



Курс

«Программирование в компьютерных сетях»

Лекция 3

Приходько Татьяна Александровна

доцент кафедры

Вычислительных технологий КубГУ





Лекция 3

HTML

Язык описания Web-страниц

Основы верстки

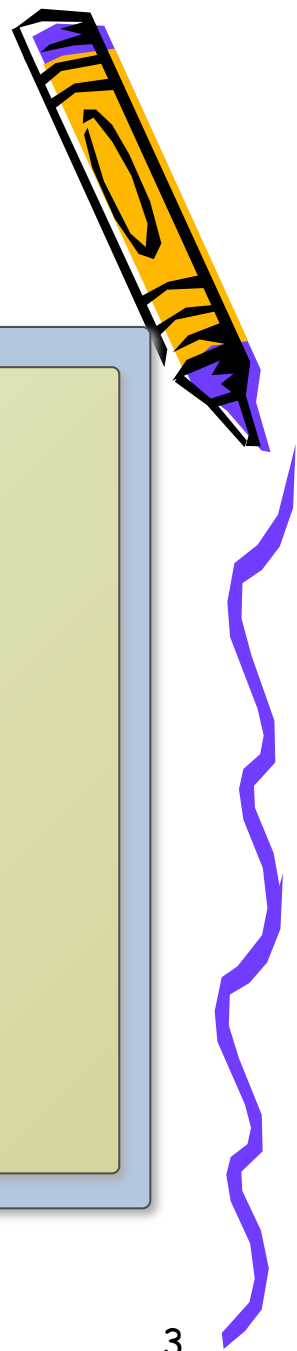
С использованием материалов сайта htmlbook.ru



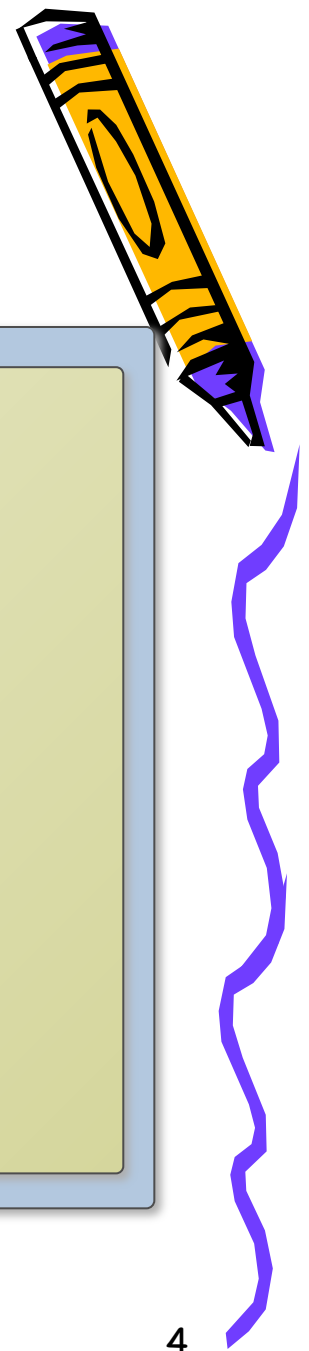
Верстка

План лекции

- Алгоритм верстки
- Виды макетов
- Изображения
- Табличная верстка
- Блочная верстка
- Задания для лабораторной



Верстка



План лекции

- Алгоритм верстки
- Виды макетов
- Изображения
- Табличная верстка
- Блочная верстка
- Задания для лабораторной



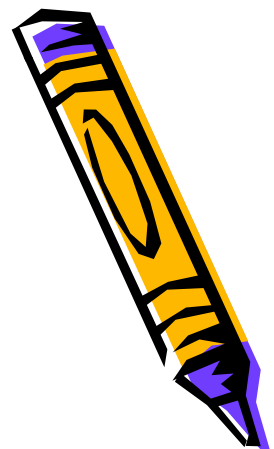
Алгоритм верстки

Верстка это процесс творческий и четких алгоритмов здесь не существует

Вначале дизайнер готовит макеты веб-страниц в графическом редакторе (например, Adobe Illustrator, Adobe Photoshop), утверждает их у заказчика и передает верстальщику на формирование HTML-кода.

Верстальщик получает работу в виде набора рисунков, где каждый из них соответствует макету отдельной страницы со своим дизайном.

Теперь необходимо проанализировать рисунок и решить, как же его превратить в веб-страницу. Для удобства происходит логическое разбиение картинки на отдельные блоки, с которыми идет дальнейшая работа.



Алгоритм верстки

Рассмотрим пример, приведенный на <http://htmlbook.ru/>

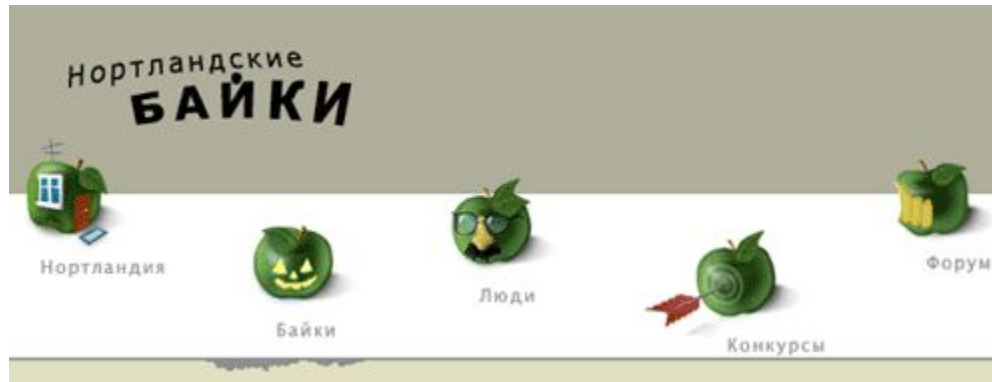
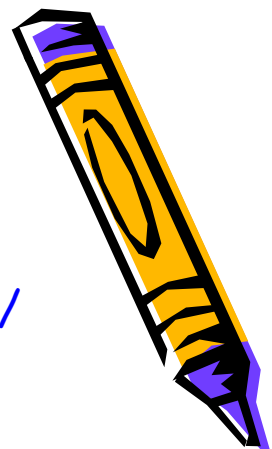
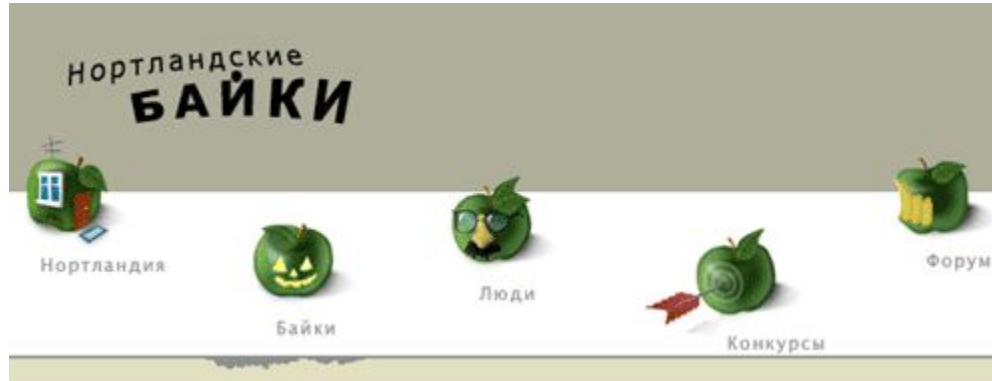


Рис. 1. Изображение «шапки» главной страницы

По задумке дизайнера серая и белая полоса заголовка должны занимать всю ширину веб-страницы, а набор пиктограмм с заголовком сайта выравнивается по центру окна браузера. Положение рисунков относительно друг друга меняться не должно и каждый из них является ссылкой на определенный раздел сайта. С учетом указанных особенностей возможны следующие варианты.



Алгоритм верстки (продолжение)



Варианты создания меню:

- Сделать один рисунок и применить к нему карту-изображение.
- Разрезать изображение на фрагменты и объединить их воедино с помощью таблицы, при этом отдельные фрагменты будут служить ссылкой.
- Воспользоваться позиционированием элементов.



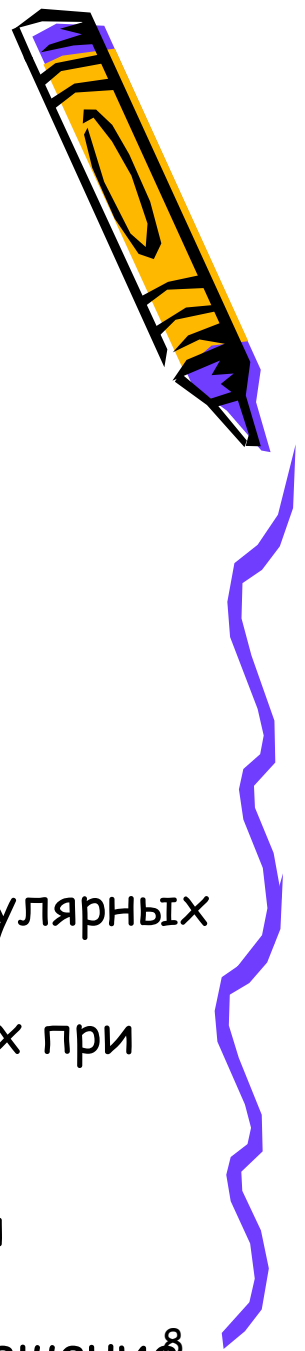
Алгоритм верстки (продолжение)

Далее предстоит продумать:

- цвет фона веб-страницы;
- тип разметки - табличная/блочная?
- гарнитуру основного шрифта, его размер и цвет;
- размер текста отдельных модулей (новостей, например);
- цвет, размер и гарнитуру шрифта заголовков;
- параметры горизонтальных линий и рамок;
- создать CSS файл.

После того, как все готово, проверить:

- корректно ли отображаются страницы в популярных браузерах?
- происходит сохранение целостности данных при изменении размера шрифта в браузере как в большую, так и меньшую сторону?
- можно ли продолжать работу с сайтом, если отключить показ изображений?
- как существенно влияет на вид страниц разрешение монитора?



Алгоритм верстки (продолжение)

Итогом работы верстальщика является набор шаблонов (темплейтов) повторяющих рисунки дизайнера, но сделанных в виде HTML-документов. А также стилевой файл, в котором прописаны не только атрибуты, необходимые для верстки, но и параметры основного текста, заголовков, подзаголовков и других текстовых элементов. Это позволяет по единым шаблонам формировать любое число веб-страниц, оформление и вид которых будет одинаков.

1

графический
дизайн
макета



2



макет
в процессе верстки

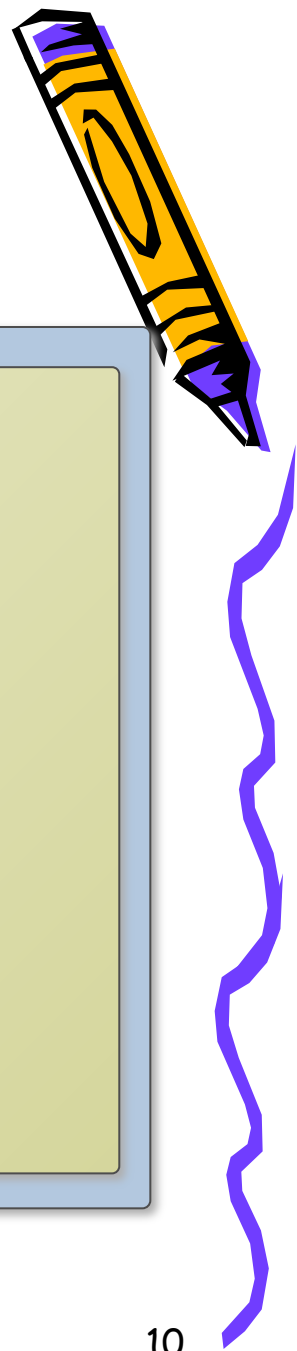
3

макет сайта
в формате HTML



html + css

Верстка



План лекции

- Алгоритм верстки
- Виды макетов**
- Изображения
- Табличная верстка
- Блочная верстка
- Задания для лабораторной



Виды макетов



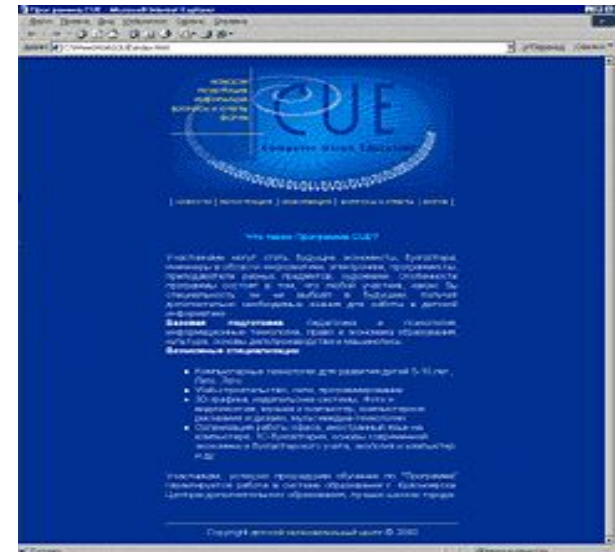
Существует два вида макетов - фиксированный и «резиновый»

Фиксированный макет

Общая ширина макета задается жестко и равна определенной величине. Как правило, отталкиваются от минимального на данный момент разрешения экрана - 1024x768, поэтому за вычетом полосы прокрутки, берется ширина макета 900 px.

