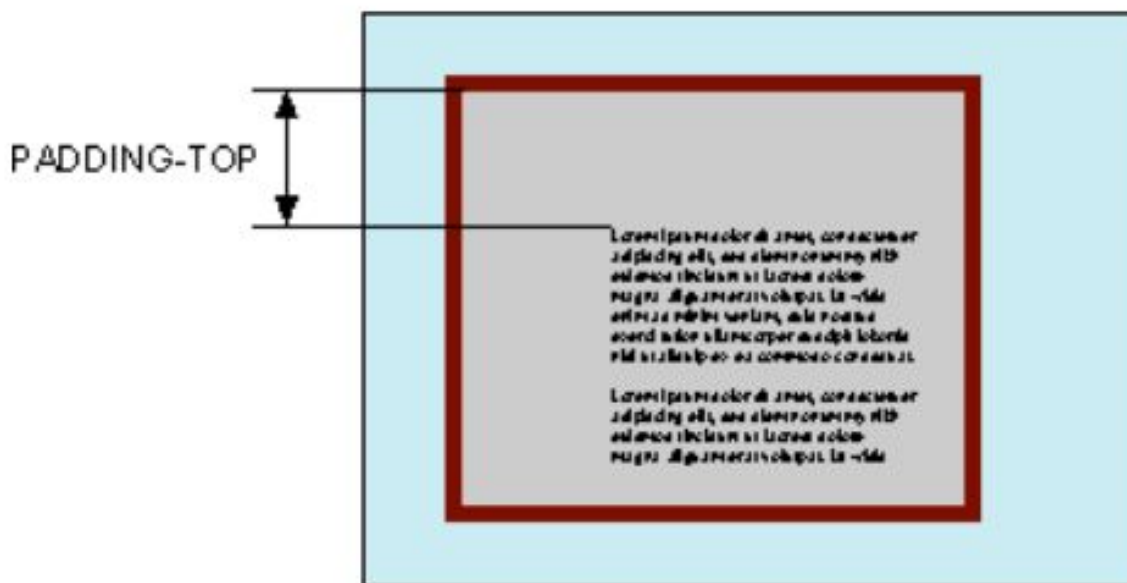


# Поле сверху (padding-top)

Устанавливает значение поля от верхнего края содержимого элемента.

Синтаксис:

`padding-top:` значение | inherit

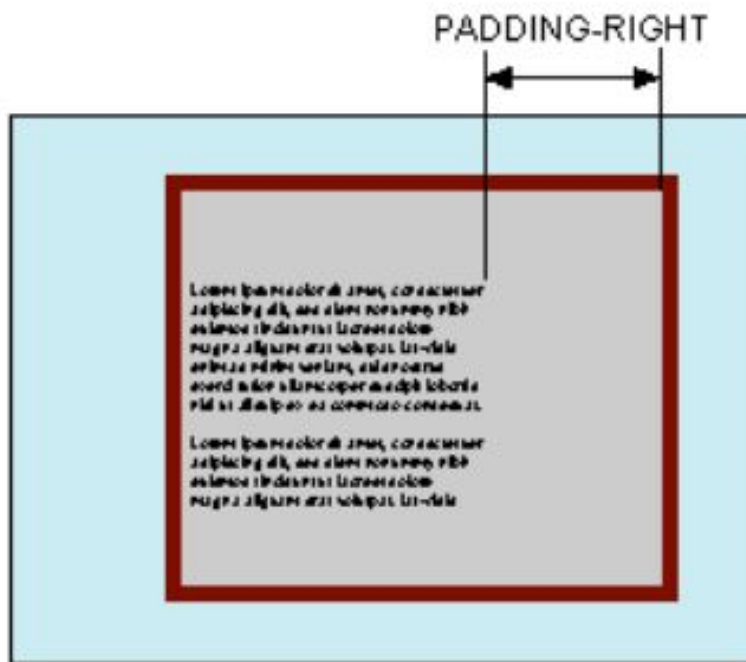


# Поле справа (padding-right)

Устанавливает значение поля от правого края содержимого элемента.

Синтаксис:

`padding-right:` значение | inherit



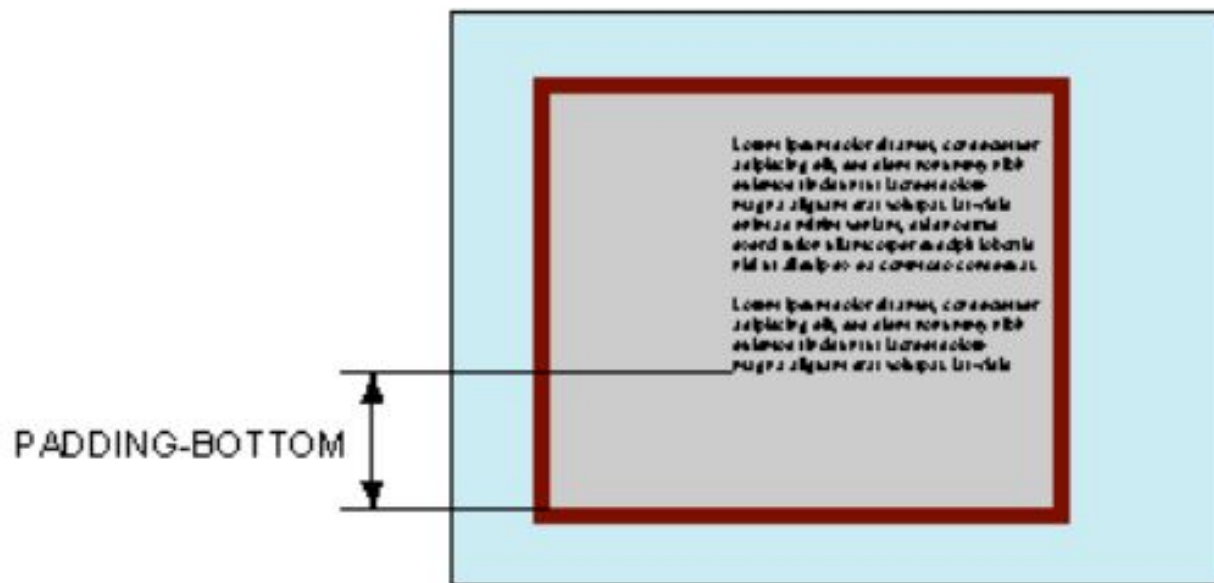


# Поле снизу (padding-bottom)

Устанавливает значение поля от нижнего края содержимого элемента.

