

# HTML. CSS. Виды вёрстки

Основы HTML и CSS. Табличная и блочная вёрстка. Фиксированная и резиновая вёрстка. Адаптивный дизайн

# Разметка

- Логическая – указывает на место и функцию элементов в логической структуре документа
- Визуальная (физическая) – указывает как именно (в каком виде) представить элемент пользователю

# HTML

- **Hypertext Markup Language** или язык гипертекстовой разметки
- Предназначен для описания структуры и форматирования документов
- Создатель – Тим Бернерс-Ли (86-91гг.)
- **Основная функция:** описание логической структуры страницы

# HTML

- Описание структуры и форматирование элементов документа осуществляется с помощью дескрипторов – **тегов**
- **Тег** — это символьное или текстовое выражение, предписывающее предпринять какое-либо определённое действие непосредственно по *разметке и форматированию* экранного представления *веб-страницы*.

Обычный текст **<i>курсив</i>** обычный  
текст

# HTML. Атрибуты тега

```
<table width="100px" border="1">
```

```
...
```

```
</table>
```

**Практически все теги** обладают **атрибутами**. Изменение **значения** этих **атрибутов**, позволяет использовать **дополнительные возможности** в **форматировании документа**

# HTML. Строчные элементы

- Располагаются и являются частью одной строки. Строка не переносится без необходимости
- Нельзя задавать ширину и высоту, поскольку они определяются содержимым строчного элемента
- Внутри строчного элемента можно вкладывать другие строчные элементы, но не блочные
- Предназначены для оформления документа на уровне отдельных фраз текста

# HTML. Строчные элементы

- `<span></span>` - выделяет фрагмент текста
- `<strong></strong>`, `<b></b>` - жирный шрифт\*
- `<em></em>`, `<i></i>` - курсив
- `<a></a>` - ссылка
- `<img />` - изображение
- И т.д.

# HTML. Блочные элементы

- Располагаются по вертикали друг за другом в новой строке
- По ширине занимают всю строку или доступное пространство
- Можно задать ширину и высоту
- Можно вкладывать другие блочные элементы
- Используются для выделения блоков текста (заголовков, абзац и т.д.) и для форматирования структуры документа

# HTML. Блочные элементы

- `<table></table>` - вывод таблицы
- `<div></div>` - создание блока
- `<h1></h1>` - заголовок
- `<p></p>` - абзац
- `<ul></ul>` - маркированный список
- И т.д.

# Структура HTML-документа

`<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">` - описывает версию документа

`<html>` - открывает документ

- `<head></head>` - «заголовок»

- `<body></body>` - «тело»

`</html>` - закрывает документ

# <head></head> содержит

- Техническую информацию о документе (напр. Кодировку)
- Название документа (<title></title>)
- Стили или ссылку на стили
- Мета-теги (например, ключевые слова)
- Различные скрипты (javascript, счётчики посещаемости и т.д.)
- **Основная функция** – описание документа

# <body></body> содержит

- Разметку страницы
- Контент, содержание документа
- **Основная функция** – вывод информации на экран

# Версии HTML

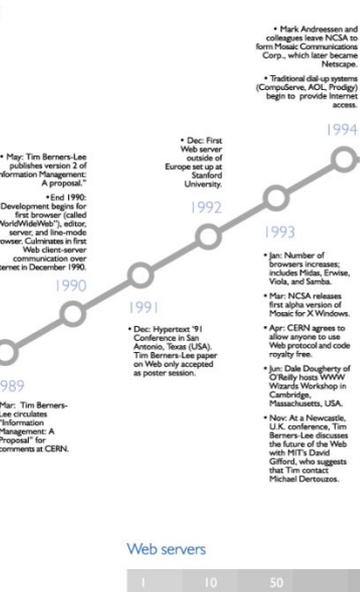
- HTML 1.0 - ~85-91
- HTML 2.0 – 22.09.1995
- HTML 3.2 – 14.01.1997
- HTML 4.0 – 18.12.1997
- HTML 4.01 – 24.12.1999
- **HTML 5.0 – 28.10.2014**

