

**FE-Morning-7 – 22.09.2017**

# План урока

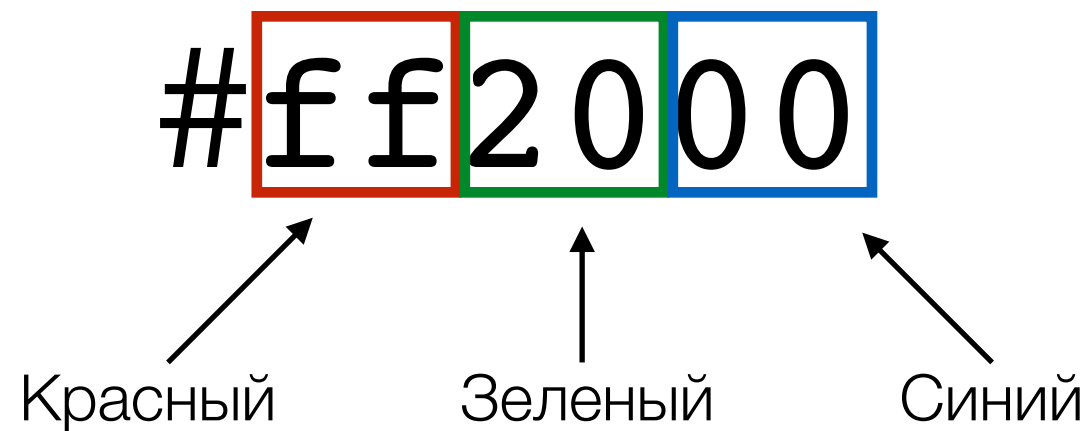
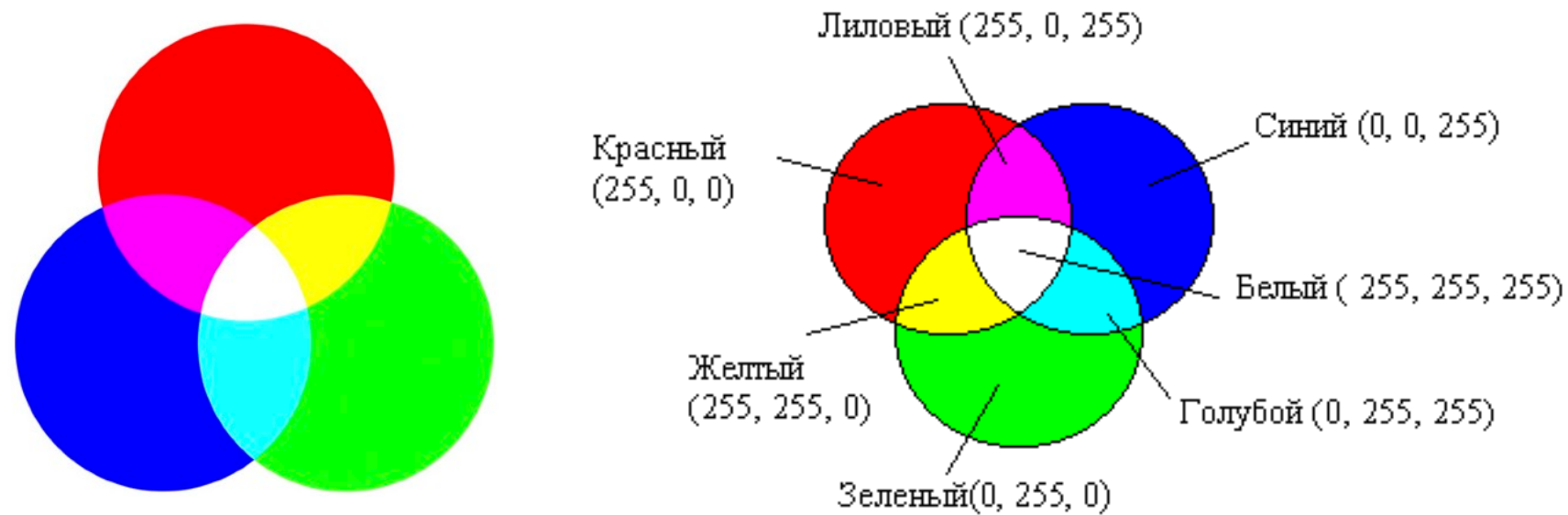
1. Каким бывает цвет в CSS? Форматы цвета. (color, background-color)
2. Прозрачность в CSS. (opacity)
3. Прозрачность цвета. (rgba)
4. Что такое потоки?
5. Обтекаемые элементы в потоке
6. Свойство float;
7. Решаем задачи на float
8. Создание колонок с применением float и inline-block
9. Примеры типичных ошибок и их решения

