

# Верстка. Основы построения лэяута.



div & span



это два **основных** элемента на которых держится вся  
блочная верстка

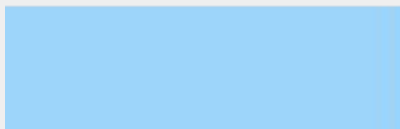
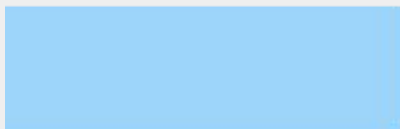
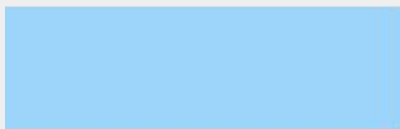


# Блочные и строчные элементы

## Блочные элементы

div, p, ul, li, h1, h2, ...

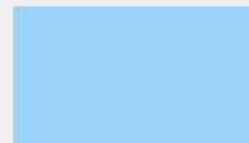
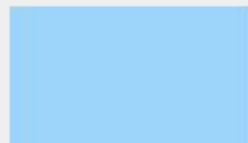
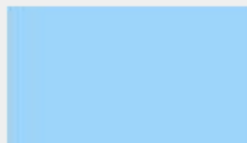
выстраиваются в столбец



## Строчные элементы

span, a, b, i ...

выстраиваются в строку



# display - определяет поведение

**block** - блочное поведение

**inline** - строчное поведение

**inline-block** - смешанное поведение

**none** - элемент скрыт

```
.element {  
  display: block;  
}
```

CSS

примечание: существует множество значений свойства **display**. Указанные - основные, частоиспользуемые.

# разница в поведении

## БЛОЧНЫЕ:

- выстраиваются в столбец
- можно задать размеры (высоту, ширину)
- имеют внутренние и внешние отступы

## СТРОЧНЫЕ:

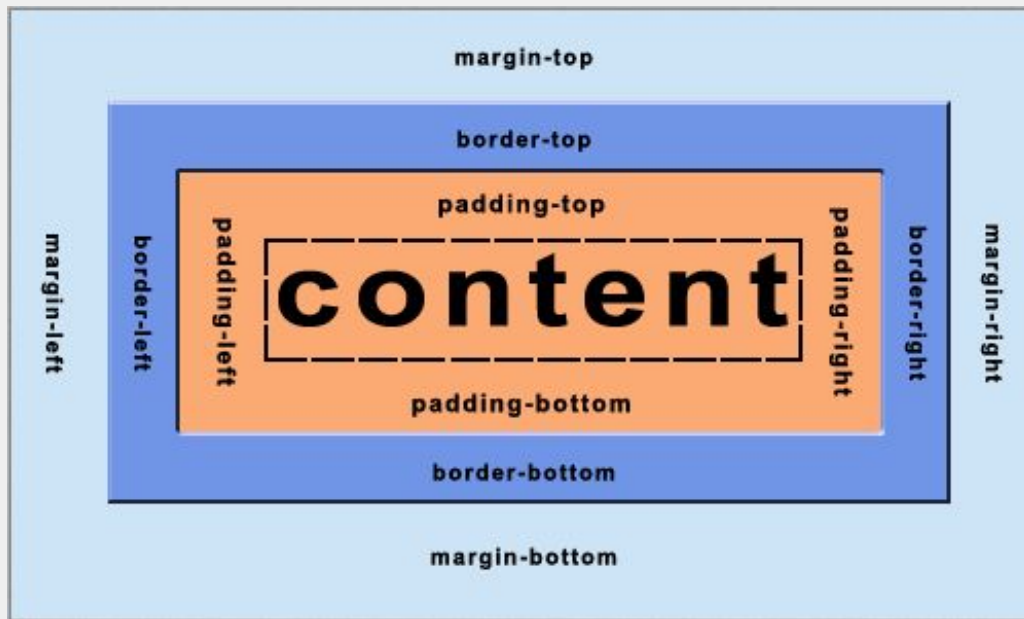
- выстраиваются в строку
- могут иметь внешние отступы только справа или слева.
- воспринимаются браузером как элемент строки (“слово”).
- имеют между собой обязательные отступы в зависимости от размера шрифта



# “Блочная модель элемента”



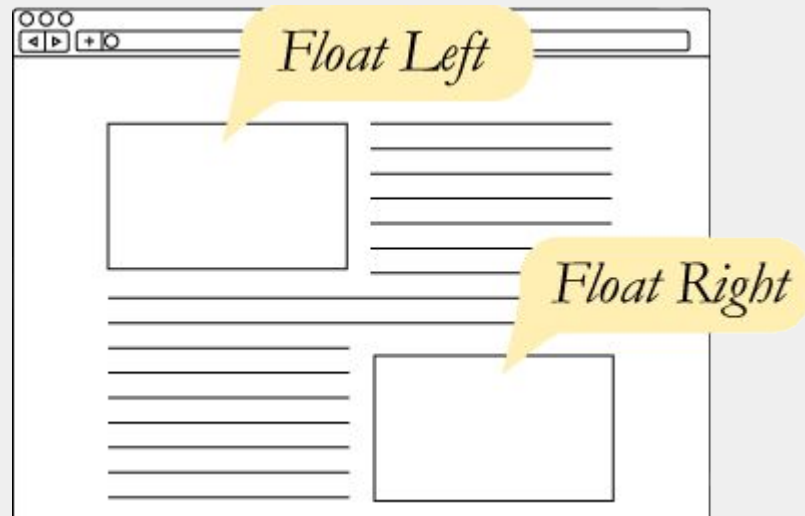
# в реальной жизни



# “обтекание”

“Обтекание” - задается свойством **float**.