# World Wide Web Consortium



## Pre-W3C Web and Internet Background

- 1945: Vannevar Bush article in Atlantic Monthly describes a photo-electrical-mechanical device called a Memex, for memory extension, which could make and follow links between documents on microfiche.
- 1960: J.C.R. Licklider publishes "Man-Computer Symbiosis."
- 1962: Douglas Engelbart publishes "Augmenting Human Intellect: A Conceptual Framework."
- 1965: Ted Nelson coins the term "Hypertext" in "A File Structure for the Complex, the Changing, and the Interconnected," 20th National Conference, New York, Association for Computing Machinery.
- 1968: Douglas Engelbart demonstrates Online System (NLS).
- 1969: Advanced Research Projects Agency commissions ARPANET to conduct research on networking.
- 1971: Ray Tomlinson of BBN creates email program to send messages across a distributed network.
- 1972: Tomlinson expands program to ARPANET users, using the "@" sign as part of the address.
- 1974: Vint Cerf and Bob Kahn publish, "A Protocol for Packet Network Interconnection", which specifies in detail the design of a Transmission Control Protocol (TCP).
- 1978: Part of TCP published separately as the Internet Protocol (IP).
- 1984: Paul Mockapetris introduces Domain Name System (DNS).
- 1989: Marc Tim Berners-Lee circulates "Information Management: A Proposal" for comments at CERN.



## W3C

- Apr: INRIA becomes W3C Host in Europe.
- Jun: W3C holds first Workshop, on Content Rating; leads to FICS.
- Jun: In response to "Body meeting" W3C forms Process ERB.
- Sep: Keio University becomes W3C Host in Asia.
- Nov: W3C launches Offices program.
- W3C Creates Advisory Board (AB).
- Jul: W3C charts Technical Architecture Group (TAG).
- Jan: ERCIM replaces INRIA as W3C Host in Europe.
- Feb: W3C adopts Royalty-Free Patent Policy.

## Starting year of current W3C Activities

- Graphics
- Hypertext Markup Language (HTML)
- Style
- Math
- Extensible Markup Language (XML)
- Document Object Model (DOM)
- Patent Policy
- Privacy
- Synchronized Multimedia
- Web Accessibility Initiative (WAI)
- Internationalization
- Voice Browser
- URI
- Device Independence
- Semantic Web
- XML Key Management
- Quality Assurance
- Multimodal Interaction
- Web Services
- XForms
- Compact Document Formats

## Web conference

- Geneva (Jun)
- Chicago (Oct)
- Darmstadt (Apr)
- Boston (Dec)
- Paris, Largest conference to date.
- Santa Clara
- Brisbane
- Toronto
- Amsterdam
- Hong Kong
- Honolulu
- Budapest
- New York

## Web servers



# CSS

- **Cascading Style Sheets** или каскадные таблицы стилей
- Язык описания внешнего вида документа
- **Основная функция:** описание внешнего вида, визуального представления элементов страницы, т.е. физической разметки

Тем самым, с помощью CSS удалось разделить логическую и визуальную структуры документа

# Версии CSS

- CSS1 – 17.12.1996 – выравнивание, габариты, отступы, параметры шрифтов, цвета,
- CSS2 – 12.05.1998 – блочная вёрстка, виды позиционирования, вид устройства, расширенные селекторы, динамическое содержимое и т.д.
- CSS2.1 – 7.06.2011 – в основном – исправление ошибок
- **CSS3 – 2011-2014 и сейчас дорабатывается**

# Синтаксис CSS

селектор                      свойство                      значение

```
body { background: #ffc910; }
```

The diagram illustrates the syntax of a CSS rule. It shows the code 'body { background: #ffc910; }' with three horizontal brackets above it. The first bracket is above 'body' and labeled 'селектор'. The second bracket is above 'background:' and labeled 'свойство'. The third bracket is above '#ffc910;' and labeled 'значение'. The word 'body' is colored blue, 'background:' is colored red, and '#ffc910;' is colored green.