## Каким может быть цвет в CSS?

1. Определенный именем (например *red*), список цветов по ссылке [https://www.w3schools.com/colors/colors\\_names.asp](https://www.w3schools.com/colors/colors_names.asp)
2. Определенный в HEX-формате (напр. *#ff3345*)
3. Определенный с помощью rgb/rgba. (напр. *rgb(255, 132, 10)*)
4. Определенный с помощью hsl/hsla (напр. *hsl(120, 100%, 50%)*)

# Откуда берется значение цвета?

в HEX и rgb формате, это значения красного, зеленого и синего цвета.



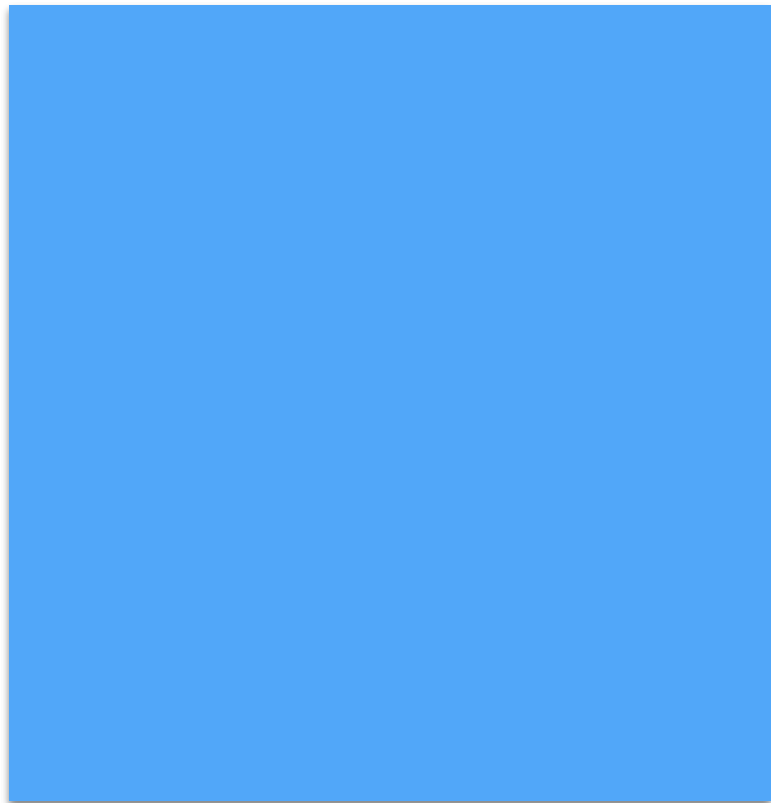
В шестнадцатеричной системе исчисления

# В каких свойствах применяются значения цвета?

- **color** – цвет текста
- **background-color** – цвет фона
- **border-color** – цвет рамки
- **box-shadow** – цвет тени
- **outline** – цвет обводки
- **text-shadow** – цвет тени текста
- **и другие.**