При этом обнаруживается большой недостаток (особенно на больших экранах) - неэффективное использование свободной площади.

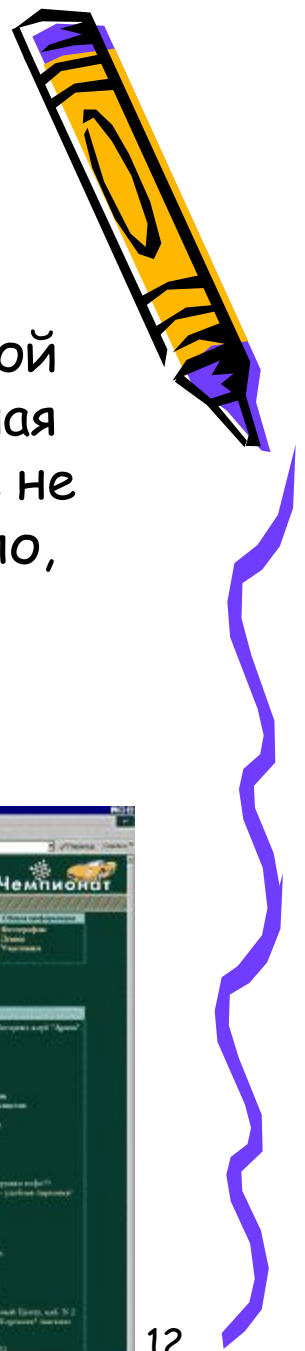
*Но зато верстка упрощается.
Можно использовать
табличную верстку.*



Виды макетов (продолжение)

«Резиновый» макет

Этот вид макета основывается на том, что в качестве одной из единиц измерения выступают проценты. Общая рабочая ширина окна браузера — 100%, и колонки макета в сумме не должны ее превышать, поэтому для удобства, как правило, везде применяют процентную запись. При изменении размеров окна происходит переформатирование данных страницы, чтобы они вписались в новую ширину.



Практически все известные сайты выбрали именно его в силу того, что эффективно задействуется вся площадь веб-страницы.



Но при этом обнаруживается некоторый набор недостатков.



Виды макетов (продолжение)

Недостатки «резиновой» верстки

- Веб-страница подстраивается под ширину окна браузера, и при достижении некоторой величины читать текст становится неудобно — строки слишком длинные и глаза устают по ним бегать. Впрочем, браузер можно свернуть в окно, подобрав его комфортный размер.
- Верстать «резиновый» макет сложнее, чем аналогичный фиксированной ширины - приходится учитывать множество дополнительных факторов и знать некоторые приемы верстки. К тому же, популярные браузеры неоднозначно трактуют некоторые параметры и в них «резиновый» макет может отображаться по-разному. Говоря проще, знать о правилах верстки надо больше.



Как правило, используется *блочная верстка* или *верстка слоями*.

Виды макетов (продолжение)

Недостатки «резиновой» верстки

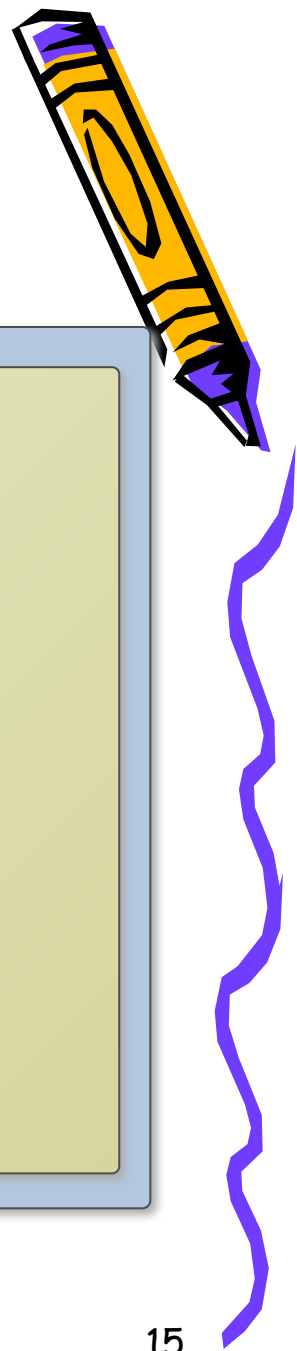
- Любой макет имеет некоторую минимальную ширину, при достижении которой веб-страница «рассыпается» или появляется горизонтальная полоса прокрутки. Например, если в документ вставлен рисунок шириной 600 пикселей, то при уменьшении окна до этой величины браузер начнет отображать полосу прокрутки.
- «Резиновый» дизайн характеризуется активным использованием фоновых изображений, которые по горизонтали собираются без швов встык. Действительно, изменить без потери качества ширину рисунков не получится, а вот «подложить» под них фон всегда пожалуйста. Как правило, используется **блочная верстка** или **верстка слоями**.



Верстка

План лекции

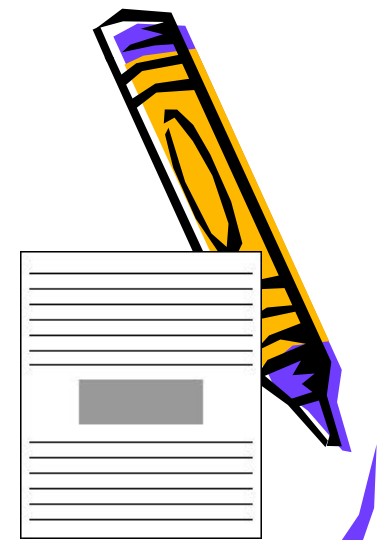
- Алгоритм верстки
- Виды макетов
- Изображения**
- Табличная верстка
- Блочная верстка
- Задания для лабораторной



Изображения

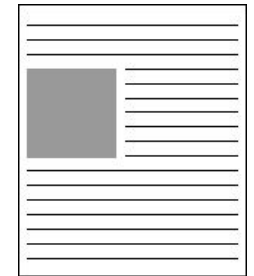
Рисунок по центру

Для выравнивания изображения по центру колонки текста, тег `` помещается в контейнер `<p>`, для которого устанавливается атрибут `align="center"`.



Обтекание рисунка текстом

Снова используем атрибут `align` тега ``. Этот атрибут задает, возле какого края окна будет располагаться рисунок, одновременно устанавливая и способ обтекания текста. Чтобы выровнять изображение по правому краю и задать обтекание слева, используют значение `right`, для выравнивания по левому краю применяют `left`.



Атрибут `align` часто используют в связке с другими атрибутами тега `` — `vspace` и `hspace`. Они определяют расстояние от обтекаемого текста до изображения. Без этих атрибутов изображение и текст будут слишком плотно прилегать друг к другу.



Изображения (продолжение)

Изображение на всю ширину макета

1. Растягивание рисунка до 100% (не самый эффективный способ, используется редко).
2. Использование бесшовного фонового изображения. Фоновый рисунок подготавливается из блоков шириной 20-30 пикселей нужной высоты. Этим изображением «вымащивается» нужная площадь.

На фоновую область можно наложить рисунок так, чтобы вместе они образовали единое целое

[ДонНТУ](#) [Портал магистров](#)  

	<p>Иваненко Иван Иванович</p> <p>Факультет компьютерных наук и технологий</p> <p>Кафедра компьютерной инженерии</p> <p>Специальность «Компьютерные системы и сети»</p>	
<p>Краткое резюме студента</p>		<p>Резюме</p> <p>Биография</p>



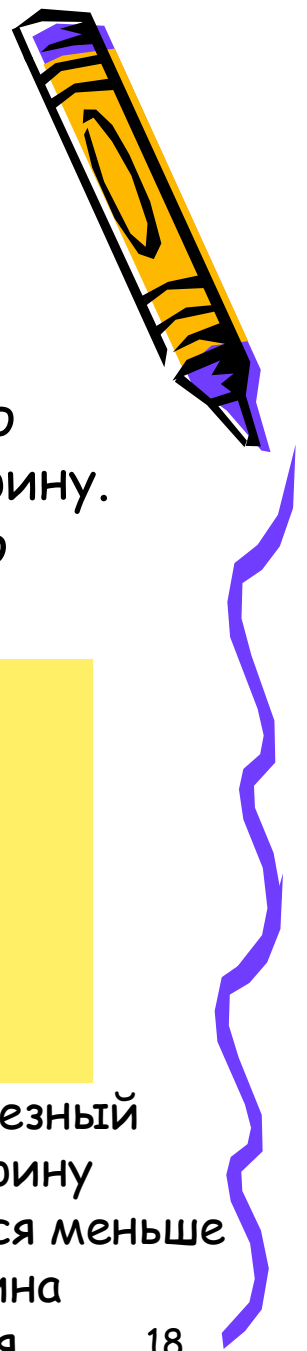
Изображения (продолжение)

Изображение на всю ширину макета

Высота блока задается с помощью свойства `height`, она совпадает с высотой рисунка, а его ширина по умолчанию равна `auto`, иными словами, занимает всю доступную ширину. Повторение фона происходит только по горизонтали, это обеспечивает значение `repeat-x` свойства `background`.

```
#headertext { /* Блок шапки (только текст) */  
  padding: 10pt 20pt 20pt 10pt;  
          /* Повторение фона */  
  background: url(..//images/bg1.gif) repeat-x;  
  height: 230px; /* Высота слоя */  
}
```

Здесь необходимо обратить внимание на один полезный атрибут, который 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 - `min-width`.





Изображения (продолжение)

Изображение на всю ширину макета

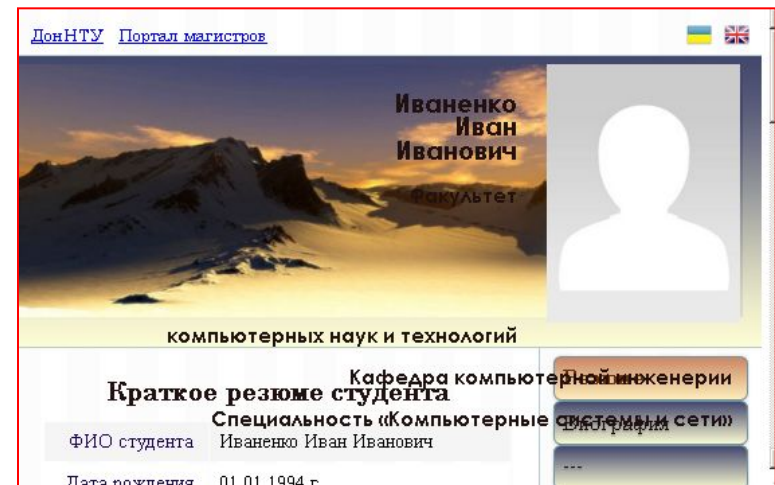
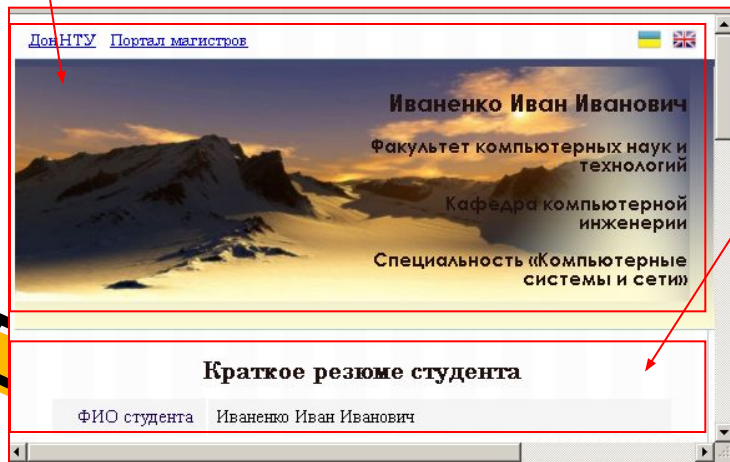
Если задать свойство **min-width** указанных ниже блоков, так:

```
header { /* Блок шапки- фото и текст) */  
  min-height: 250px;  
  min-width: 900px;  
  clear: both;}
```

```
#middleblock { /*Блок контента*/  
  border-top: solid 1pt #b4cdd4;  
  min-width: 900px;}
```

то, при минимизации окна, содержимое не
искажится, просто добавится
горизонтальная полоса прокрутки:

Иначе, получим искажение
текстового блока:



Верстка

План лекции

- Алгоритм верстки
- Виды макетов
- Изображения
- Табличная верстка**
- Блочная верстка
- Варианты дизайна персональной страницы
- Задания для лабораторной

Табличная верстка

Таблица с невидимой границей представляет собой словно модульную сетку, в блоках которой удобно размещать элементы веб-страницы. Простота и быстрота верстки, а также корректное отображение в различных браузерах — вот основные преимущества табличной верстки.