# Селекторы CSS

- Универсальный \* {описание}
  - Селекторы тегов тег {...}
  - Селекторы атрибутов [атрибут тега] {...}
  - Классы .класс {...}
  - Идентификаторы #id {...}
  
  - Псевдокласс селектор: псевдокласс
  - Псевдоэлемент {...}
- так же

# Селекторы CSS: селекторы ТЕГОВ

В CSS:

наименование тега {свойство:значение;}

В HTML:

<тег> содержимое </тег>

# Селекторы CSS: классы

В CSS:

.название класса {свойство:значение;}

В HTML:

<тег class="название класса"> содержимое </тег>

**Когда нужно определить стиль для индивидуального элемента веб-страницы или задать разные стили для одного тега**

# Селекторы CSS: идентификаторы

В CSS:

```
#идентификатор {свойство:значение;}
```

В HTML:

```
<тег id="идентификатор"> содержимое  
</тег>
```

**Когда нужно определить стиль для индивидуального элемента веб-страницы + для обращения скриптов к данному элементу**

# Селекторы CSS: псевдоклассы и псевдоэлементы

селектор:псевдокласс(элемент) {свойство:  
значение;}

**Псевдоклассы:** описание свойств элементов, которые могут меняться в зависимости от действий пользователя, а также описание свойств элементов в зависимости от положения в документе.

:first-child, :last-child, :hover, :focus, :link, :active, :visited, :lang

**Псевдоэлементы:** генерация содержимого, которого нет в исходном коде страницы, описание элементов, не определенных в древе документа.

:before, :after, :first-line, :first-letter

# Способы добавления стилей

- Внутренние (как атрибут тега)

```
<div style="font-size: 120%; font-family: monospace; color: #cd66cc"></div>
```

- Глобальные стили (в head документа)

```
<head>
```

```
<style>
```

```
селектор {свойство:значение;}
```

```
</style>
```

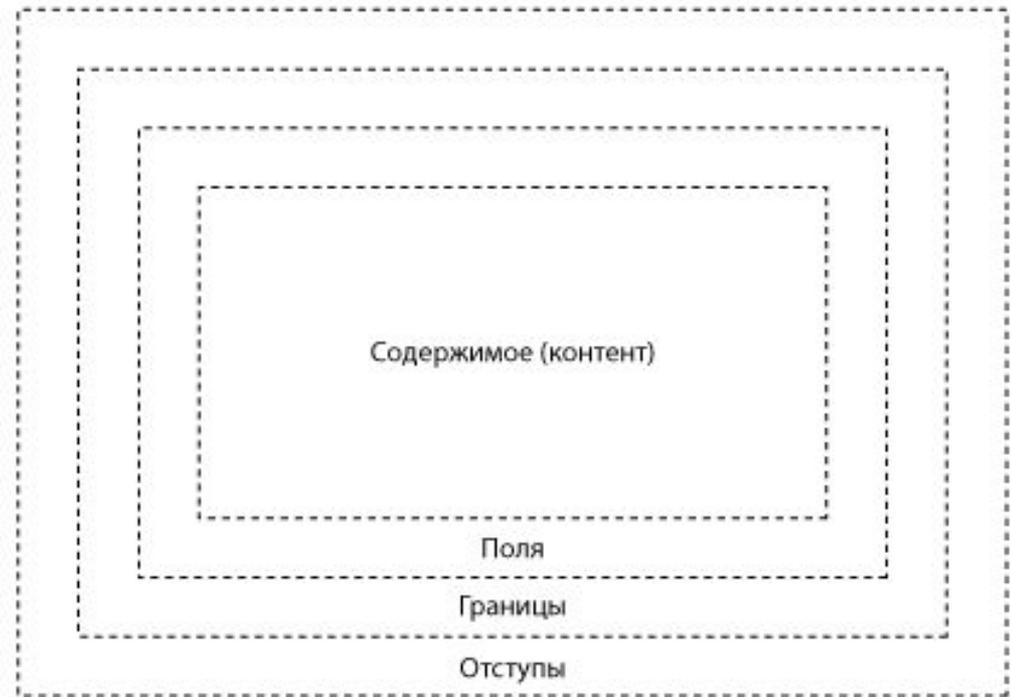
```
</head>
```

- Связанные стили (в отдельном файле)

```
<link rel="stylesheet" href="адрес файла стилей.css">
```