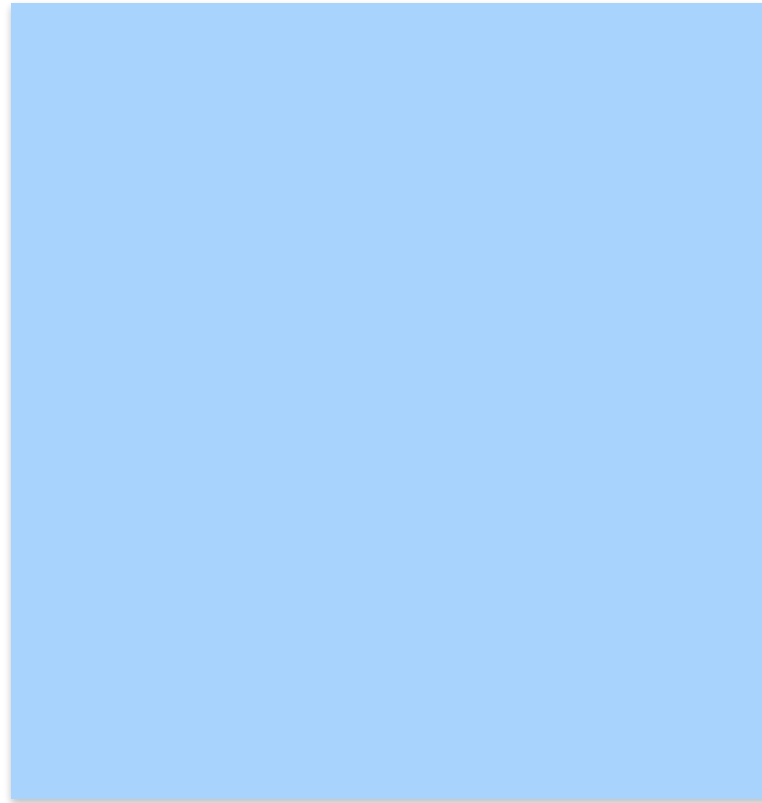
# Прозрачность в CSS

Прозрачность задается свойством **opacity**



100%

```
selector{  
  opacity: 1;  
}
```



50%

```
selector{  
  opacity: 0.5;  
}
```

0%

```
selector{  
  opacity: 0;  
}
```

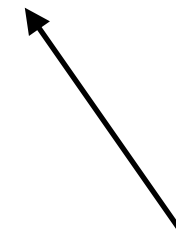
# Прозрачность цвета (rgba)

Можно изменить в последнем значении функций rgba/hsla

```
selector{  
  background-color:rgba(0, 0, 0, 0.5);  
}
```



