Web-страницы. Дизайн и графика

Оформление документа

УТВЕРЖДЕНО
Правлением Союза
(Протокол №17 от 19.12.2017 г.)

ОДОБРЕНО
Решением Экспертного совета
при Союзе «Агентство развития
профессиональных сообществ
и рабочих кадров
«Молодые профессионалы
(Ворлдскиллс Россия)»
(Протокол № 43/12 от 15.12.2017 г.)

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ДЕМОНСТРАЦИОННОГО ЭКЗАМЕНА ПО СТАНДАРТАМ ВОРЛДСКИЛЛС РОССИЯ ПО КОМПЕТЕНЦИИ «ВЕБ-ДИЗАЙН И РАЗРАБОТКА»

Перечень знаний, умений, навыков в соответствии со Спецификацией стандарта компетенции «Веб-дизайн и разработка»

1	Организация работы и управление Специалист должен знать и понимать: • Принципы и практики, которые позволяют продуктивно работать в команде; • Аспекты систем, которые позволяют повысить продуктивность и выработать оптимальную стратегию; • Как проявить инициативу и предприимчивость в целях выявления, анализа и оценивания информации из различных источников. Специалист должен уметь: • Решать распространенные задачи веб-дизайна и разработки кода; • Учитывать временные ограничения и сроки; • Производить отладку кода программ и находить ошибки; • Использовать компьютер или устройство и целый ряд программных пакетов; • Применять исследовательские приемы и навыки, чтобы быть в курсе последних отраслевых решений; • Планировать график рабочего дня с учетом требований; • Включать ссылки на изображения, шрифты и др. файлы при архивации данных;

2	Коммуникативные и межличностные навыки Специалист должен знать и понимать: • Как решить проблемы в общении, в том числе выявление проблемы, ее исследование, анализ, решение, макетирование, пользовательское тестирование и оценка результатов; • Принципы, лежащие в основе сбора и представления информации; • Дизайн-концепции и техники, в том числе черновое макетирование страниц (wireframing), объектно- событийное моделирование (storyboarding) и создание блок-схем; • английский язык в рамках чтения и понимания официальной технической документации по используемым технологиями и языкам программирования. Специалист должен уметь: • Представить продукт, который отвечает требованиям клиента и спецификации; • Собирать, анализировать и оценивать информацию; • Использовать навыки грамотности для толкования стандартов и требований; • Планировать и организовывать общение с клиентом; • Критиковать свои проекты и идеи

3	Графический дизайн веб-страниц Специалист должен знать и понимать: • Структуру и общепринятые элементы веб-страниц различных видов и назначений; • Вопросы, связанные с когнитивными, социальными, культурными, технологическими и экономическими условиями при разработке дизайна; • Как создавать и оптимизировать графику для сети Интернет; • Как создавать дизайн по предоставляемым инструкциям и спецификациям; • Какие умения и навыки необходимы для выбора цвета, типографики и композиции; • Принципы и методы адаптации графики для использования ее на веб-сайтах; • Правила поддержания фирменного стиля, бренда и стилевых инструкций; • Ограничения, которые накладывают мобильные устройства и разрешения экранов при использовании их для просмотра веб-сайтов; • Принципы построения эстетичного и креативного дизайна; • Современные стили и тенденции дизайна.

	Специалист должен уметь: • Создавать и анализировать разработанные визуальные ответы на поставленные вопросы, в том числе об иерархии, типографики, эстетики и композиции; • Создавать, использовать и оптимизировать изображения для веб-сайтов; • Анализировать целевой рынок и продукцию, которую продвигает, используя дизайн; • Выбирать дизайнерское решение, которое будет наиболее подходящим для целевого рынка; • Принимать во внимание влияние каждого элемента, который добавляется в проект во время разработки дизайна; • Использовать все требуемые элементы при разработке дизайна; • Учитывать существующие правила корпоративного стиля; • Создавать «отзывчивый» дизайн, который будет отображаться корректно на различных устройствах и при разных разрешениях; • Придерживаться оригинальной концепции дизайна проекта и улучшать его визуальную привлекательность; • Превращать идею в эстетичный и креативный дизайн.