Синтаксис:

`padding-bottom:` значение | inherit

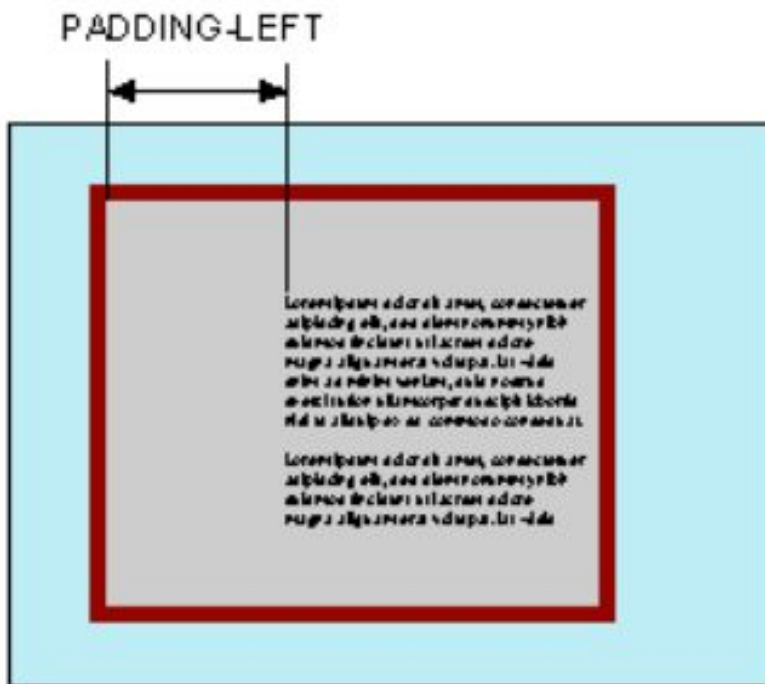


# Поле слева (padding-left)

Устанавливает значение поля от левого края содержимого элемента.

Синтаксис:

`padding-left:` значение | inherit



# Поля (свойство padding)

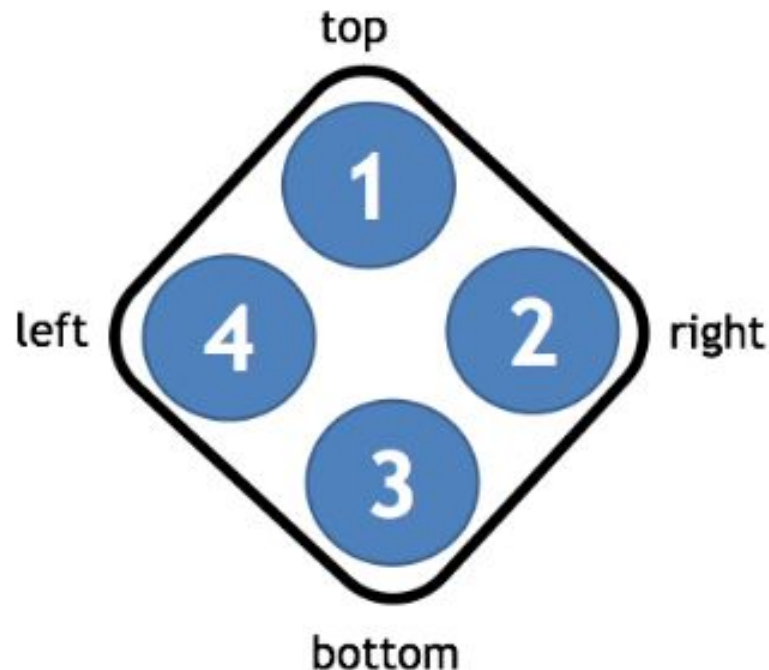
## Синтаксис:

`padding:` [значение | проценты] {1, 4} | inherit

Число значений	Результат
1	Поля будут установлены одновременно от каждого края элемента.
2	Первое значение устанавливает поля от верхнего и нижнего края, второе — от левого и правого.
3	Первое значение задает поле от верхнего края, второе — одновременно от левого и правого края, а третье — от нижнего края.
4	Поочередно устанавливается поля от верхнего, правого, нижнего и левого края.

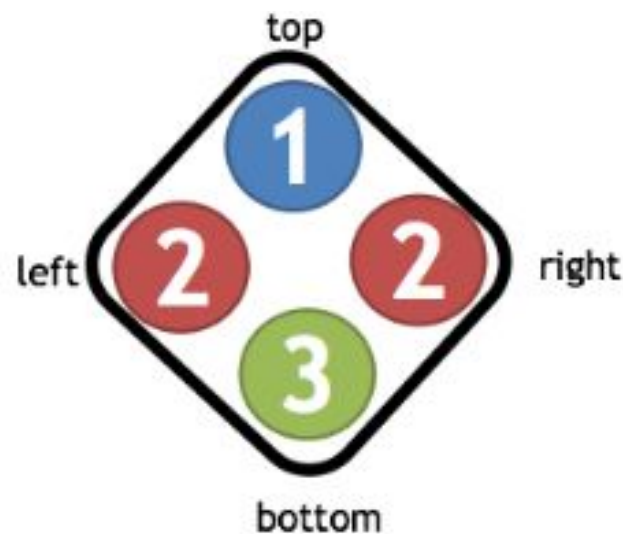
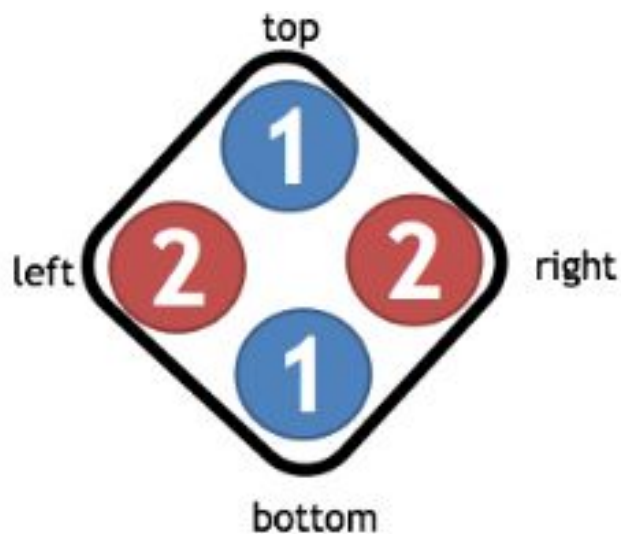
# Порядок задания значений в свойствах padding, border, margin

Если установлено одно или четыре значения:



# Порядок задания значений в свойствах padding, border, margin

Если установлено два значения:      Если установлено три значения:



# Границы (border)

Границы — это линии вокруг полей элемента.