Основные достоинства таблиц:

- Простое создание колонок
- “Резиновая” верстка
- «Склейка» изображений
- Однозначная трактовка браузерами

Основные недостатки таблиц:

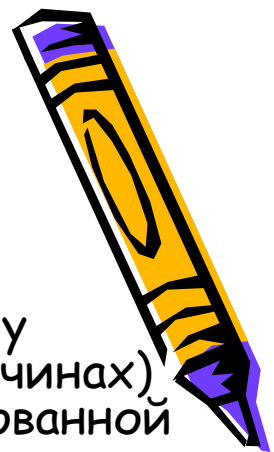
- Долгая загрузка
- Громоздкий код
- Плохая индексация поисковиками
- Нет разделения содержимого и оформления
- Несоответствие стандартам
- Невозможность конвертации в мобильную версию

Табличная верстка целесообразна когда

- Высота колонок должна быть одинаковой
- Макет должен занимать всю высоту окна браузера, независимо от объема информации
- Нет времени на сложную верстку



Секреты табличной верстки



По умолчанию ширина таблицы устанавливается браузером автоматически исходя из содержимого ячеек. Поэтому ширину таблицы указывают всегда **в процентах** (относительных величинах), для «резинового» макета или **в пикселях** для макета фиксированной ширины.

Если нужно создать ячейку фиксированной ширины, то ее размер задается в абсолютных величинах (например, пикселях). Помните, что даже фиксированная ячейка может меняться, если в ячейку добавили рисунок, размер которого превышает ширину ячейки.

Выход: 1) Рассчитывать размер содержимого под фиксированную ячейку;
2) Применить атрибут **table-layout** со значением **fixed** тега `<table>`.

Применение этого свойства позволяет обрезать рисунок, если он не помещается целиком в ячейку;

3) Воспользоваться стилевым свойством **overflow** со значением **scroll**. Это свойство добавляет полосы прокрутки к содержимому, но из-за того, что он применяется к блочным элементам им нельзя воспользоваться для тега `<td>`. Поэтому приходится вкладывать внутрь ячейки тег `<div>` и устанавливать стилевое свойство для него



Секреты табличной верстки

Скорость загрузки таблицы

Пока таблица не загрузится полностью, ее содержимое не начнет отображаться. Дело в том, что браузер, прежде чем показать содержимое таблицы, должен **вычислить необходимые размеры ячеек, их ширину и высоту**. А для этого необходимо знать, что в этих ячейках находится. Поэтому браузер и ожидает, пока загрузится все, что находится в ячейках, и только потом отображает таблицу.

Исходя из этого факта, таблицы не используют для хранения большой информации (от 100 кБ). А **чтобы ускорить загрузку табличного макета, его разбивают на отдельные таблицы или используют свойство `table-layout`**, применение которого позволяет несколько повысить скорость отображения содержимого таблицы.



Секреты табличной верстки (продолжение)



При «резиновом» макете ширина таблицы устанавливается в процентах от ширины окна браузера и, таким образом, напрямую зависит от нее. Здесь возможны два варианта:

- ширина всех ячеек задана в процентах;
- сочетание процентов и пикселей, когда ширина одних ячеек устанавливается в процентах, а других — в пикселях.

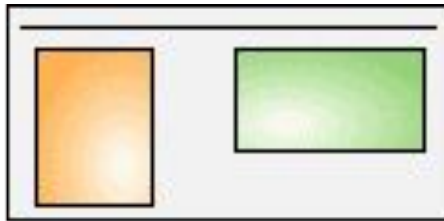
Процентная запись для таблиц имеет ряд преимуществ — используется все свободное пространство веб-страницы, а сам макет подстраивается под ширину окна браузера. Минимальный размер зависит от содержимого таблицы. Ниже дан пример таблицы: 20% : 40% : 40%



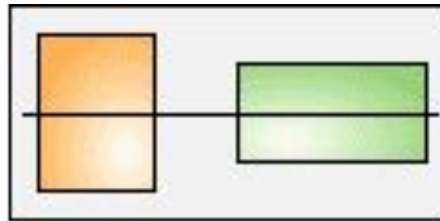
Нужно также помнить об отступах - **cellspacing** и **cellpadding**.

Секреты табличной верстки (продолжение)

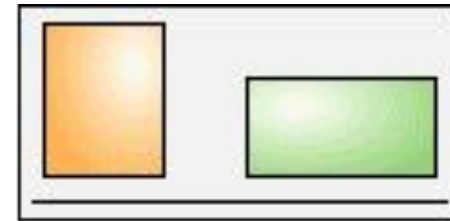
За счет сочетания атрибутов `align` (горизонтальное выравнивание) и `valign` (вертикальное выравнивание) тега `<td>`, допустимо устанавливать несколько видов положений элементов относительно друг друга.



а. Выравнивание по
верхнему краю



б. Выравнивание по
центру



в. Выравнивание по
нижнему краю

Для выравнивания содержимого ячеек по **верхнему/нижнему** краю, для тега `<td>` требуется установить атрибут `valign` со значением `top/bottom`.

Для выравнивания по центру ячейки используется `<td align="center">`.



Секреты табличной верстки (продолжение)

В данном примере характеристики ячеек управляются с помощью параметров тега `<td>`, но их удобнее изменять через стили. В частности, выравнивание в ячейках указывается свойствами CSS `vertical-align` и `text-align`:



```
<style type="text/css">
TABLE { width: 100%; /* Ширина таблицы */ }
#col1, #col3 {
  width: 10%; /* Ширина первой колонки */
  background: #f0f0f0; /* Цвет фона первой колонки */
}
#col2 {
  width: 25%; /* Ширина второй колонки */
  background: #fc5; /* Цвет фона второй колонки */
  text-align: right; /* Выравнивание по правому краю */
}
#col1, #col2 {
  vertical-align: top; /* Выравнивание по верхнему краю */
  padding: 5px; /* Поля вокруг содержимого ячеек */
}
</style>
```

```
<td id="col1"></br></br></td>
```

```
<td id="col2">
Колонка № 2 с
выравниванием по
верху</td>
```

```
<td
id="col3">Колонка
№ 3 без
выравнивания </td>
```



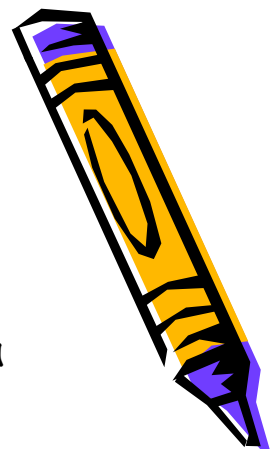
Колонка № 2 с выравниванием по верху

Колонка № 3 без
выравнивания

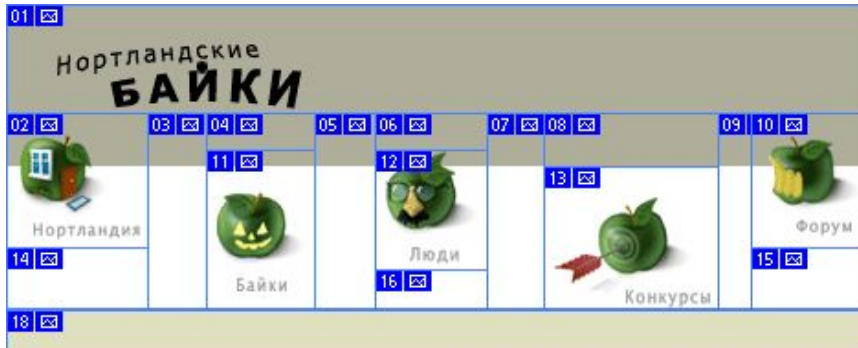
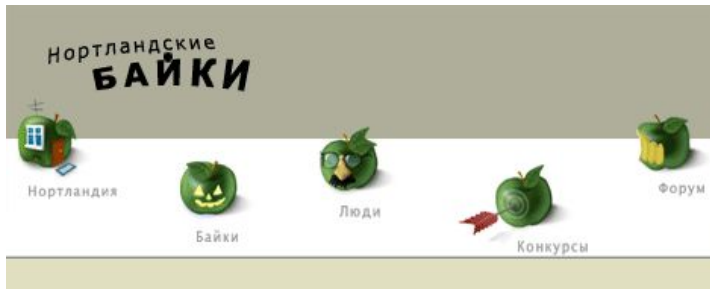


Секреты табличной верстки (продолжение)

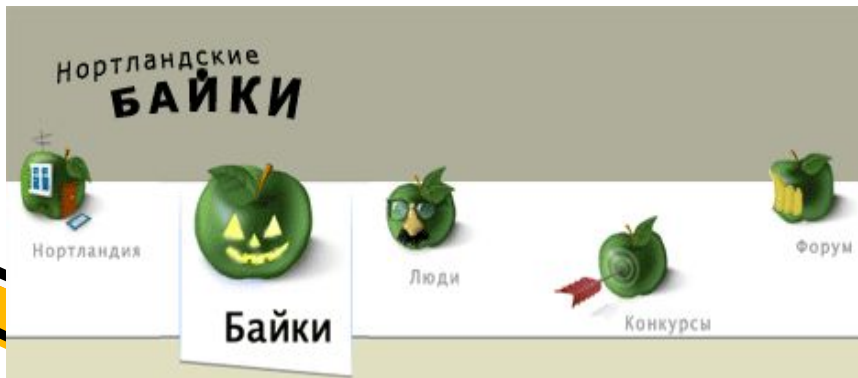
Разрезание и склейка изображений



Исходное изображение дизайна страницы, взятое с htmlbook.ru



нарезается на части в графическом редакторе, например Adobe Photoshop, который сам формирует html-код полученной таблицы.