4	Верстка страниц Специалист должен знать и понимать: • Методы обеспечения доступа к страницам веб-сайтов аудитории с ограниченными возможностями; • World Wide Web Consortium (W3C) стандарты HTML и CSS; • Методы верстки веб-сайтов и их стандартную структуру; • Web accessibility initiative (WAI); • Как применять соответствующие CSS правила и селекторы для получения ожидаемого результата; • Лучшие практики для Search Engine Optimization (SEO) и интернет-маркетинга; • Как встраивать и интегрировать анимацию, аудио, видео и другую мультимедийную информацию, управлять поведением остальных элементов на странице. Специалист должен уметь: • Создавать html-страницы сайта на основе предоставленных графических макетов их дизайна; • Корректно использовать CSS для обеспечения единого дизайна в разных браузерах; • Создавать адаптивные веб-страницы, которые способны оставаться функциональными на различных устройствах при разных разрешениях; • Создавать веб-сайты полностью соответствующие текущим стандартам W3C (http://www.w3.org);

Лендинг – это небольшой одностраничный сайт, полностью заточенный под то, чтобы попадающий на него клиент выполнил целевое действие.
Как правило, лендинг состоит из одной страницы, на которой размещена вся необходимая пользователю информация о товаре или услуге.
Основная цель такой страницы – привлекать внимание клиента, мотивировать его сделать заказ. Она может предлагать пользователю товар или услугу, а может агитировать за подписку, к примеру за рассылку. Все зависит от того, на какое целевое действие вы ориентируетесь.

https://www.activetraffic.ru/wiki/landing-page/

Особенности Web-графики

При сохранении иллюстрации, подготовленной в графическом редакторе, всегда возникают два вопроса:

- ✔ как не потерять качество изображения;
- ✔ как получить файл небольшого объёма.

Эти вопросы особенно актуальны при подготовке графики для Web, ведь посещаемость сайта напрямую зависит от времени загрузки его страниц. Очень хочется сохранить хорошее качество иллюстраций, но “тяжёлая” графика испытывает терпение пользователя и ударяет по его кошельку.

.

Посещаемость сайта (при прочих равных условиях) тем выше, чем быстрее он просматривается. Пользователь любит быстрое обслуживание и не хочет тратить деньги на просмотр картинок “для настроения”.

Что понимается под объёмом страницы? Конечно, не геометрический размер страницы на экране, а сумма килобайтов, которые нужно загрузить по сети для построения экранного образа. Это размер самого HTML-файла, размер картинок, размер файлов со скриптами и стилевыми таблицами.

Основной источник лишних килобайт — картинки, и к ним нужно относиться с особым вниманием. Отказаться от лишних декораций, шершавых фонов, анимаций, уменьшить геометрические размеры иллюстраций, оптимизировать графику.

Под оптимизацией графики понимается поиск разумного соотношения “килобайтный объём/качество” (небольшой объём при приемлемом качестве изображения!!!!!).

Правила подбора цветовой схемы

- Ключевой цвет.
- Дополнительные цвета.
- Фон.

Ключевой цвет

Наличие ключевого цвета определяет стиль и эмоции.

Как правило, его оттенок близок к трем основным: красный, зеленый, синий.

Красный: тепло, сила, стабильность, комфорт, шик.

Зеленый: радость, надежность.

Синий: спокойствие, сила, простота.

1.


2.


3.


1.

2.

1.

2.

Дополнительные цвета, в основном создают фон.