У каждой линии есть толщина, стиль и цвет.

Стиль начертания линий может быть индивидуален для каждой из сторон блока.

# Толщина границы (border-width)

Синтаксис:

```
border-width: [значение | thin | medium | thick] {1,4} | inherit
```

Три переменные — thin (2 пиксела), medium (4 пиксела) и thick (6 пикселей) задают толщину границы. Для более точного значения, толщину можно указывать в пикселах или других единицах.

Разрешается использовать одно, два, три или четыре значения, разделяя их между собой пробелом.

# Стиль границы

Синтаксис:

```
border-style: [none | hidden | dotted | dashed | solid | double |  
groove | ridge | inset | outset] {1,4} | inherit
```

Вид рамки в зависимости от стиля границы элемента:





# Цвет границы (border-color)

Синтаксис:

```
border-color: [цвет | transparent] {1,4} | inherit
```

transparent — устанавливает прозрачный цвет.

# Универсальное свойство границ

Синтаксис:

```
border : [width, style, color] {1, 3} | inherit
```

Одновременно устанавливает толщину, стиль и цвет границы вокруг элемента. Значения могут идти в любом порядке, разделяясь пробелом, браузер сам определит, какое из них соответствует нужному свойству. Для установки границы только на определенных сторонах элемента, возможно воспользоваться свойствами **border-top**, **border-bottom**, **border-left**, **border-right**.

# Все свойства границ

	<b>border-width</b>	<b>border-style</b>	<b>border-color</b>
<b>border-top</b>	border-top-width	border-top-style	border-top-color
<b>border-right</b>	border-right-width	border-right-style	border-right-color
<b>border-bottom</b>	border-bottom-width	border-bottom-style	border-bottom-color
<b>border-left</b>	border-left-width	border-left-style	border-left-color

# Скругленные уголки (свойство border-radius)

Устанавливает радиус скругления уголков рамки. Если рамка не задана, то скругление также происходит и с фоном.

Синтаксис

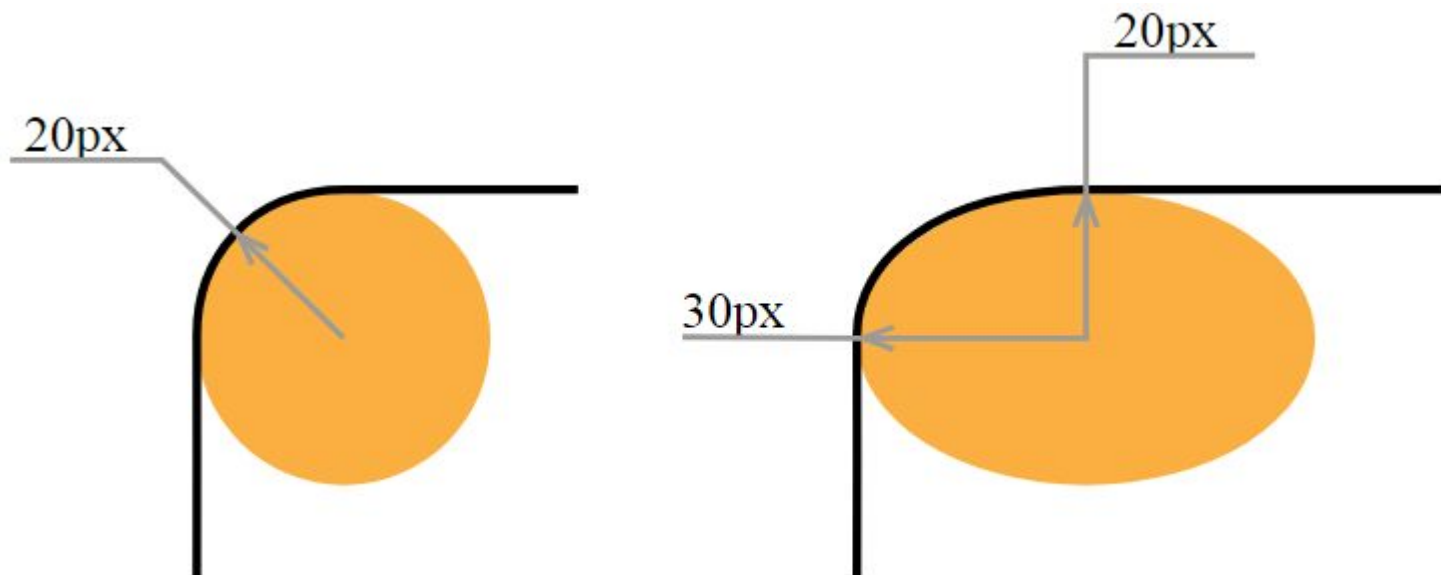
```
border-radius: <радиус>{1,4} [ / <радиус>{1,4} ]
```

# Значения border-radius

Число	Результат
1	Радиус указывается для всех четырех уголков.
2	Первое значение задает радиус верхнего левого и нижнего правого уголка, второе значение — верхнего правого и нижнего левого уголка.
3	Первое значение задает радиус для верхнего левого уголка, второе — одновременно для верхнего правого и нижнего левого, а третье — для нижнего правого уголка.
4	По очереди устанавливает радиус для верхнего левого, верхнего правого, нижнего правого и нижнего левого уголка.

# Эллиптические уголки

В случае задания двух параметров через слэш, то первый задает радиус по горизонтали, а второй по вертикали (эллиптические уголки). На рисунке показана разница между обычным скругленным уголком и эллиптическим уголком.