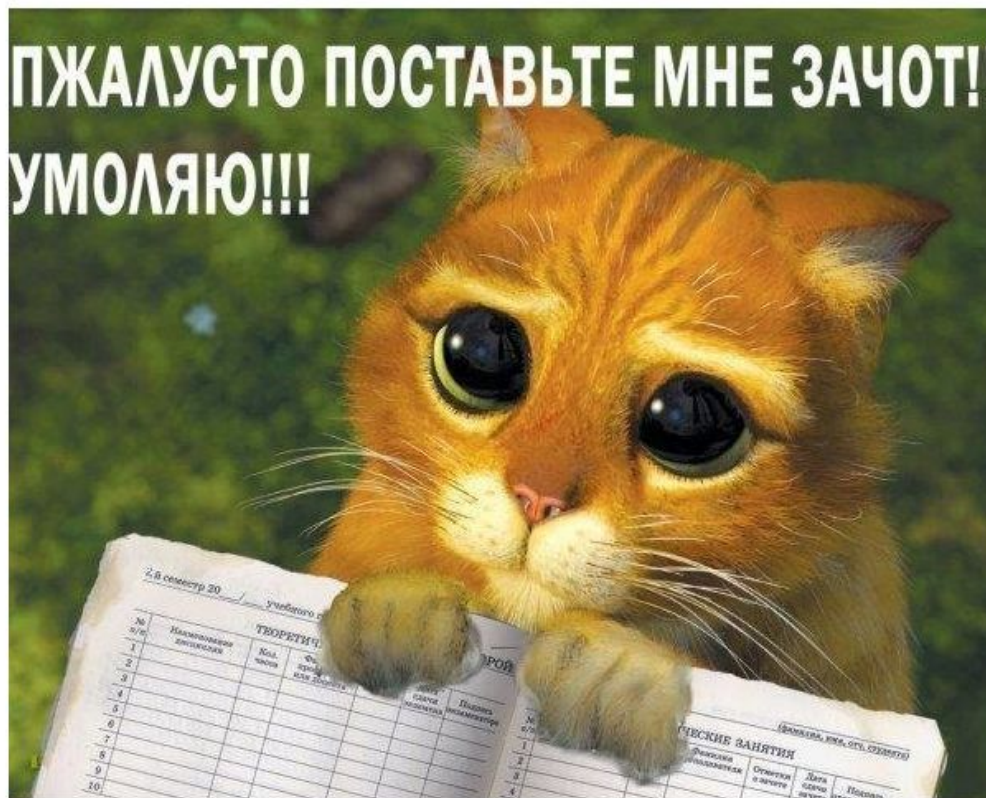
В обработчике события `JavaScript onmouseover` при наведении курсора на ячейку, ее содержимое меняется на другую картинку.

В обработчике события `onmousedown` срабатывает ссылка для перехода на другую страницу



Пример табличной разметки заглавной страницы

[RUS](#) | [UKR](#) | [ENG](#) | [ДонНТУ Портал магистров ДонНТУ](#)



**Иваненко Иван
Иванович**

Факультет:

Кафедра:

Специальность:

Контакты:



Верстка

План лекции

- Алгоритм верстки
- Виды макетов
- Изображения
- Табличная верстка
- Блочная верстка**
- Варианты дизайна персональной страницы
- Задания для лабораторной

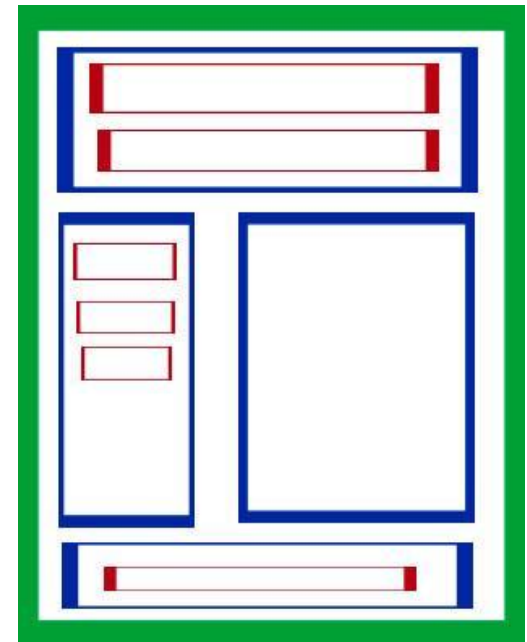
Блочная верстка (слоями)

Слой — это элемент веб-страницы, созданный с помощью тега `<div>`, к которому применяется стилевое оформление.

Благодаря этому тегу HTML-код распадается на ряд четких наглядных блоков, за счет чего верстка слоями называется также блочной версткой. Код при этом получается более компактным, чем при табличной верстке, к тому же поисковые системы его лучше индексируют.

Как уже говорилось выше, сначала нужно продумать дизайн, затем выделить отдельные смысловые части в отдельные блоки (слои). Блоки могут быть вложенными.

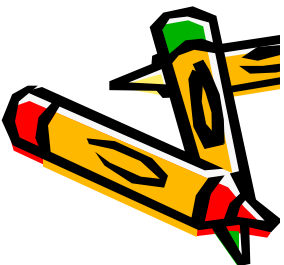
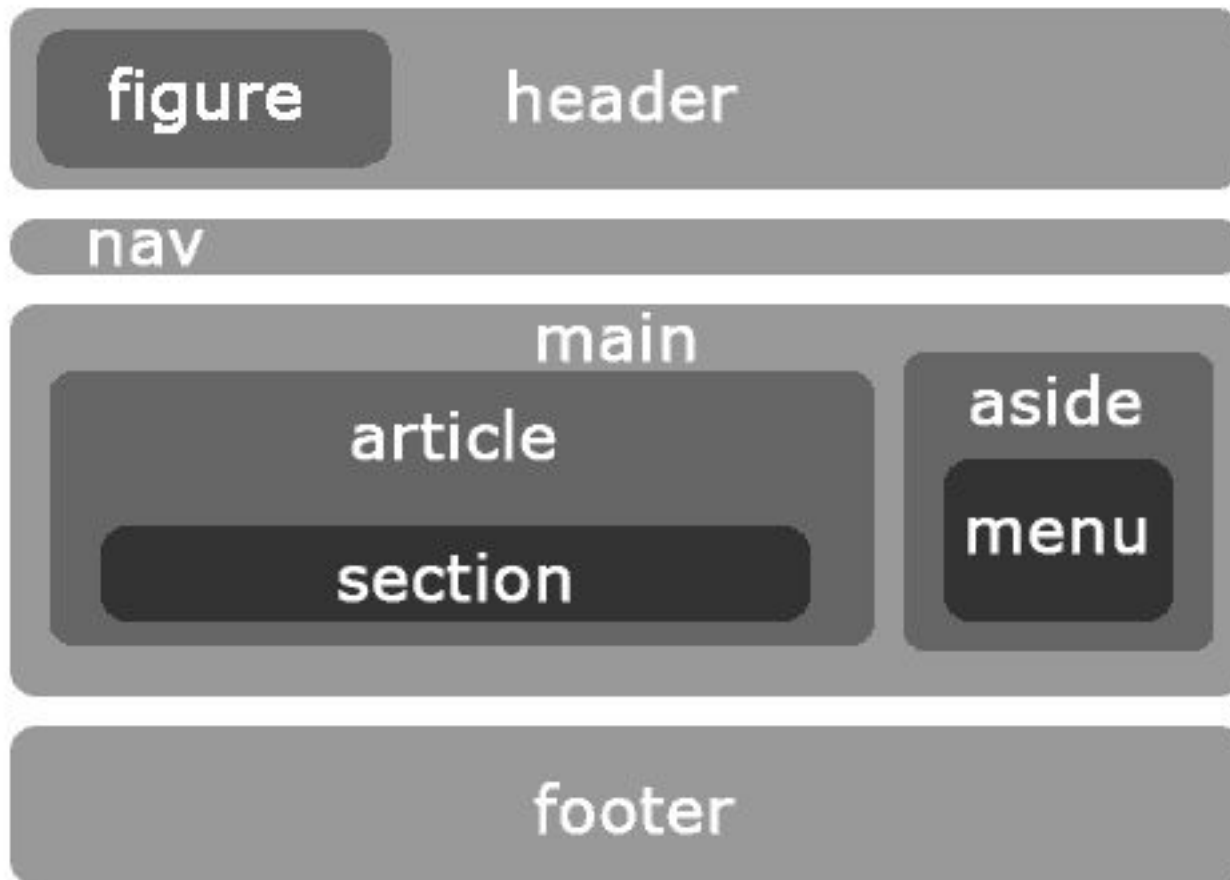
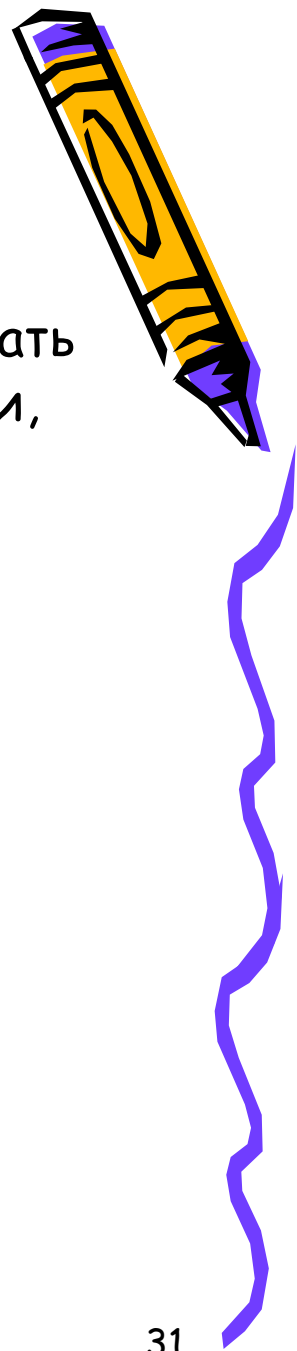
В данном случае - разметка — это, указание элементам дизайна сайта, где они должны находиться на странице. Делается это с помощью каскадных таблиц - [CSS](#).



Блочная верстка

(продолжение)

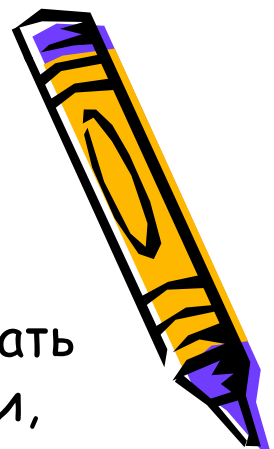
Ориентируясь на новые стандарты HTML5, лучше сразу брать курс на правильную разметку, т.е. называть блоки именами, принятыми в стандартах HTML5:



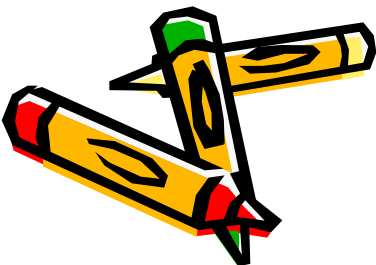
Блочная верстка

(продолжение)

Ориентируясь на новые стандарты HTML5, лучше сразу брать курс на правильную разметку, т.е. называть блоки именами, принятыми в стандартах HTML5:



```
<html>
  <head> </head>
<body>
  <div id="header"> header </div>
  <div id="nav"> nav </div>
  <div id="content"> content
  </div>
  <div id="footer"> footer </div>
</body>
</html>
```



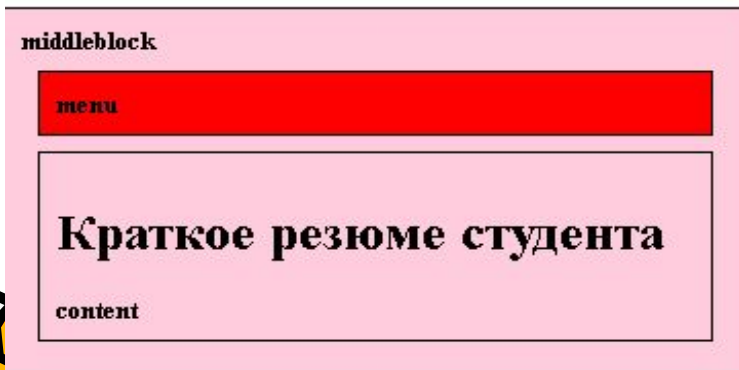
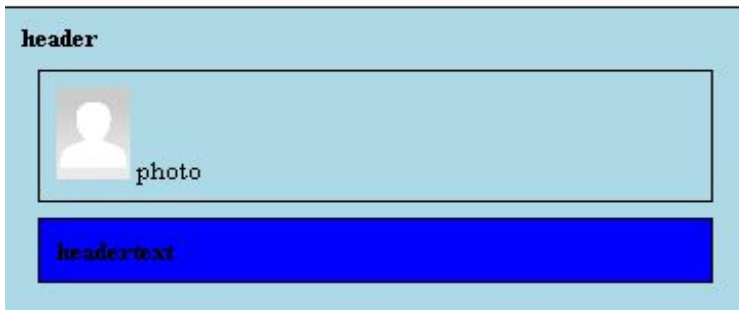
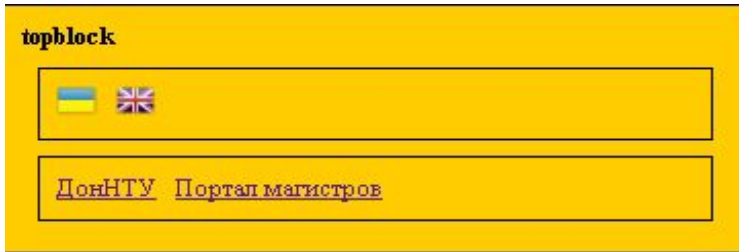
Но пока блокам не задан стиль, эта разметка отобразит следующее:



```
header
nav
content
footer
```


Блочная верстка (продолжение)

Разберем пример блочной верстки, который дан вам в качестве шаблона студенческого сайта.