Коэффициент прозрачности



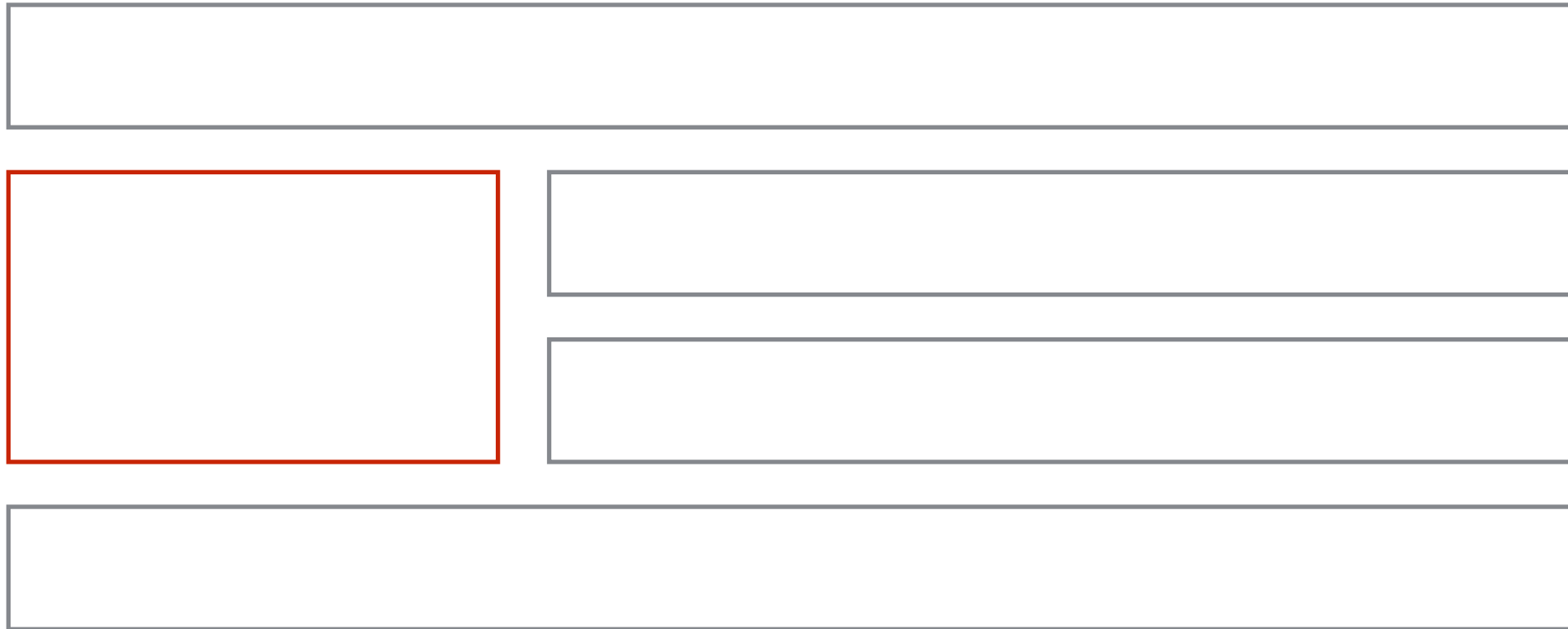
```
selector{  
  background-color:hsla(137, 20%, 100%, 0.5);  
}
```

# Поговорим о потоках?

1. Что такое поток?
2. Как ведут себя элементы в потоке?
3. Как вытащить элемент из потока?
4. Что произойдет с потоком если мы уберем красный элемент?



# Обтекаемый поток



Выделенный красным элемент находится в потоке,  
но при этом остальные элементы его «обтекают»

# Свойство float

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



Left



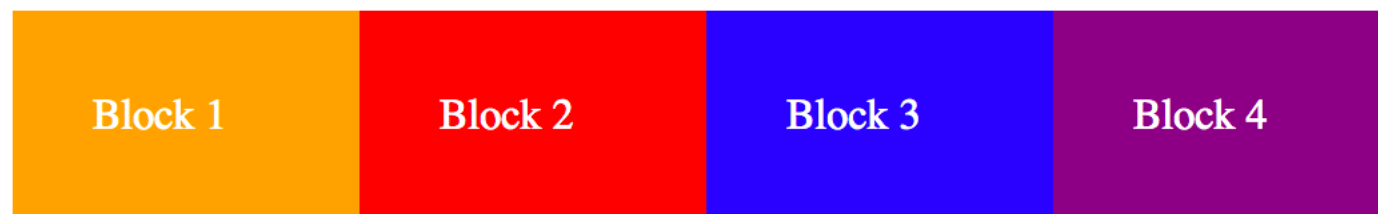
Right

## Задача с float'ами

1. Сделать заглавную букву текста большой и обтекаемой

**T**his is some text. This is some text. This is some text. This is some text. This is some text. This is  
some text. This is some text. This is some text. This is some text. This is some text. This is some  
text. This is some text. This is some text. This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text. This is some text. This is some text.

- ## 2. Сделаем горизонтальные блоки.



# Сделать следующую структуру

"Never increase, beyond what is necessary, the number of entities required to explain anything." William of Ockham (1285-1349)

**20%**

## Free Web Building Tutorials

At W3Schools you will find all the Web-building tutorials you need, from basic HTML and XHTML to advanced XML, XSL, Multimedia and WAP.

W3Schools - The Largest Web Developers Site On The Net!