# Другие свойства скруглённых уголков

`border-bottom-left-radius`

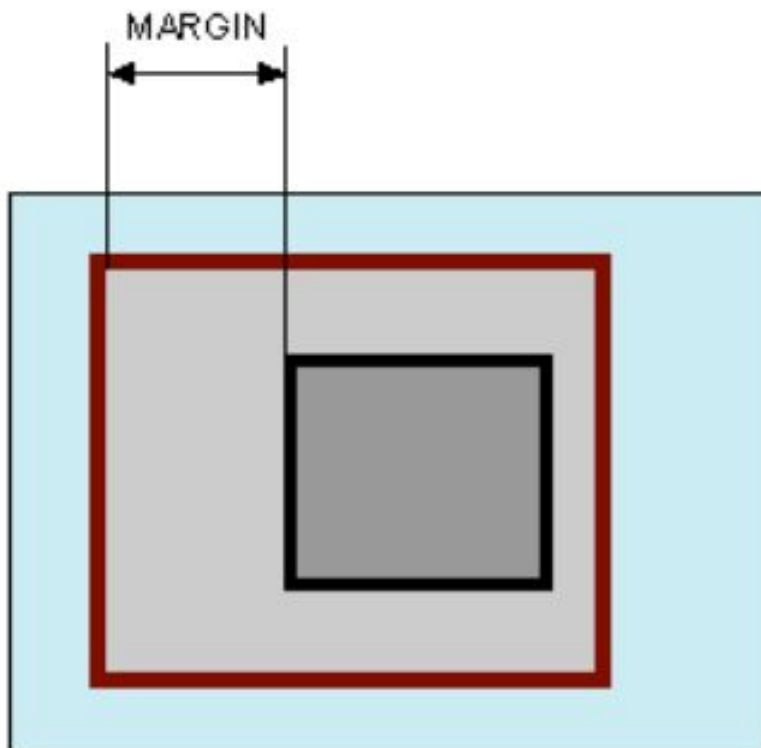
`border-bottom-right-radius`

`border-top-left-radius`

`border-top-right-radius`

# Отступы (margin)

Отступ — пустое пространство от внешнего края границы, полей или содержимого блока.





# Отступы (margin)

Особенности:

- Отступы прозрачны, на них не распространяется цвет фона или фоновая картинка, заданная для блока;
- Отступы в отличие от полей могут принимать отрицательное значение, это приводит к сдвигу всего блока в указанную сторону;
- Для отступов характерно явление под названием «схлопывание», когда отступы у близлежащих элементов не суммируются, а объединяются меж собой;
- Отступы, заданные в процентах, вычисляются от ширины контента блока. Это касается как вертикальных, так и горизонтальных отступов.

# Отступы (свойство margin)

Синтаксис:

```
margin: [значение | проценты] {1, 4} | inherit
```

Порядок установки значений такой же, как и для padding и border. Возможно установить от 1 до 4 значений.

# Отступы на отдельных краях

Как и для свойств `bottom` и `padding` для отступов существуют специфические свойства, характерные только для определённых положений сверху, справа, снизу или

слева: `margin-top`, `margin-right`, `margin-bottom`, `margin-left`.

# Ширина

Устанавливает ширину контента в блоке.

Синтаксис:

```
width: значение | проценты | auto | inherit
```

auto — устанавливает ширину исходя из типа и содержимого элемента.

# Ширина блока

Ширина блока это комплексная величина и складывается из нескольких значений свойств:

- `width` — ширина контента, т.е. содержимого блока;
- `padding-left` и `padding-right` — поле слева и справа от контента;
- `border-left` и `border-right` — толщина границы слева и справа;
- `margin-left` и `margin-right` — отступ слева и справа.

# Ширина блока



# Ширина блока

Пример:

```
width: 300px; /* Ширина слоя */  
margin: 7px; /* Значение отступов */  
border: 4px solid black; /* Параметры границы */  
padding: 10px; /* Поля вокруг текста */
```

Ширина =  $300 + 7 + 7 + 4 + 4 + 10 + 10 = 342\text{px}$

# Высота блока

Устанавливает высоту контента в блоке.

Синтаксис:

`height:` значение | проценты | auto | inherit

auto — устанавливает высоту исходя из типа и содержимого элемента.



# Высота блока



# Свойство overflow

Управляет отображением содержания блочного элемента, если оно целиком не помещается и выходит за область заданных размеров.

Синтаксис:

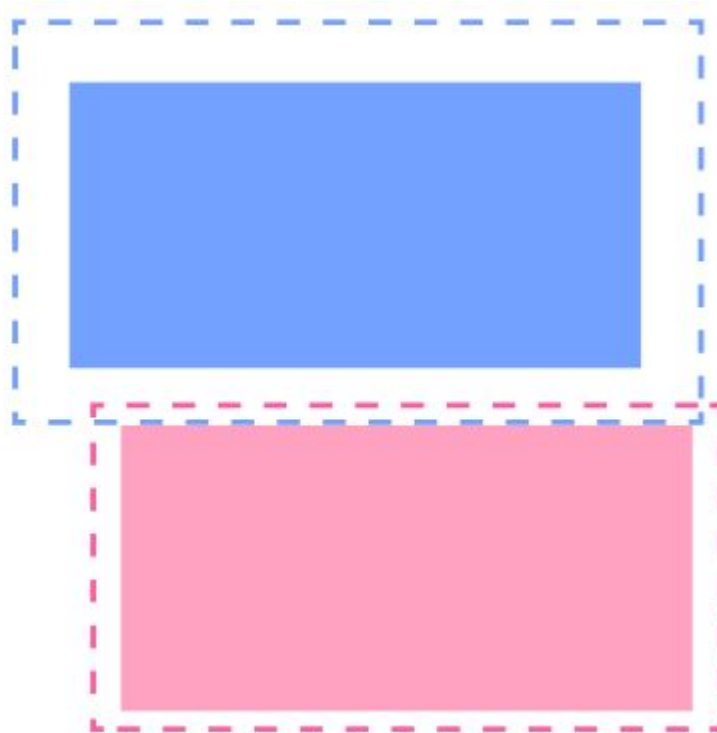
```
overflow: auto | hidden | scroll | visible | inherit
```

Свойства:

- visible — отображается все содержание элемента, даже за пределами установленной высоты и ширины;
- hidden — отображается только область внутри элемента, остальное будет скрыто;
- scroll — всегда добавляются полосы прокрутки;
- auto — полосы прокрутки добавляются только при необходимости.