Для ясности блоки шаблона окрашены в разные цвета и между ними заданы промежутки с помощью свойств `margin` и `padding`

Стилевое оформление страницы:

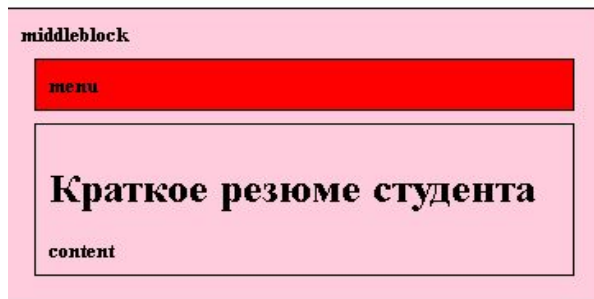
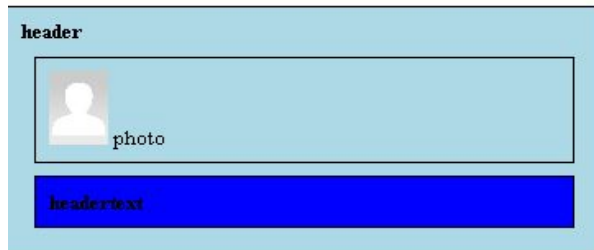
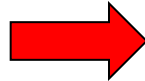
```
<style>
  div {border: 1px solid black;
padding:10px;margin:10px; }
  #topblock {background:#fc0; width:50%}
  #header {background:lightblue; width:50%}
  #middleblock {background:#fcd; width:50%}
  #nav {background:red}
  #headertext{background:blue;}
  #footer{background:lightgray; width:50%}
</style>
</head>
```



Блочная верстка (продолжение)



Разметка данной
страницы



```
<body>
  <body lang=ru>

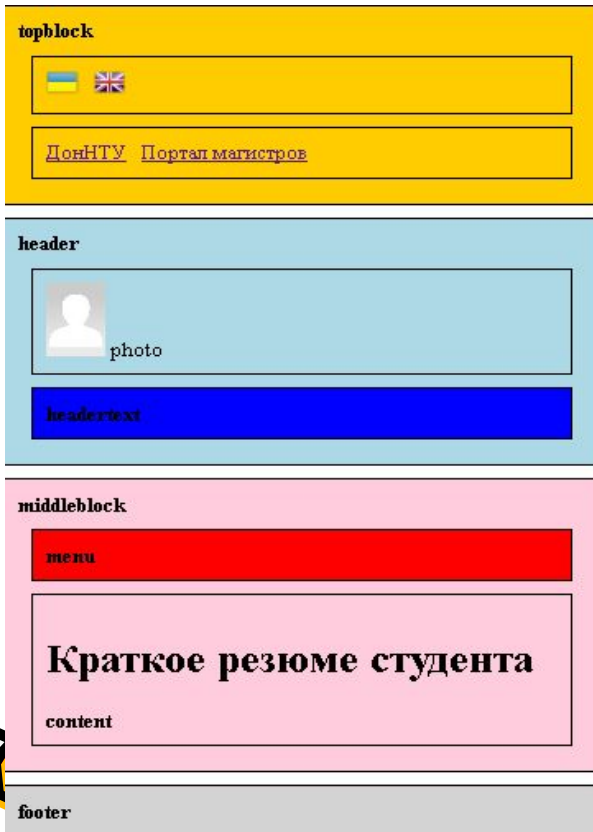
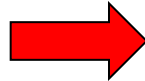
  <div id="topblock">      <strong>topblock</strong>
    <div id="langbox"> ...
  </div>
  <div id="donntu">
    <a href="http://... " ...</a>
  </div>
</div><!-- topblock -->

  <div id="header">      <strong>header</strong>
  <div id="photo">
    <a href="ivanenko_big.jpg">
  </a>
    photo
  </div>
  <div id="headertext">  <strong>
headertext</strong>
  </div>
</div> ><!-- header-->
```



Блочная верстка (продолжение)

Разметка данной
страницы
(продолжение)

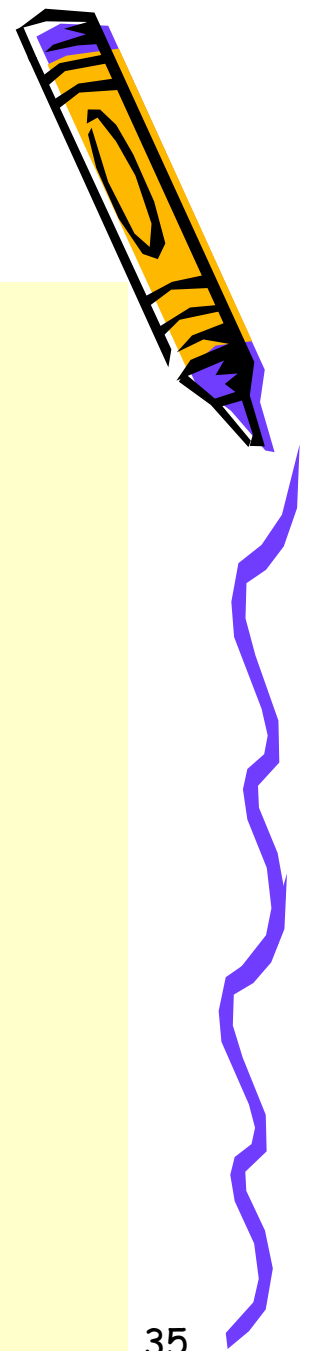


```
<div id="middleblock">  
    <strong>middleblock</strong>  
<div id="nav">  
<strong> menu</strong>  
</div>
```

```
<div id="content">  
<h1>Краткое резюме студента</h1>  
<table id="resume">  
</table>  
    <strong>content</strong>  
</div>  
</div> ><!-- middleblock -->
```

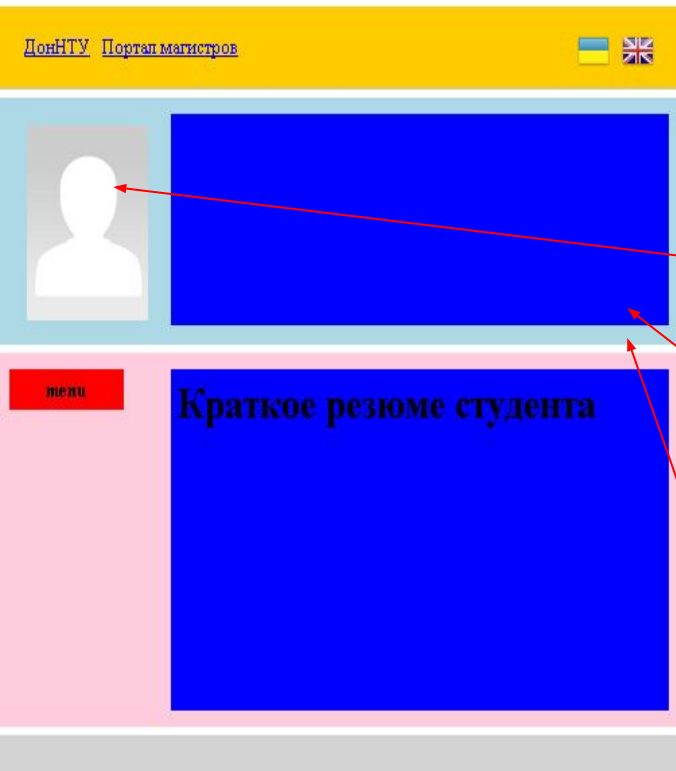
```
<div id="footer">  
<strong>footer</strong>  
</div>
```

```
</body>
```



Блочная верстка (продолжение)

Теперь уберем
границы всех блоков
и установим
обтекание в стилях:

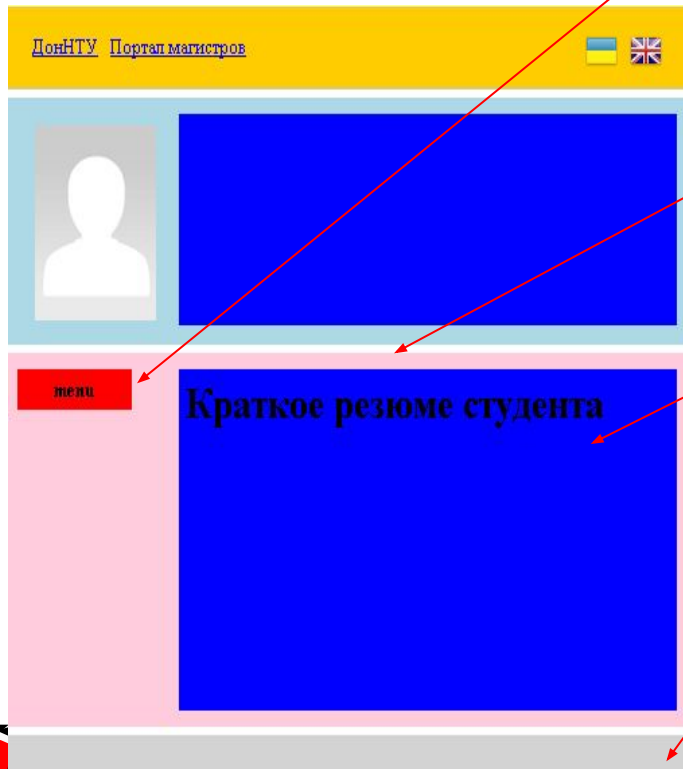
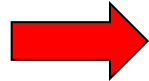


```
#topblock { /* верхний блок */
  min-height: 36px; background:#fc0; width:50%;
  border-bottom: solid 1pt #b4cdd4;
  min-width:500px;
}
#langbox { /* Блок языков */
  float: right; border:0;
  padding: 5pt 8pt 0pt 0pt;
}
#donntu { /* Блок ДонНТУ и портала магистров */
  float: left; border:0;
  padding: 5pt 0pt 0pt 8pt; }
#photo { /* Фото */
  float: left; width: 10%;
  padding: 5pt 10pt 5pt 10pt; }
#headertext { /* Текст заголовка */
  padding: 10pt 20pt 20pt 10pt;
  background:blue;
  height: 90px; /* Высота слоя */
  min-width:330px; float: right; }
#header { /* Блок шапки) */
  min-height: 140px; ackground:lightblue;
  width:50%; min-width:500px;
  clear: both;}
```



Блочная верстка (продолжение)

А так же ограничим
уменьшение блоков
при уменьшении
страницы



```
#nav { /* Блок меню */  
  float: left; background:red;  
  width: 15%; height:300px;  
  padding: 5 pt 0pt 2pt 0pt;  
  text-align: center;  
}  
#middleblock {  
  background:#fcd; width:50%;  
  min-height:220px;  
  min-width:500px;}  
#content {  
  background:blue;  
  float:right;  
  min-width:360px; min-height:200px; }  
#footer{  
  background:lightgray;  
  width:50%;  
  min-width:500px;}  
</style>
```

Чтобы все смотрелось именно так, как хочется, *float* следует указывать для всех слоев

Блочная верстка (Секреты)

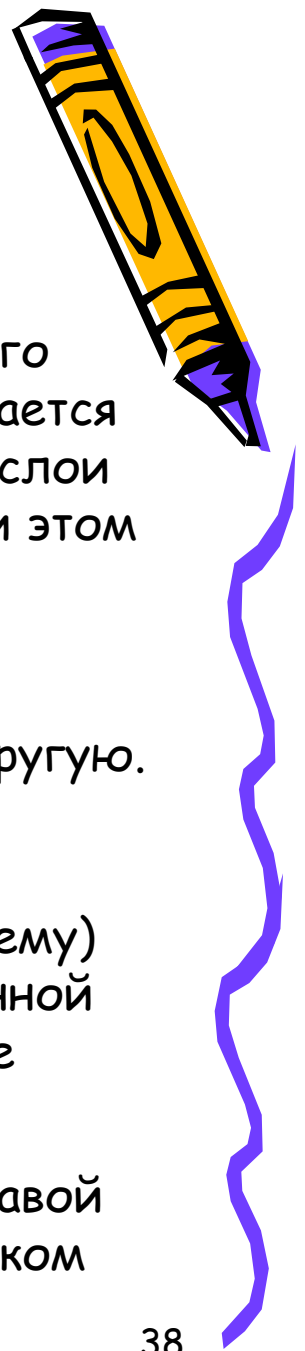
При верстке двухколонного макета очень часто применяется свойство `float`, которое добавляется к каждому слою. Принцип его применения следующий. Вокруг слоя при добавлении `float` создается обтекание, но поскольку ширина каждого слоя жестко задана, то слои располагаются не друг под другом, а рядом по горизонтали. При этом и формируются колонки макета.