# Свойства CSS

- Позиционирование блоков: `position`; `left`, `top`, `bottom`, `right`; `z-index` и т.д.
- Размеры, поля, границы, отступы блочных (и некоторых строчных) элементов: `height`, `width`; `margin`; `border`; `padding`;
- Обтекание и обрезка контента: `float`; `clear`; `overflow`;



# Свойства CSS

- Отображение в строке: `display`;
- Фоновое изображение и заливка фона: `background`, `background-image`;
- Настройки текста: `font-family`, `font-size` (и т. д.); `color`, `text-decoration`;

# Верстка

- Процесс формирования документа через набор кода или с помощью WYSIWYG-редакторов
- Верстка в веб обычно представляет собой реализацию разработанного дизайна в виде HTML/CSS кода
- Современная вёрстка должна отвечать стандартам HTML5 и CSS3

# Виды вёрстки: табличная вёрстка

- Таблица состоит из строк и столбцов, которые образуют ячейки
- При табличной вёрстке модульная сетка (разметка структуры) создается ячейками (строками и столбцами) таблицы
- Позиционирование элементов страницы осуществляется через помещение в ячейки

# Табличная вёрстка

- В HTML вывод таблицы осуществляется с помощью тега `<table>`. Вложенный тег `<tr>` формирует строку таблицы. В неё вкладывается тег `<td>`, который формирует ячейку в данной строке, т.е. столбец.

```
<table>
  <tr>
    <td> Ячейка 1 </td>
    <td> Ячейка 2 </td>
    <td> Ячейка 3 </td>
  </tr>
  <tr>
    <td> Ячейка 4 </td>
    <td> Ячейка 5 </td>
    <td> Ячейка 6 </td>
  </tr>
</table>
```

Ячейка 1	Ячейка 2	Ячейка 3
Ячейка 4	Ячейка 5	Ячейка 6

# Табличная вёрстка

- Количество столбцов в каждой строке должно быть одинаковым (но можно объединять ячейки по вертикали и горизонтали с помощью атрибутов `rowspan="число"` и `colspan="число"` тега `<td>`)

```
<table>
  <tr>
    <td rowspan="2"> Ячейка 1 и 4 </td>
    <td colspan="2"> Ячейка 2 и 3 </td>
  </tr>
  <tr>
    <td> Ячейка 5 </td>
    <td> Ячейка 6 </td>
  </tr>
</table>
```

Ячейка 1 и 4	Ячейка 2 и 3	
	Ячейка 5	Ячейка 6

# Табличная вёрстка

- В атрибутах ячейки (`<td></td>`) или соответствующем селекторе CSS можно указывать:
- Выравнивание содержимого по горизонтали (`align=""`) и вертикали (`valign=""`)
- Ширину (`width=""`) и высоту (`height=""`) ячейки. При этом, ячейки обладают одинаковой шириной по столбцу и одинаковой высотой по строке, т.е. **зависят друг от друга**

# Табличная вёрстка

- В каждую ячейку можно вкладывать новую таблицу или любой другой блочный элемент
- Можно настраивать различные свойства ячеек (отступы, границы, поля) и вложенных элементов с помощью CSS