# Схлопывание отступов

Для положительных значений отступов выбирается наибольшее значение из двух отступов, и оно устанавливается как расстояние между блоками.



# Отмены схлопывания

Схлопывание не срабатывает:

- для элементов, у которых установлено свойство padding;
- для элементов, у которых на стороне схлопывания задана граница;
- на элементах с абсолютным позиционированием, т. е. таких, у которых position установлено как absolute;
- на плавающих элементах (для них свойство float задано как left или right);
- для строчных элементов;
- для <html>.

# Поток документа

Порядок вывода объектов на странице и называется «ПОТОКОМ».

# Позиционирование элементов

Позиционированием называется положение элемента в системе координат. Различают четыре типа позиционирования: **нормальное, абсолютное, фиксированное и относительное.**

# Использование position

Свойство position задает позиционирование элемента относительно исходного положения или родителя.

Синтаксис:

```
position: absolute | fixed | relative | static | inherit
```

# Нормальное позиционирование

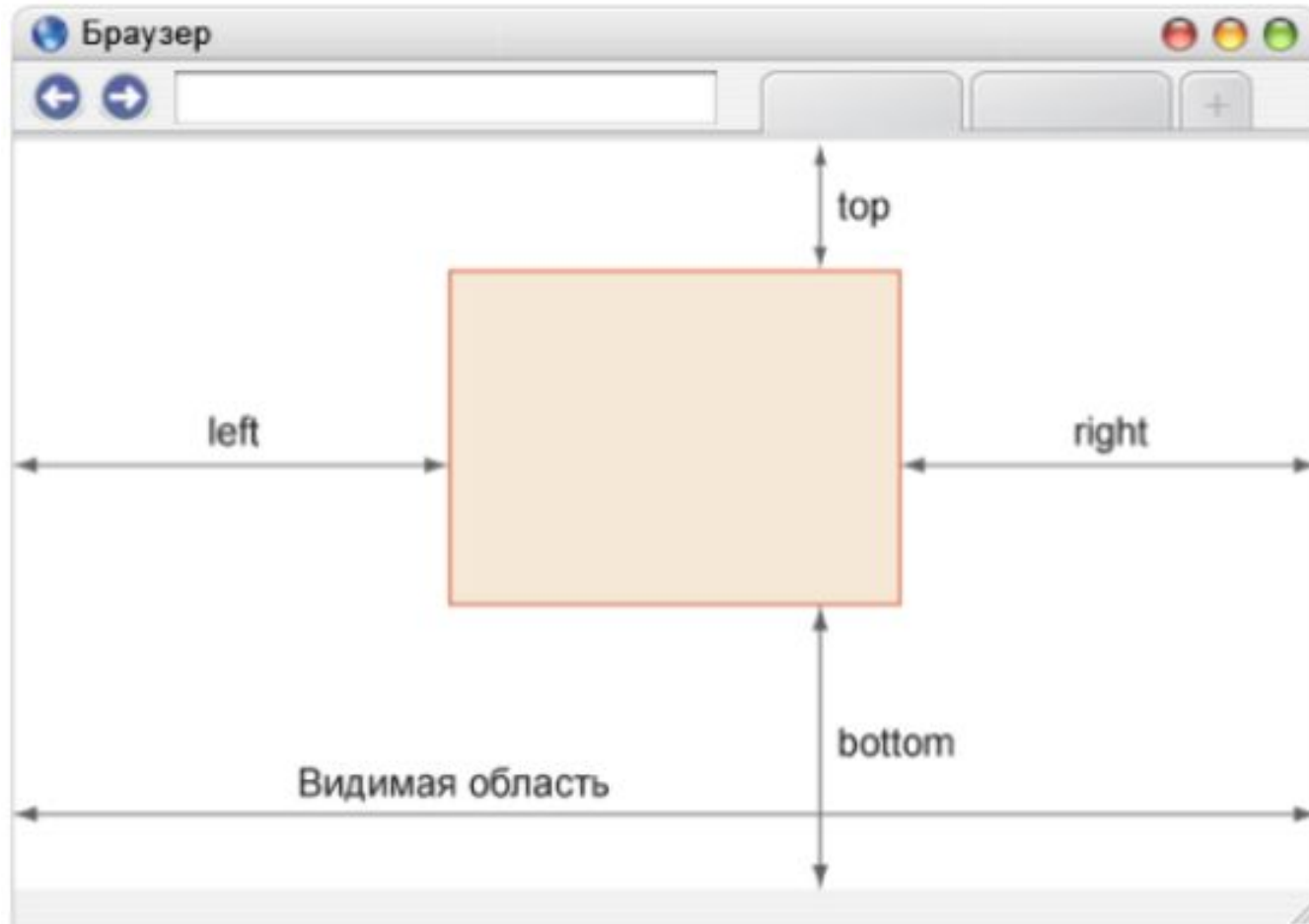
Если для элемента свойство `position` не задано или его значение `static`, элемент выводится в потоке документа как обычно. Иными словами, элементы отображаются на странице в том порядке, как они идут в исходном коде HTML.



# **Абсолютное позиционирование**

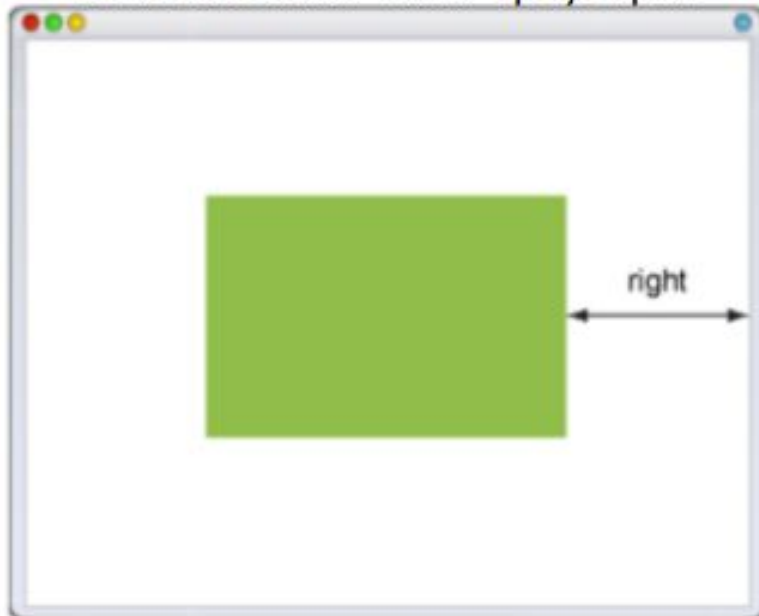
При абсолютном позиционировании элемент не существует в потоке документа и его положение задаётся относительно краёв браузера.