Такой способ создания многоколоночного макета имеет и определенный **недостаток**: при уменьшении окна браузера до определенной величины, колонки «перепрыгивают» одна под другую.

Чтобы этого избежать дополнительно используют

а) стилевое свойство `min-width`, применяя его к базовому (нижнему) блоку-контейнеру. Тогда при уменьшении этого блока до указанной ширины, появляются полосы прокрутки, содержимое больше не уменьшается.

б) стилевое свойство `margin-left`, добавляя его к правой колонке и вписывая оба блока в общий третий. В таком случае слои остаются на своих исходных местах, независимо от размеров окна браузера.

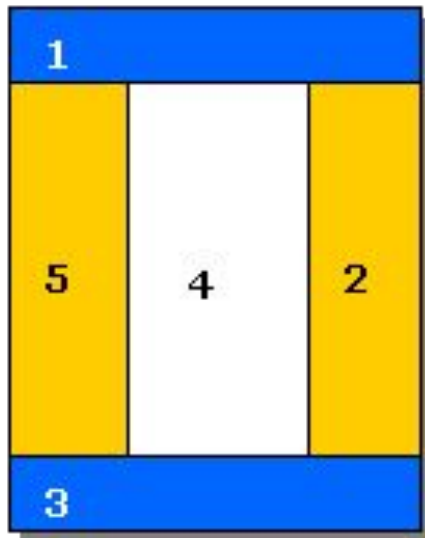


Верстка

План лекции

- Алгоритм верстки
- Виды макетов
- Изображения
- Табличная верстка
- Блочная верстка
- Варианты дизайна персональной страницы**
- Задания для лабораторной

Варианты дизайна персональной страницы

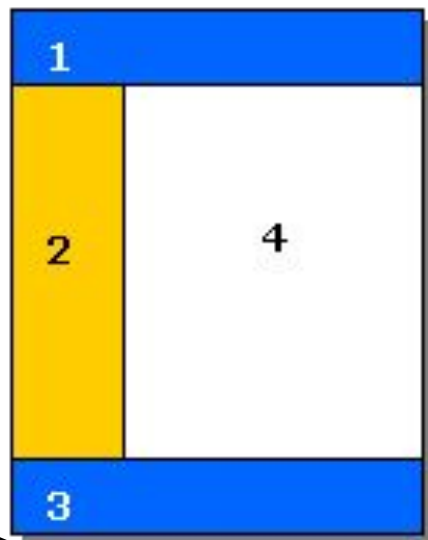


- **1** - верхняя часть страницы (Header).
Используется для размещения логотипа, эмблемы, названия фирмы, страницы и другой наиболее важной информации.
- **2 и 5** - области размещения гиперссылок или панели навигации сайта (Menu).
- **4** - область размещения текста (Content).
- **3** - нижняя часть страницы (Footer).
Используется для размещения контактной информации или иных данных, менее важных, по сравнению с размещенными в области 1 до

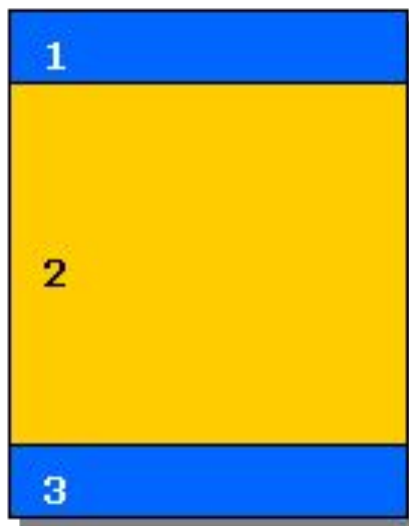


Варианты дизайна персональной страницы

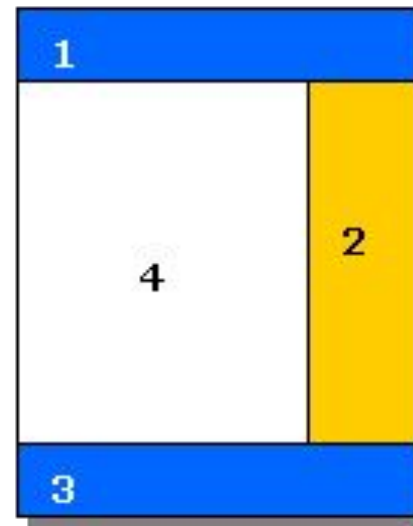
Адаптивный (самонастраивающийся к изменению размеров экрана) дизайн компоновки веб-страниц (1-5).



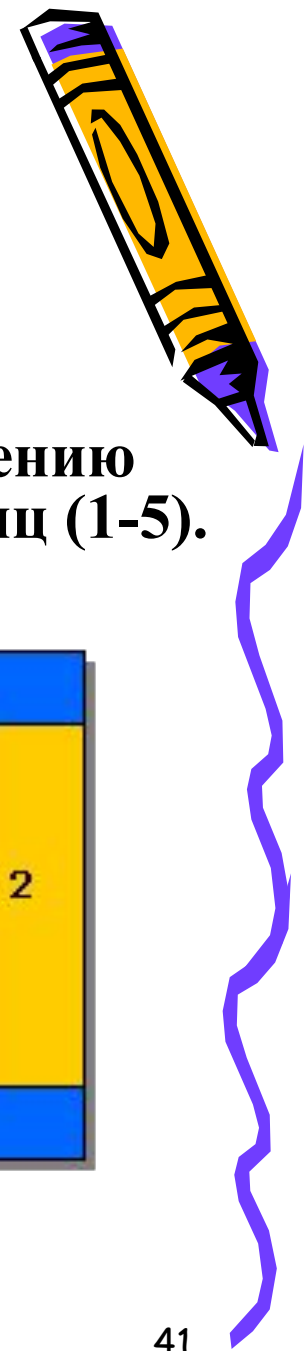
A1



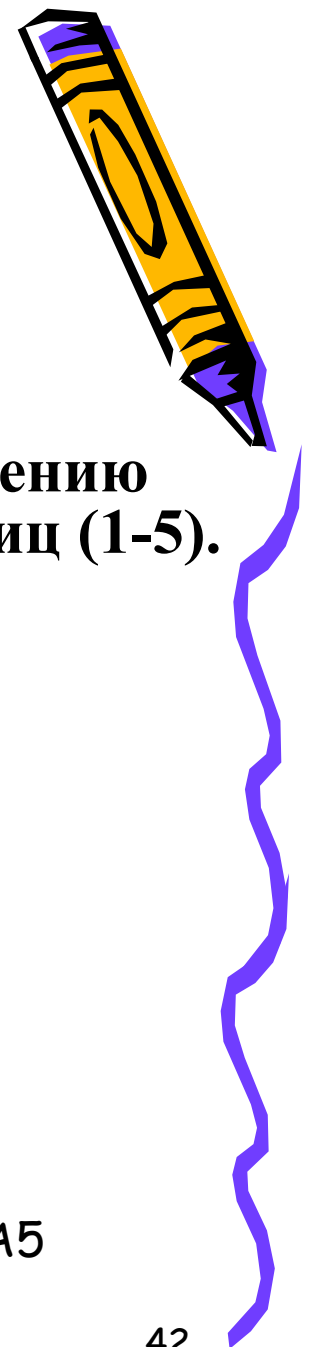
A2



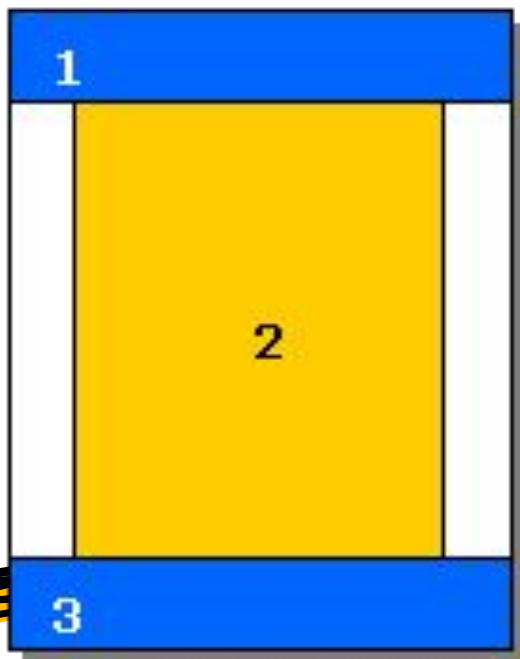
A3



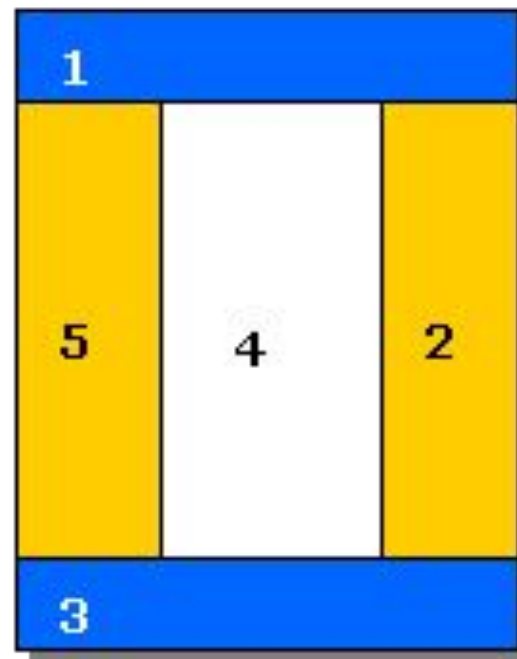
Варианты дизайна персональной страницы



Адаптивный (самонастраивающийся к изменению размеров экрана) дизайн компоновки веб-страниц (1-5).



A4

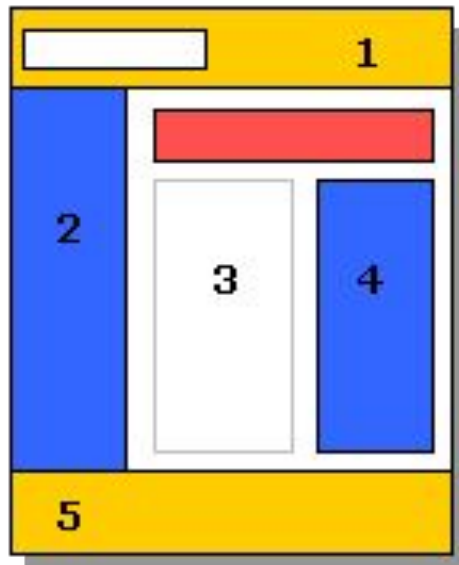


A5





Варианты дизайна персональной страницы

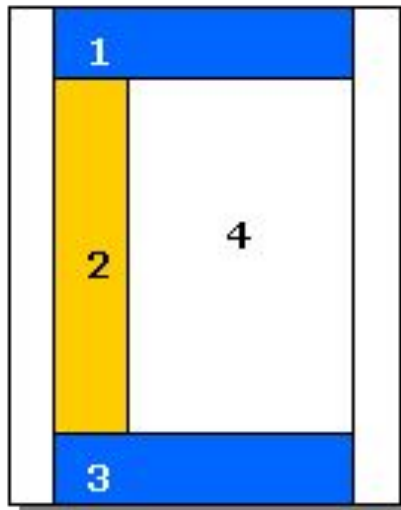
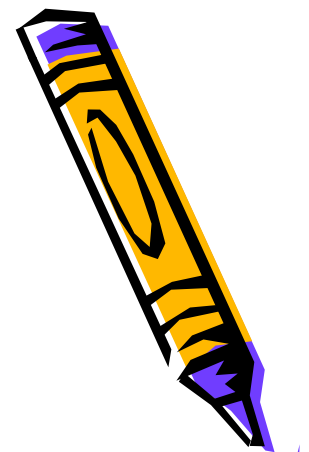


Вариант **жёстко-переменного** (адаптивного к изменению размеров экрана с жёсткими, неизменяющимися областями) дизайна компоновки страниц. Перспективная и сложная компоновка страниц. Используется в современных профессионально выполненных сайтах.