# Табличная вёрстка: плюсы и минусы

## Преимущества:

- Простота по сравнению с блочной
- Кроссбраузерность

## Недостатки:

- Громоздкий код
- Недостаточная гибкость

# Виды вёрстки: блочная вёрстка

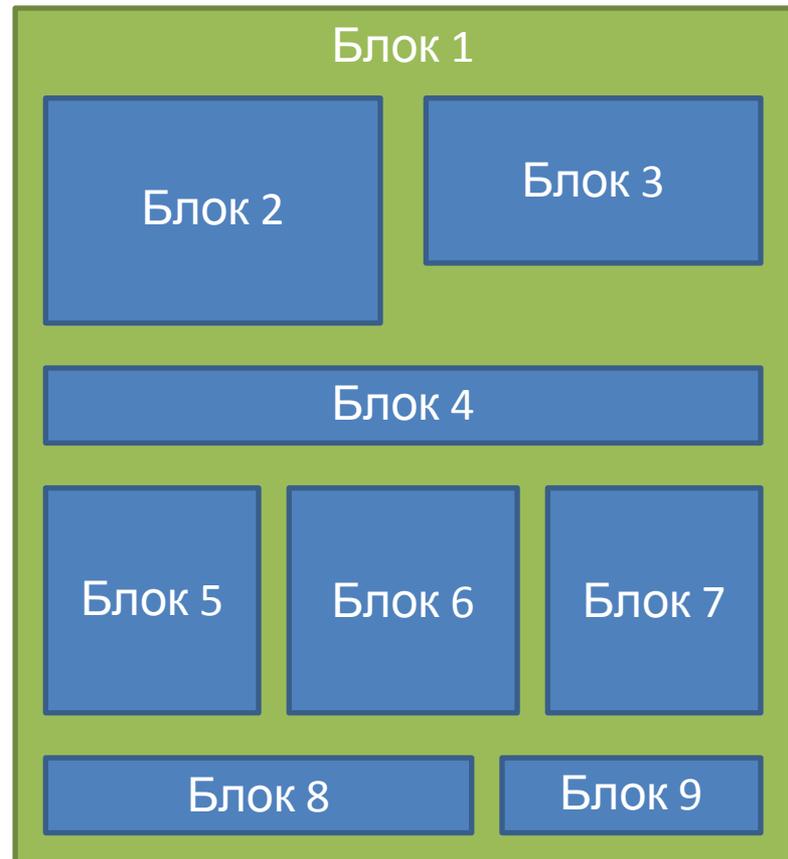
- Разметка страницы осуществляется с помощью универсальных блочных элементов `<div></div>` (слоёв)
- При этом HTML-код содержит только теги разметки и логического форматирования, а всё визуальное оформление выносится в CSS
- Таблицы используются только для вывода табличных данных

# Блочная вёрстка

- `<div>` представляет собой прямоугольную область
- Блоку присваивается класс из таблицы CSS
- В CSS настраивается позиционирование блока, выравнивание относительно других элементов, строчность\блочность\*, поля, отступы, границы, фон, выравнивание и стили содержимого

# Блочная вёрстка

- Габариты каждого блока не зависят от габаритов любого другого
- Можно выводить блоки из потока документа с помощью позиционирования в CSS
- В `<div>` можно вкладывать любые строчные или блочные элементы в т.ч. другие `<div>`



# Блочная вёрстка: плюсы и минусы

## Преимущества:

- Более компактный код
- Намного больше возможностей, гибкость
- Блоки загружаются быстрее таблиц

## Недостатки:

- Требуется больше знаний и навыков

# <table> против <div>

	Таблица	div
Колонки	Колонки формируются ячейками таблицы, их высота одинакова и взаимосвязана.	Колонки создаются разными слоями, их высота разная и зависит от содержания.
Ширина	Если ширина таблицы явно не указана, она вычисляется на основе содержимого таблицы.	По умолчанию слой занимает всю доступную ему ширину.
Расположение	Строки таблицы отображаются в том порядке, как они представлены в коде. Ячейки идут слева направо и сверху вниз.	Порядок слоёв в коде может не соответствовать их положению в браузере.
Загрузка	Как правило, пока таблица не загрузится полностью, содержимое её не будет показываться. Если на веб-странице размещена большая таблица, загрузка страницы существенно замедляется.	Слои отображаются последовательно, по мере загрузки документа.
HTML-код	Код для создания таблиц может быть громоздок, особенно если требуется объединить множество ячеек или сделать несколько вложенных таблиц.	Код, как правило, компактный.