**80%**

# Как правильно верстать «КОЛОНКИ»

float: left;

float: right;

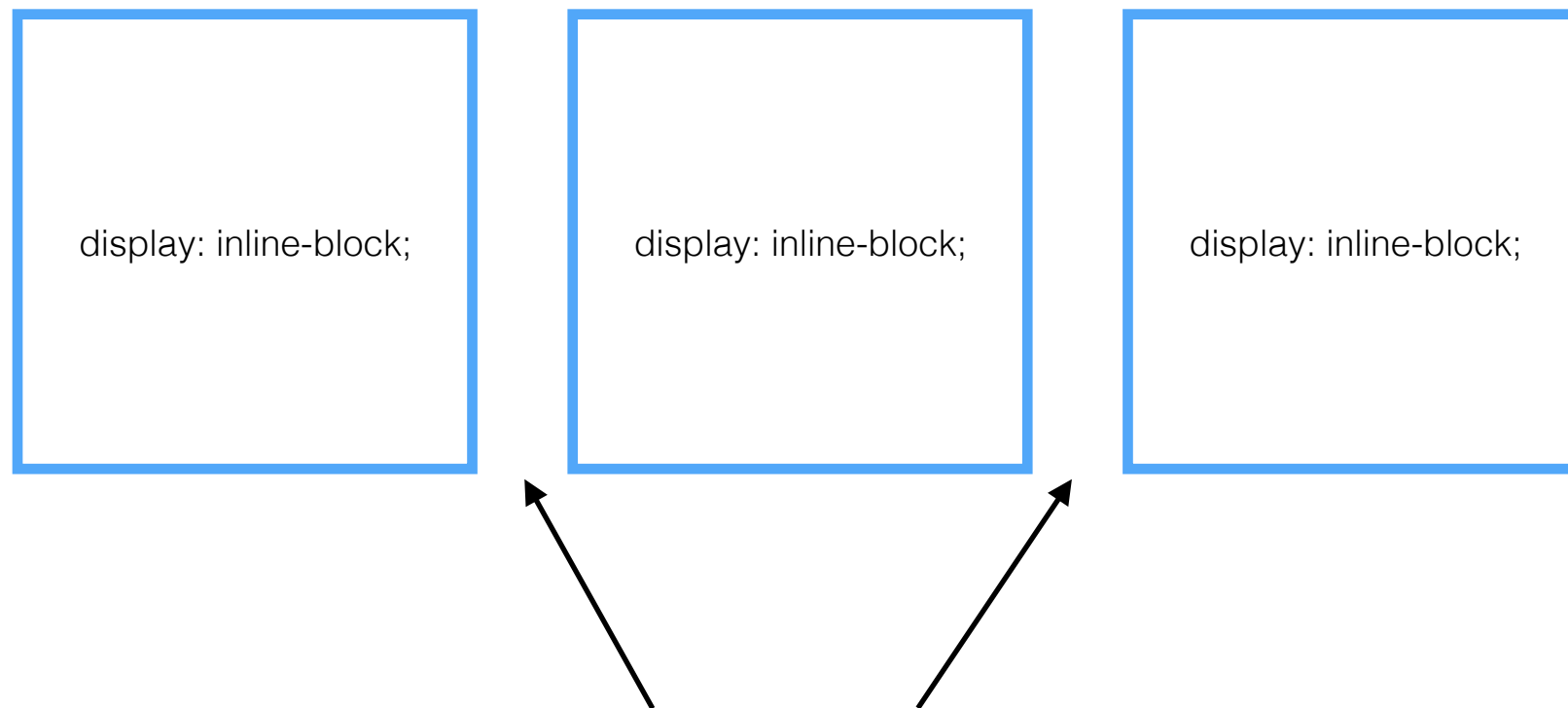
**FreebiesGallery**

[Go to Website](#)