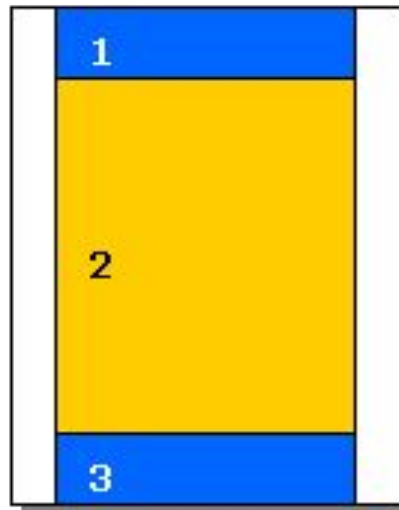
- А6** **Основное преимущество:** дизайн сайта занимает всё пространство экрана. Свободного места не остаётся. **Основной недостаток:** дизайн может искажаться на мониторах с различной разрешающей способностью.



Варианты дизайна персональной страницы



A7



A8

Варианты жёсткой (не адаптивной к изменению размеров экрана) дизайна компоновки страниц.

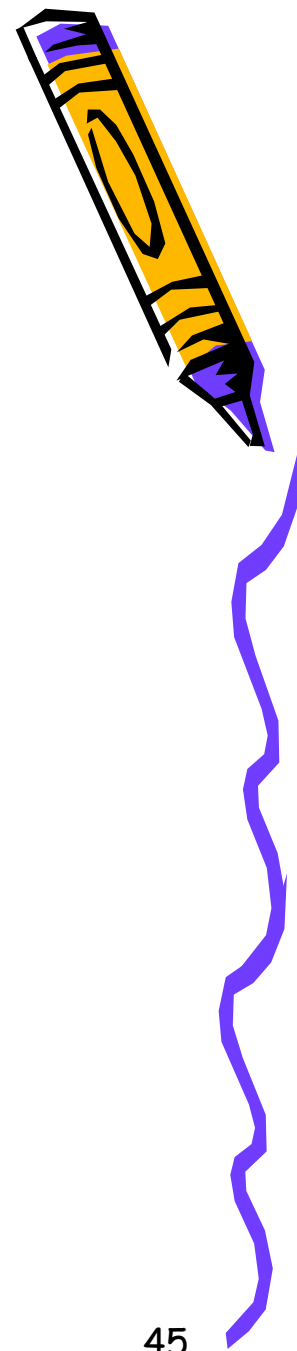
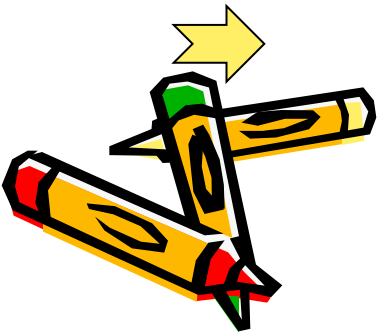
Основное преимущество: дизайн сайта отображается на экранах различной разрешающей способности в том виде, в котором он был задуман веб-дизайнером.

Основной недостаток: часть содержимого может не поместиться на экран или на экране остаётся свободное пространство.





Создание дизайна страницы



Варианты дизайна персональной страницы

Перед тем, как приступить к верстке страницы, нам необходимо сделать для нее дизайн. К сожалению, в рамках данного курса у нас совсем мало времени для разбора даже базовых принципов дизайна. И, хотя для курса по созданию персонального сайта нам необязателен профессиональный дизайн, но верстать красивые страницы все-таки гораздо приятнее.

Далее я привожу некоторую последовательность действий, которую можно использовать для создания несложного, но симпатичного дизайна на базе хорошей фотографии или рисунка.



Варианты дизайна персональной страницы



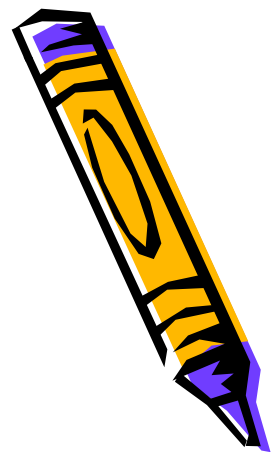
1. Подбор изображения

Для интересного дизайна нам в первую очередь нужна интересная фотография или рисунок. Выбирать изображение нужно исходя из тематики сайта.

Оформим страничку для рассказа О.Генри «Последний лист»

Возьмем рисунок размером примерно 600x450 пикселей

Варианты дизайна персональной страницы



2. Выбор цветовой гаммы

Для выбора цветовой гаммы страницы мы воспользуемся нашей фотографией или картинкой. Нам необходимо выбрать 2-3 базовых цвета, на которых будет строиться вся композиция. При этом постарайтесь выбрать два близких по цвету оттенка и, как минимум, один контрастный к ним. Наиболее удобно это делать в графическом редакторе, используя инструмент "пипетка".

Старайтесь выбирать преобладающие на фотографии или рисунке цвета. Исходя фотографии, получилась следующая цветовая палитра



Варианты дизайна персональной страницы



2. Выбор цветовой гаммы

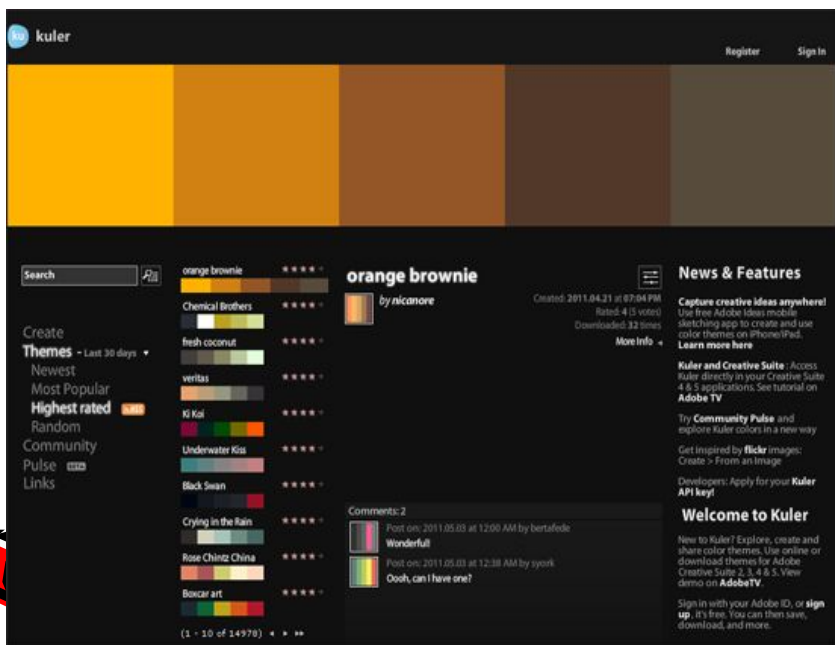
Два наиболее популярных бесплатных средства для подбора палитры сайта:

1. **Kuler:**

<https://kuler.adobe.com/>

2. **Color Palette Generator:**

<http://www.degraeve.com/color-palette/index.php>



Color Palette Generator

Enter the URL of an image to get a color palette that matches the image. This is useful for coming up with a website color palette that matches a key image a client wants to work with.

URL of image:



dull	vibrant

3. **ColourLovers**

Варианты дизайна персональной страницы

3. Выбор шрифтов

- Не увлекайтесь большим количеством шрифтов - вполне достаточно пары гарнитур
- Если используете в заголовке рукописный шрифт, то в пару к нему хорошо будет смотреться шрифт без засечек
- Если шрифт полужирный, то в пару к нему хорошо смотрится шрифт, нарисованный тонкими линиями
- Для набора основного текста используйте кегль в 10-12 пунктов
- Не увлекайтесь смешением различных начертаний шрифта (прямой, наклонный, полужирный)



Варианты дизайна персональной страницы



4. Определение функциональных областей

У нас определены следующие функциональные блоки

header - Заголовок сайта

menu - Меню сайта

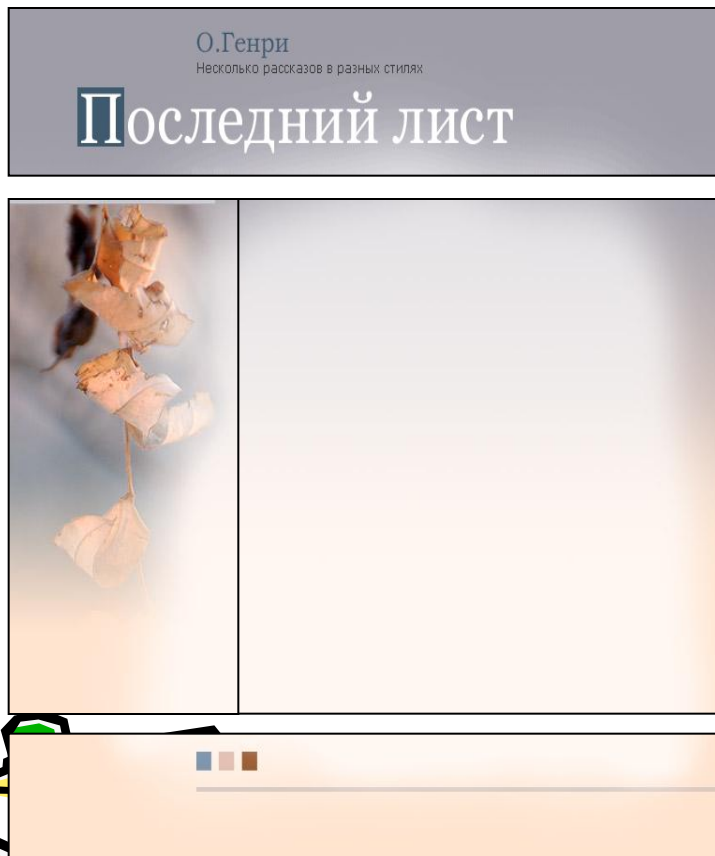
content - Основное содержание

footer - Копирайт

На них и остановимся при создании дизайна страницы.



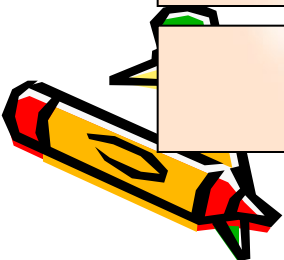
Варианты дизайна персональной страницы



5. Компоновка сайта

Вы можете взять за основу любой из приведенных выше вариантов композиции либо придумать свой собственный.

При выборе композиции обращайте внимание на то, каким образом размещается графика в дизайне страницы. По возможности в готовом дизайне графические фрагменты должны полностью помещаться в тот или иной блок.



О.Генри

Несколько рассказов в разных стилях

Последний лист

Из сборника
"Горящий светильник"

В небольшом квартале к западу от Вашингтон-сквера улицы перепутались и переломались в короткие полоски, именуемые проездами. Эти проезды образуют странные углы и кривые линии. Одна улица там даже пересекает самое себя раза два. Некому художнику удалось открыть весьма ценное свойство этой улицы. Предположим, сборщик из магазина со счетом за краски, бумагу и холст повстречает там самого себя, идущего восвояси, не получив ни одного цента по счету!

И вот люди искусства набрели на своеобразный квартал Гринич-Виллидж в поисках окон, выходящих на север, кровель XVIII столетия, голландских мансард и дешевой квартирной платы. Затем они перевезли туда с Шестой авеню несколько оловянных кружек и одну-две жаровни и основали "колонию".