**float** - определяет по какой стороне (правой или левой) будет выравниваться элемент, при том что другие элементы и контент будут его обтекать с противоположной стороны.





# Основная проблема флоатов.



# шаг 1. Создаем блоки с флоатом

```
.float-1 {  
  float: left;  
  width: 200px;  
}
```

```
.float-2 {  
  float: left;  
  width: 200px;  
}
```



шаг 2. Создаем блок в который будут  
вложены блоки с флоатом

Я - блок в котором будут лежать только блоки с флоатом.



шаг 3. Кладем под него еще один произвольный блок

Я - блок в котором будут лежать только блоки с флоатом.

Я блок снизу



шаг 4. Добавляем блоки с флоатом в верхний блок.

Я - блок в котором будут лежать только блоки с флоатом.

```
.float-1 {  
  float: left;  
  width: 200px;  
}  
.float-2 {  
  float: left;  
  width: 200px;  
}
```

Я блок снизу

**ву - аля, сэр, извольте получить!**



структура этого безобразия:

```
<div class="wrap">  
  <div class="float-1"></div>  
  <div class="float-2"></div>  
</div>  
<div class="wrap-bottom">  
  Я блок снизу  
</div>
```



# лекарство – свойство `clear`

```
<div class="wrap">
  <div class="float-1"></div>
  <div class="float-2"></div>

  <div class="div-clear"></div>
</div>
<div class="wrap-bottom">
  Я блок снизу
</div>
```

html

```
.float-1, .float-2 {
  float: left;
  width: 200px;
}
.div-clear {
  clear: both;
}
```

css

Я - блок в котором будут лежать только блоки с флоатом.



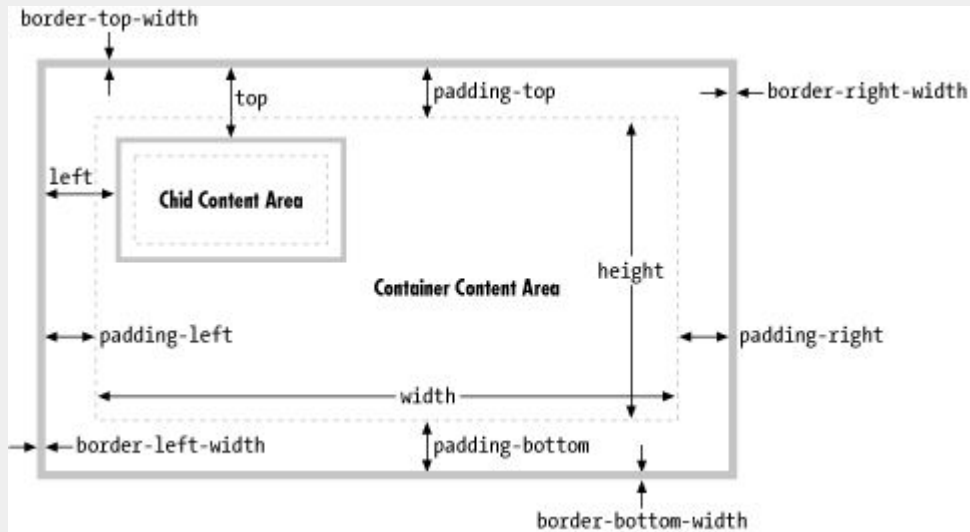
Я блок снизу

браузер



# другие способы позиционирования

Правила позиционирования задаются свойством **position**. Если значение этого свойства отлично от стандартного, то на элемент начинают действовать **свойства позиционирования**. Такие как **top**, **right**, **bottom**, **left**;





# другие способы позиционирования

## Значение свойства **position**:

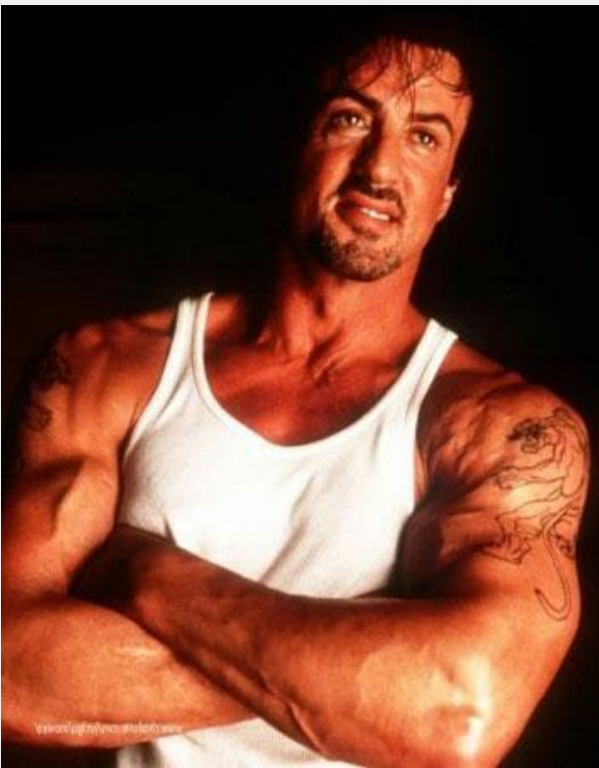
**static** - стандартное (дефолтное) значение для всех элементов. При нем свойства позиционирования на элемент не действуют.

**relative** - задает элемент-родитель относительно которого будут отпозиционированы его дети (чайлды).

**absolute** - значение задающее позицию относительно элемента родителя (со значение отличным от **static**)

**fixed** - задает поведение позиционирования относительно окна браузера





„Проверь себя  
в реальных условиях“

*- Сильвестр Сталлоне*



**LoftSchool**  
ОТ МЫСЛИТЕЛЯ К СОЗДАТЕЛЮ

*beginners course.*