# Абсолютное позиционирование

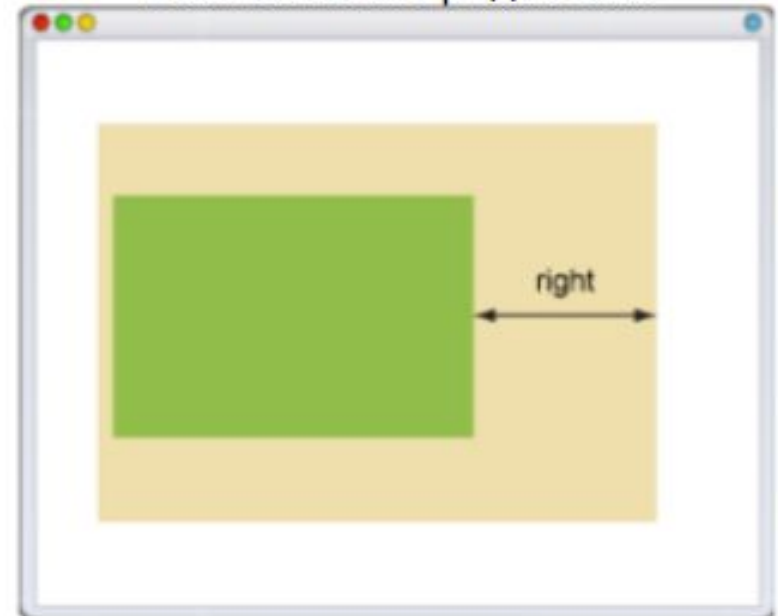


# Свойства top, right, bottom, left

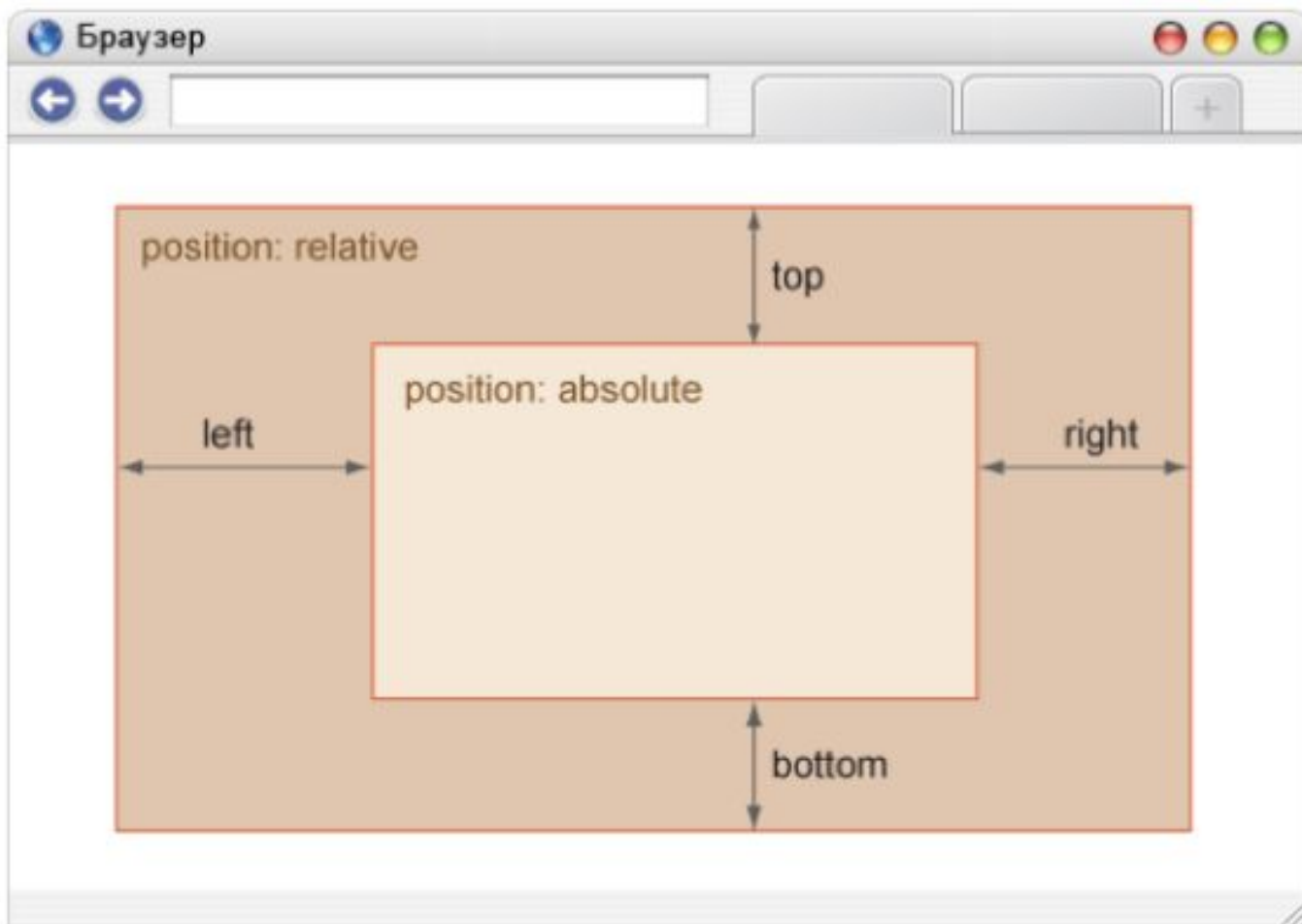
Значение свойства right  
относительно окна браузера:



Значение свойства right  
относительно родителя:



# Относительное позиционирование и вложенные спои



# Фиксированное позиционирование

Фиксированное положение слоя задаётся значением `fixed` свойства `position` и по своему действию похоже на абсолютное позиционирование. Но в отличие от него привязывается к указанной свойствами `left`, `top`, `right` и `bottom` точке на экране и не меняет своего положения при прокрутке веб-страницы.