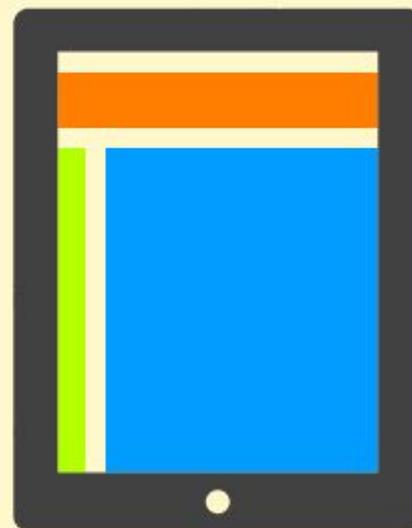
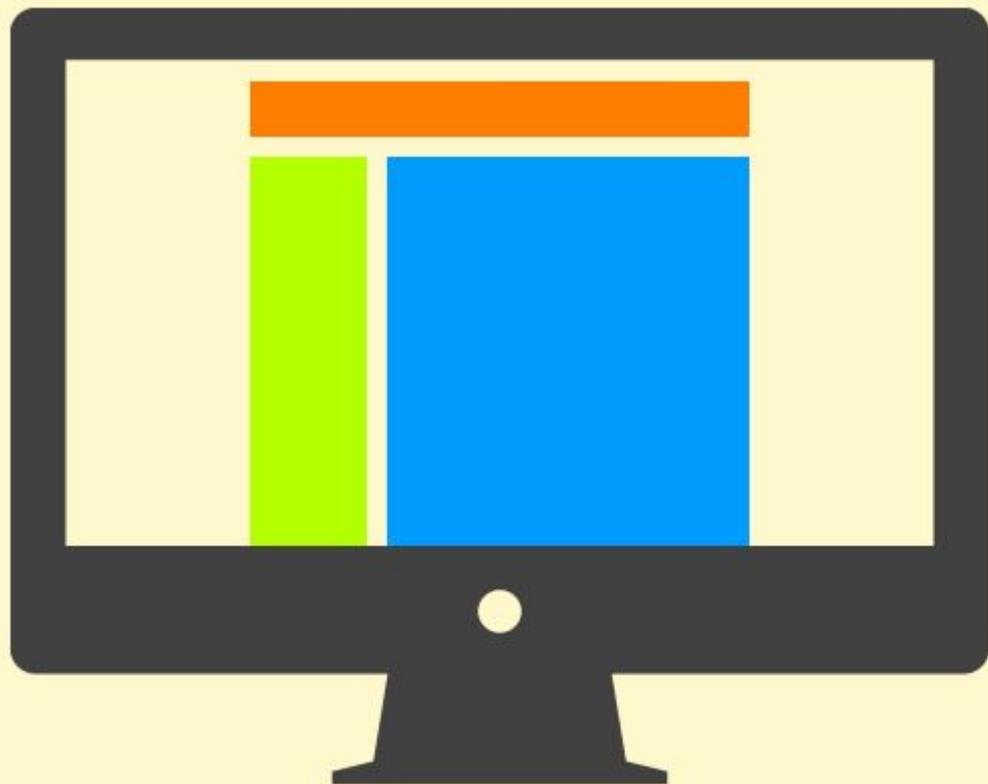
# Виды вёрстки (дизайна)

- Фиксированная
- Резиновая
- Адаптивная\отзывчивая

# Фиксированная вёрстка

- Размеры всех элементов жестко определены (заданы в пикселях)
- При изменении разрешения (размера окна браузера), размер элементов в пикселях не меняется
- Простота перевода дизайна в код
- Некорректное отображение при изменении разрешения

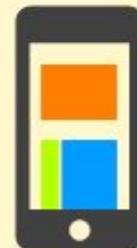
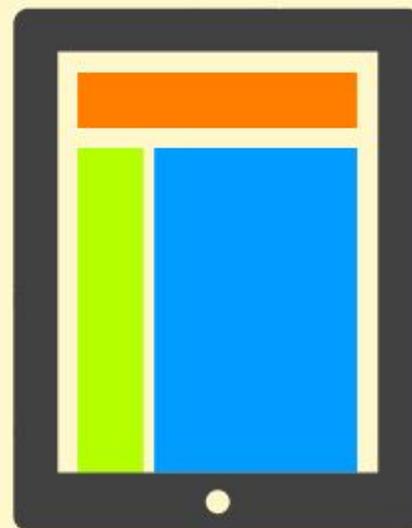
# Фиксированная вёрстка



# Резиновая вёрстка

- Размеры всех элементов заданы в процентах от разрешения (размера окна)
- При изменении разрешения (размера окна браузера), размер элементов изменяется пропорционально
- Более-менее подстраивается под размер экрана
- Сложнее корректно реализовать дизайн в коде.