Студия Сю и Джонси помещалась наверху трехэтажного кирпичного дома. Джонси - уменьшительное от Джозанны. Одна приехала из штата Мэйн, другая из Калифорнии. Они познакомились за столиком одного ресторанчика на Восьмой улице и нашли, что их взгляды на искусство, цинкорный салат и модные рукава вполне совпадают. В результате и возникла общая студия. Это было в мае. В ноябре неприветливый чужак, которого доктора именуют Пневмонией, незримо разгуливал по колонии, касаясь то одного, то другого своими ледяными пальцами. По Восточной стороне этот душегуб шагал смело, поражая десятки жертв, но здесь, в лабиринте узких, поросших мохом переулков, он плелся нога за ногу.

Господина Пневмонию никак нельзя было назвать галантным старым джентльменом. Миниатюрная девушка, малокровная от калифорнийских зефиров, едва ли могла считаться достойным противником для дюжего старого тупицы с красными кулачищами и одышкой. Однако он свалил ее с ног, и Джонси лежала неподвижно на крашеной железной кровати, глядя сквозь мелкий переплет голландского окна на глухую стену соседнего кирпичного дома. Однажды утром озабоченный доктор одним движением косматых седых бровей вызвал Сю в коридор.

- У нее один шанс... ну, скажем, против десяти, - сказал он, стряхивая ртуть в термометре. - И то, если она сама захочет жить. Вся наша фармакопея теряет смысл, когда люди начинают действовать в интересах гробовщика. Ваша маленькая барышня решила, что ей уже не поправиться. О чем она думает?



Создание дизайна страницы



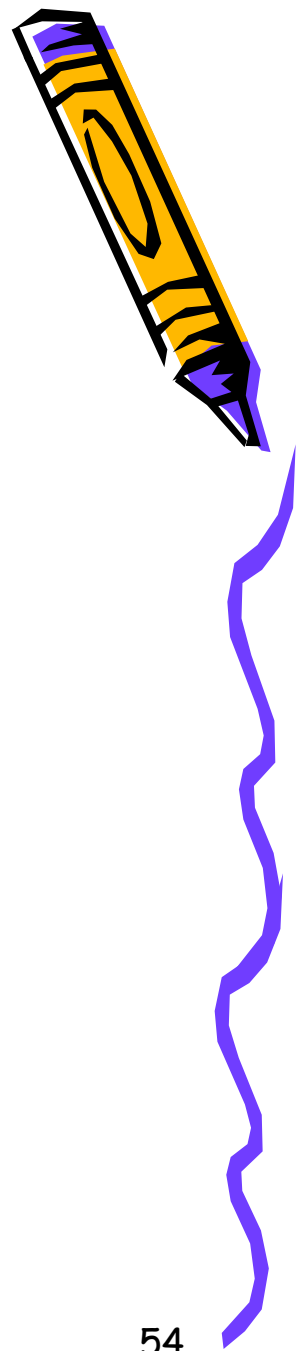
6. Готовая страница

Графические блоки
выполнены в
выбранной цветовой
гамме в редакторе
Photoshop

Разметка выполнена с
помощью таблицы с
тремя строками и
двумя столбцами в
средней строке.

Блочная верстка (Секреты)

Остальные тонкости блочной верти вы узнаете в процессе изучения каскадных таблиц стиля на следующих занятиях



Примеры студенческих работ



Главная Задания ДонНТУ 11:23


Краткое резюме студента

Образование	<ul style="list-style-type: none">○ 2001-2012: ОШ №3 г.Донецка (золотая медаль)○ 2012-2016: Бакалавр, ДонНТУ, Факультет компьютерных наук и технологий, системное программирование Средняя оценка за первый семестр: 5А.
Внешкольное образование, дополнительные курсы	<ul style="list-style-type: none">○ 2006-2012: Музыкальная школа №1, гитара (с отличием)○ 2010-2011: Курсы Macromedia Flash (с отличием)○ 2011-2013: Школа иностранных языков "Sherwood", английский язык (с отличием)

Навигация

Факультет компьютерных наук и технологий

Кафедра компьютерной инженерии



Группа КИ-12:



Факультет компьютерных наук и технологий

Кафедра компьютерной инженерии

Специальность «Системное программирование»



Резюме Биография Анимация CSS JAVA - SCRIPT Индивидуальный проект

Биография студента



Примеры студенческих работ



Факультет компьютерных наук и технологий
Кафедра компьютерной инженерии
Специальность «Компьютерные системы и сети»

Краткое резюме студента

- Резюме
- Биография
- Анимации
- Задания по CSS

Факультет компьютерных наук и технологий
Кафедра компьютерной инженерии
Специальность «Компьютерные системы и сети»

- Резюме
- Биография
- Анимация
- Задания по CSS
- Задания по JavaScript
- Задание КР

ВУЗ	2012-2016 - Донецкий национальный технический университет Факультет компьютерных наук и технологий Кафедра компьютерной инженерии Специальность «Компьютерные системы и сети», бакалаврат
Средний балл	5
Владение языками	русский («в совершенстве»), украинский («в совершенстве»), английский («выше среднего»)



Примеры студенческих работ



ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ НАУК И ТЕХНОЛОГИЙ
КАФЕДРА КОМПЬЮТЕРНОЙ ИНЖЕНЕРИИ
СПЕЦИАЛЬНОСТЬ "СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ"

Резюме Биография Анимация Задание по CSS Задание по JS

Индивидуальный раздел

Фехтование (нем. fechten — «сражаться, бороться») — система

Факультет компьютерных наук и технологий
Кафедра компьютерной инженерии
Специальность «Системное программирование»

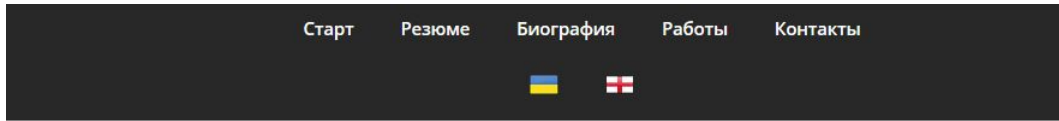
Краткое резюме студента

ФИО студента	
Дата рождения	
Место рождения	

Резюме
Биография
Анимация
Задания по CSS

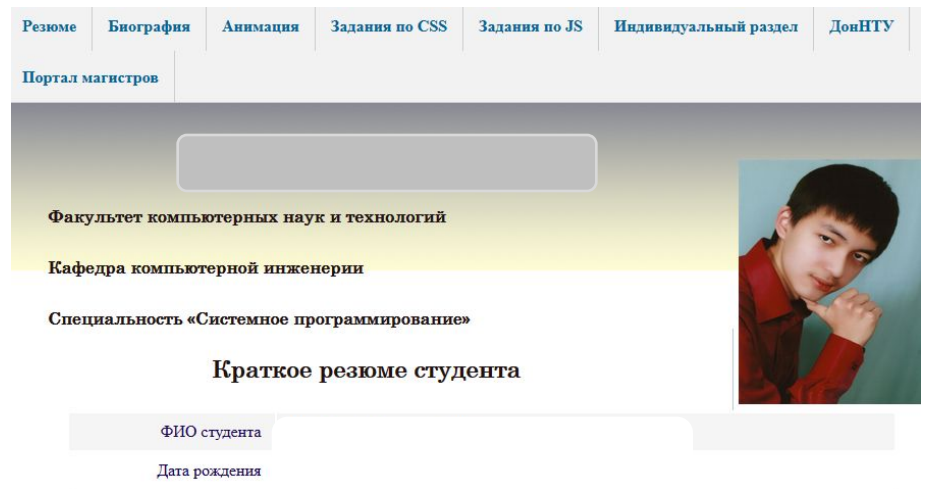


Примеры студенческих работ



Факультет компьютерных наук и технологий
Кафедра компьютерной инженерии
Специальность «Системное программирование»

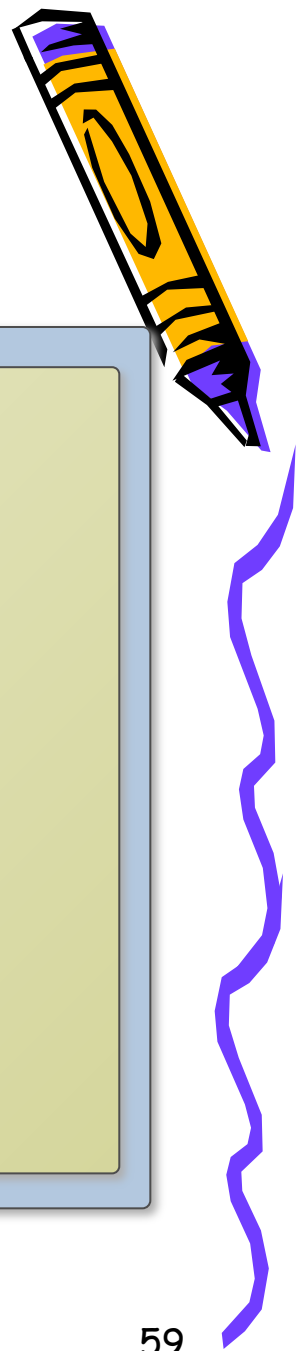
[Просмотр](#)



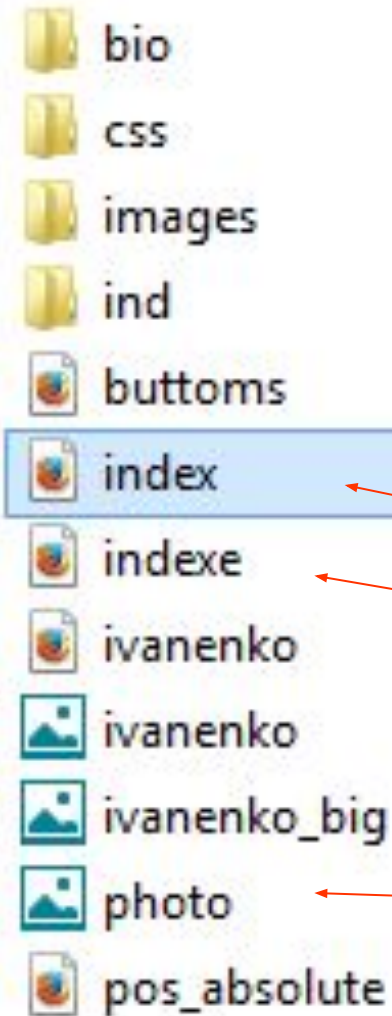
Верстка

План лекции

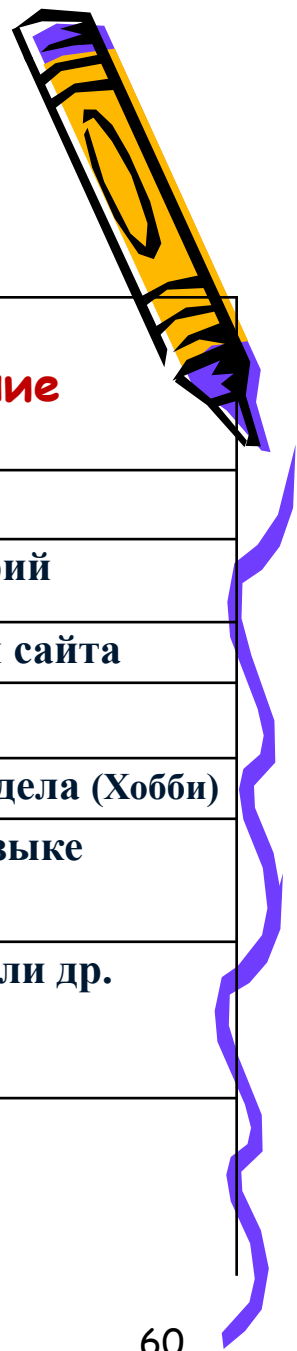
- Алгоритм верстки
- Виды макетов
- Изображения
- Табличная верстка
- Блочная верстка
- Задания для лабораторной



Структура персонального сайта студента



Название каталога	Название файла	Назначение
Корневой каталог (ivanenko)		
bio/	Index(u).html	Каталог для биографий
css/	*.css	Набор таблиц стилей сайта
images/	*.gif, *.png, *.jpg	Каталог рисунков
ind/	index.html	Каталог для инд. раздела (Хобби)
	index.htm	Резюме на русском языке (главная страница)
	indexe.htm	Резюме на англ.яз (Или др. иностр. яз.)
	Ivanenko_big.jpg photo.jpg	Фотографии



Задания для лабораторной

Задание №2

Ваш небольшой сайт придется **преобразовать по правилам блочной верстки.**

В этом Вам поможет выполнение лабораторной работы №6.

Вопросы?