Нет полос прокрутки при нахождении за пределами видимой области.

# Плавающие элементы

Плавающими (обтекаемыми) называются такие элементы, которые обтекаются по контуру другими объектами веб-страницы, например, текстом.

# Использование плавающих элементов

Использование:

- галереи небольших фотографий;
- двух и трехколоночные макеты;
- горизонтальные меню;
- иллюстрации в тексте;
- многоколоночный текст.

# Свойство float

Синтаксис:

```
float: left | right | none | inherit
```

Свойство float превращает элемент в плавающий, при этом он прижимается к левому или правому краю родителя, а текст его обходит с других сторон.



# Обтекания нет



ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

# Для картинки установлено float: left



ipsum dolor sit amet, consectetur  
elit, sed do eiusmod tempor incididunt ut  
dolore magna aliqua. Ut enim ad minim  
s nostrud exercitation ullamco laboris nisi ut  
a commodo consequat. Duis aute irure dolor in  
reprehenderit in voluptate velit esse cillum dolore eu fugiat  
nulla pariatur. Excepteur sint occaecat cupidatat non proident,  
sunt in culpa qui officia deserunt mollit anim id est laborum.

# Для картинки установлено float: right

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.



# Влияние обтекания

Плавающие элементы не оказывают влияние на высоту блока, в котором они находятся.

# Отмена обтекания

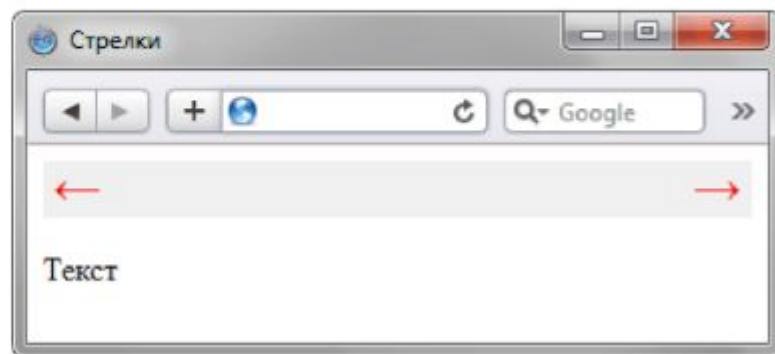
## Ширина элемента

Если плавающий элемент будет занимать всю доступную ширину, то остальные элементы, следующие за ним, будут начинаться с новой строки.

# Отмена обтекания

## Ширина элемента

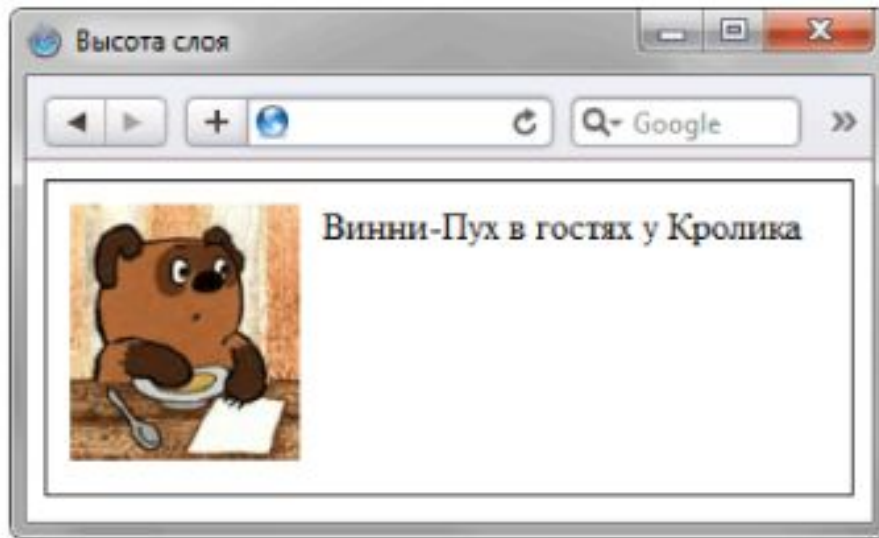
```
.left { float: left; width: 50%; }  
.right { float: right; width: 50%; text-align: right; }
```



# Отмена обтекания

Использование **overflow**

Использование **overflow** со значением **auto**, **scroll** или **hidden** отменяет действие **float**.



# Отмена обтекания

## Свойство clear

Значения:

- left — отменяет обтекание с левого края элемента (float: left);
- right — отменяет обтекание с правой стороны элемента (float: right);
- both — отменяет обтекание элемента одновременно с правого и левого края. Это значение рекомендуется устанавливать, когда требуется отменить обтекание элемента, но неизвестно точно с какой стороны.



# Блочные элементы

Блочным называется элемент, который отображается на веб-странице в виде прямоугольника. Такой элемент занимает всю доступную ширину, высота элемента определяется его содержимым, и он всегда начинается с новой строки. К блочным элементам относятся

теги `<address>`, `<blockquote>`, `<div>`, `<fieldset>`, `<form>`, `<h1>`, ... , `<h6>`, `<hr>`, `<ol>`, `<p>`, `<pre>`, `<table>`, `<ul>` и некоторые устаревшие.

# Блочные элементы

Свойство для определения блочного элемента: **display:**  
**block.**

# Особенности блочных элементов

- Блоки располагаются по вертикали друг под другом;
- На прилегающих сторонах элементов действует эффект схлопывания отступов;
- Запрещено вставлять блочный элемент внутрь строчного;
- По ширине блочные элементы занимают всё допустимое пространство;
- Если задана ширина или высота контента (свойства `width` и `height`), то ширина блока складывается из значений `width` (`height`), полей, границ, отступов слева и справа (сверху и снизу);
- Высота блочного элемента вычисляется браузером автоматически, исходя из содержимого блока.