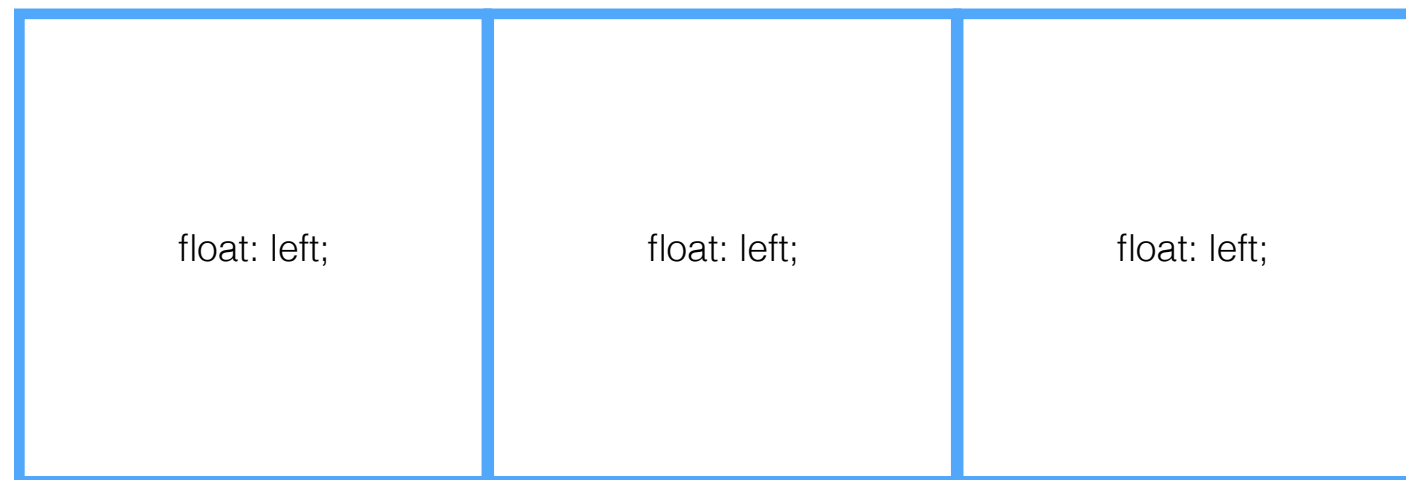
PROFESSIONAL WEBSITES  
IN JUST \$299



# Как правильно верстать «КОЛОНКИ»



Отступы между блоками – это пробелы



## **Какие проблемы возникают при верстке float'ами?**

1. float – съедают высоту.
2. Все элементы после элемента с float – будут его обтекать
3. При float: right – элементы меняют порядок на обратный

## Свойство **clear**;

`clear: none | left | right | both | initial | inherit;`

Отменяет обтекание элементов.

left – отменяет обтекание слева

right – отменяет обтекание справа

both – отменяет обтекание с двух сторон



# Девять правил float-элементов

1. Плавающие элементы не могут выходить за край своего контейнера-родителя.
2. Каждый плавающий элемент будет отображаться справа или ниже от предыдущего при float:left, либо слева и ниже, при float:right.
3. Блок с float:left не может быть правее, чем блок с float:right.
4. Плавающий элемент не может выходить за пределы верхней границы своего контейнера.
5. Плавающий элемент не может располагаться выше, чем родительский блок или предыдущий плавающий элемент.
6. Плавающий элемент не может располагаться выше, чем предыдущая строка inline-элементов
7. Плавающий блок должен быть расположен как можно выше.
8. Один плавающий элемент, следующий за другим, не может выходить за пределы своего контейнера — происходит перенос на следующую строку.
9. Блок с float:left должен быть расположен как можно левее, а с float:right — как можно правее.

# Практика

<https://drive.google.com/file/d/0B2ojNJUDLrjEbDRIS2NsYWNldE0/view?usp=sharing>

1. Используем float'ы! Сделать верхнюю часть(header) – обязательно!
2. Внизу – выравниваем display: inline-block'ом
3. Когда сделали header – позовите меня.
4. Время на выполнение **30 минут**. Если не успеваете – останавливайтесь.