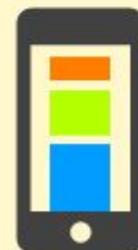
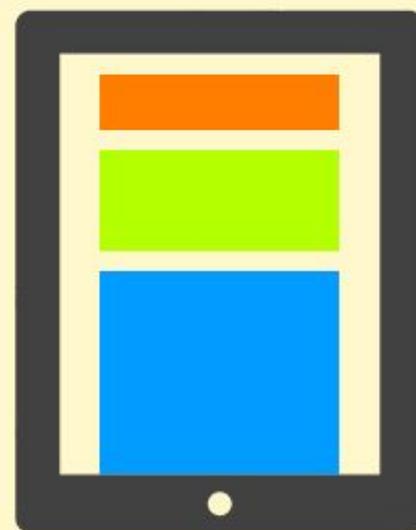
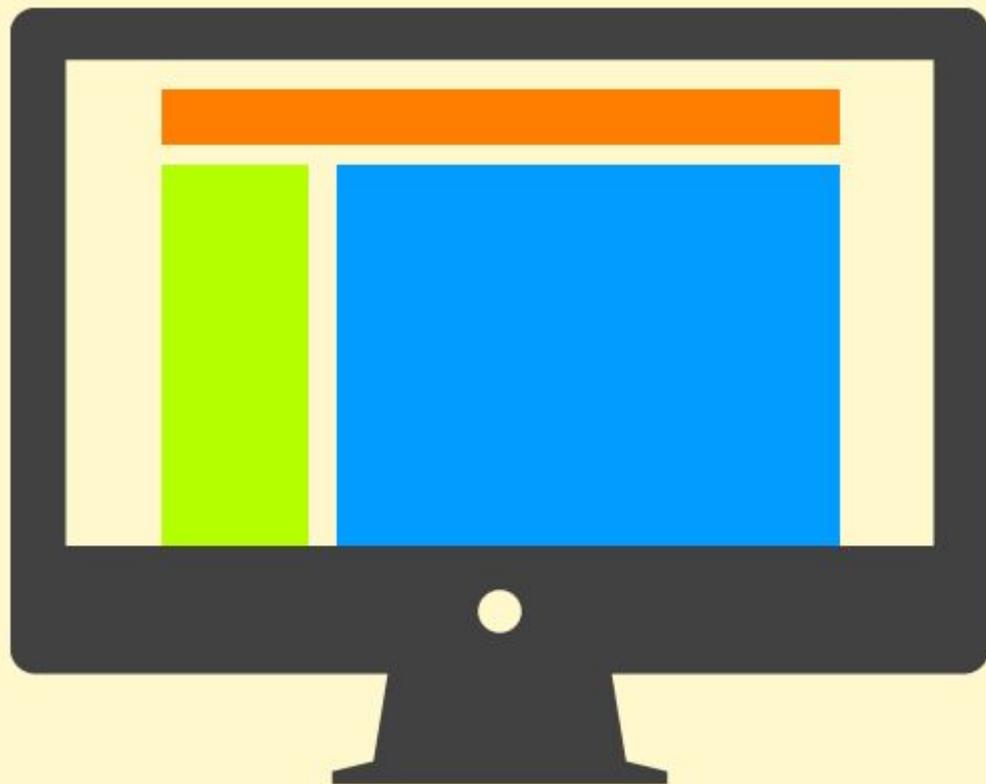
# Резиновая вёрстка



# Адаптивная вёрстка

- Адаптируется к набору разрешений (прыгает по контрольным точкам). Размеры элементов указываются в пикселях, при этом значения изменяются в зависимости от разрешения.
- Подстраивается под любое устройство и размер экрана (с нюансами). Позволяет избавиться от «мобильных версий»
- Гораздо сложнее фиксированной и резиновой

# Адаптивная вёрстка



# Отзывчивая вёрстка

- Приспосабливается к любому разрешению. Размеры элементов указываются в относительных единицах.
- При изменении разрешения (размера окна браузера), размер элементов и их положение изменяется так, как это необходимо для корректного отображения
- Подстраивается под любое устройство и размер экрана. Позволяет избавиться от «мобильных версий»
- Гораздо сложнее фиксированной и резиновой

# Отзывчивая вёрстка