# Анонимный блок

Если в одном блочном элементе содержатся строчные элементы наряду с блочными, то вокруг строчных элементов генерируется анонимный блок. Для таких блоков применяется стиль по умолчанию, а также наследуемый стиль, заданный для его родителя

Первый абзац

Анонимный блок

Второй абзац

# Строчные элементы

Строчными называются такие элементы документа, которые являются непосредственной частью строки. К строчным элементам относятся теги `<img>`, `<span>`, `<a>`, `<q>`, `<code>` и др.

Используются для изменения вида текста или его логического выделения.

# Строчные элементы

Свойство для определения строчного элемента:

`display: inline.`

# Свойства строчных элементов

- Внутри строчных элементов допустимо помещать текст или другие строчные элементы. Вставлять блочные элементы внутрь строчных запрещено;
- Эффект схлопывания отступов не действует;
- Свойства, связанные с размерами (width, height) не применимы;
- Ширина равна содержимому плюс значения отступов, полей и границ;
- Несколько строчных элементов идущих подряд располагаются на одной строке и переносятся на другую строку при необходимости;
- Можно выравнивать по вертикали с помощью свойства vertical-align.

# Анонимный строчный элемент

Серым цветом на рисунке помечен текст внутри тега `<span>` , а пунктиром анонимные строчные элементы. Для таких элементов задать стиль напрямую нельзя, он наследуется от родителя.

Выполните восемь повторений



# Блочно-строчные элементы

Блочно-строчный элемент – элемент, по структуре похожий на блочный, но обтекаемый остальными объектами, как строчный.

# Блочно-строчные элементы

- Внутри строчно-блочных элементов допустимо помещать текст, строчные или блочные элементы;
- Высота элемента вычисляется браузером автоматически, исходя из содержимого блока;
- Ширина равна содержимому плюс значения отступов, полей и границ;
- Несколько элементов идущих подряд располагаются на одной строке и переносятся на другую строку при необходимости;
- Можно выравнивать по вертикали с помощью свойства `vertical-align`;
- Разрешено задавать ширину и высоту;
- Эффект схлопывания отступов не действует.

# Блочно-строчные элементы

Свойство для определения блочно-строчного элемента: **display: inline-block**.

# Свойство display

Многоцелевое свойство, которое определяет, как элемент должен быть показан в документе.

Синтаксис:

```
display: block | inline | inline-block | inline-table | list-item |  
none | run-in | table | table-caption | table-cell | table-column-  
group | table-column | table-footer-group | table-header-group |  
table-row | table-row-group
```

# СВОЙСТВО display

Многоцелевое свойство, которое определяет, как элемент должен быть показан в документе.

Синтаксис:

```
display: block | inline | inline-block | list-item | none
```

Свойства:

- block — элемент показывается как блочный;
- inline — элемент показывается как строчный;
- Inline-block — это значение генерирует блочный элемент, который обтекается другими элементами веб-страницы подобно встроенному элементу;
- list-item — элемент выводится как блочный и добавляется маркер списка;
- none — временно удаляет элемент из документа. Занимаемое им место не резервируется и веб-страница формируется так, словно элемента и не было.

# СВОЙСТВО `vertical-align`

Выравнивает элемент по вертикали относительно своего родителя, окружающего текста или ячейки таблицы.

`vertical-align`: `baseline` | `bottom` | `middle` | `sub` | `super` |  
`text-bottom` | `text-top` | `top` | `inherit` | значение | проценты

# СВОЙСТВО `vertical-align`

- `baseline` — выравнивает базовую линию текущего элемента по базовой линии родителя. Если родительский элемент не имеет базовой линии, то за нее принимается нижняя граница элемента;
- `bottom` — выравнивает основание текущего элемента по нижней части элемента строки, расположенного ниже всех;
- `middle` — выравнивание средней точки элемента по базовой линии родителя плюс половина высоты родительского элемента;
- `sub` — элемент изображается как подстрочный, в виде нижнего индекса. Размер шрифта при этом не меняется;
- `super` — элемент изображается как надстрочный, в виде верхнего индекса. Размер шрифта остается прежним;
- `top` — выравнивание верхнего края элемента по верху самого высокого элемента строки.

# СВОЙСТВО min-width

Синтаксис:

`min-width:` значение | проценты | inherit

Устанавливает минимальную ширину элемента. Если окно браузера по ширине становится меньше заданной минимальной ширины элемента, то ширина элемента остается неизменной, а в окне появляется горизонтальная полоса прокрутки.



# Свойство max-width

Синтаксис:

`max-width:` значение | проценты | `inherit`

Устанавливает максимальную ширину элемента. Значение ширины элемента будет вычисляться в зависимости от значений установленных свойств `width`, `max-width` и `min-width`.

# Вычисление ширины

$\text{min-width} < \text{width} < \text{max-width} \Rightarrow \text{width}$

$\text{width} < \text{max-width} \Rightarrow \text{width}$

$\text{min-width} > \text{width} > \text{max-width} \Rightarrow \text{min-width}$

$\text{min-width} > \text{width} < \text{max-width} \Rightarrow \text{min-width}$

# СВОЙСТВО min-height

Синтаксис:

`min-height: значение | проценты | inherit`

Задаёт минимальную высоту элемента. Значение высоты элемента будет вычисляться в зависимости от установленных значений свойств `height`, `max-height` и `min-height`.

# Свойство max-height

Синтаксис:

`max-height: значение | проценты | inherit`

устанавливает максимальную высоту элемента.

Значение высоты элемента будет вычисляться в зависимости от значений установленных свойств height, max-height и min-height.

# Вычисление высоты

$\text{min-height} < \text{height} < \text{max-height} \Rightarrow \text{height}$

$\text{min-height} > \text{height} > \text{max-height} \Rightarrow \text{min-height}$

$\text{min-height} > \text{height} < \text{max-height} \Rightarrow \text{min-height}$

$\text{min-height} < \text{height} \Rightarrow \text{height}$

$\text{min-height} > \text{height} \Rightarrow \text{min-height}$

$\text{min-height} < \text{max-height} \Rightarrow \text{max-height}$