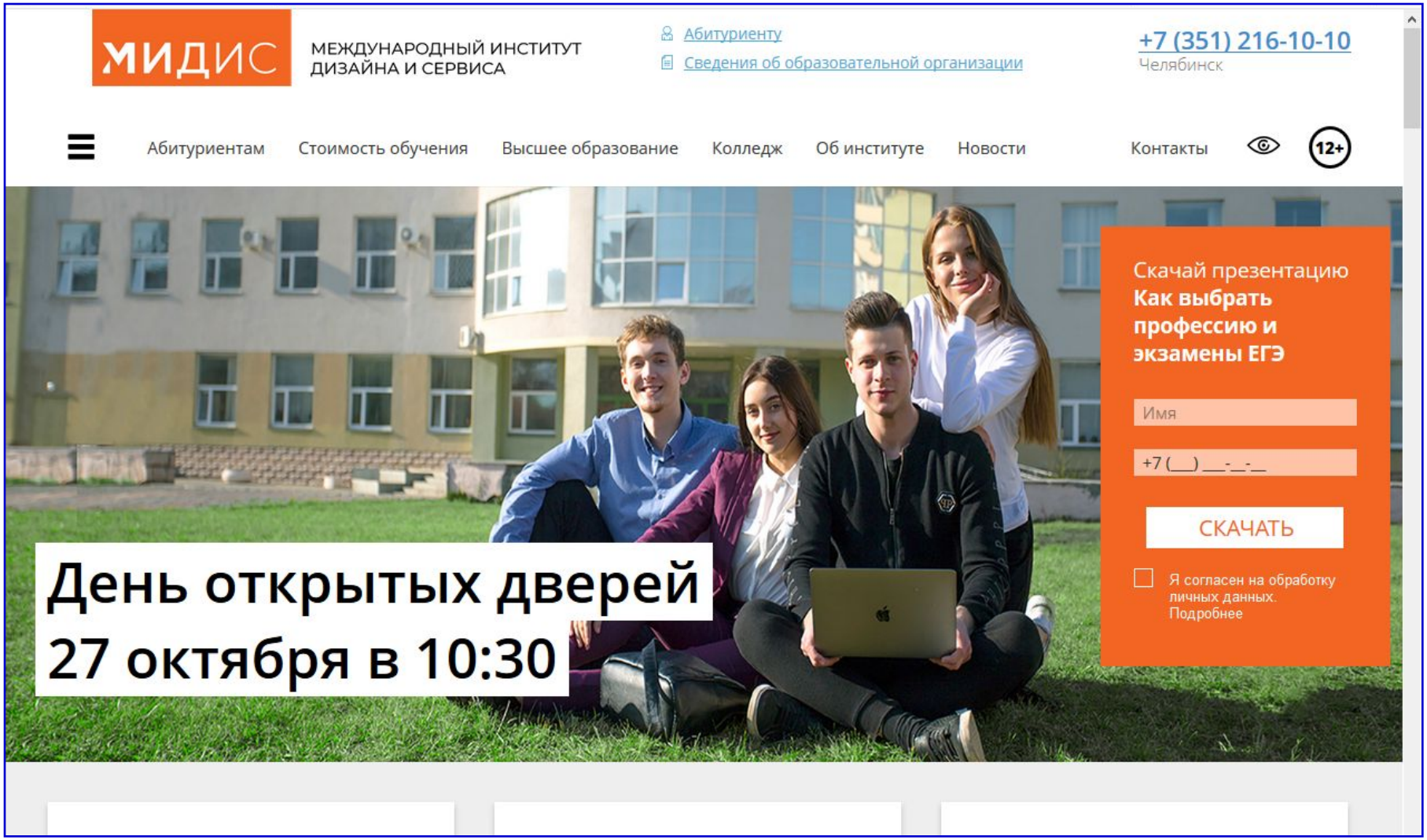
- Желательно - основной фон не белый (не напрягает яркостью).
- Если фоном выбран рисунок – приглушенная яркость.
- Фоном могут быть светлые и темные оттенки друг друга и контрастные ключевому цвету.

Образец ужасного подбора фонового рисунка.

Старайтесь использовать одну и ту же картинку на разных страницах.

Во-первых, такое украшение обойдётся дешевле: картинка будет загружаться по сети только для первой страницы, а для других — из кеша компьютера.

Во-вторых, пользователь, увидев одну и ту же картинку на всех страницах, получит ощущение *фирменности* дизайна, связанности страниц друг с другом. Пример многократного использования картинок — кнопки навигационной панели. Уместно повторять на каждой странице логотип сайта. При этом на внутренних страницах его нужно делать ссылкой на главную. Инвариантом страниц является и рисованный заголовок сайта.

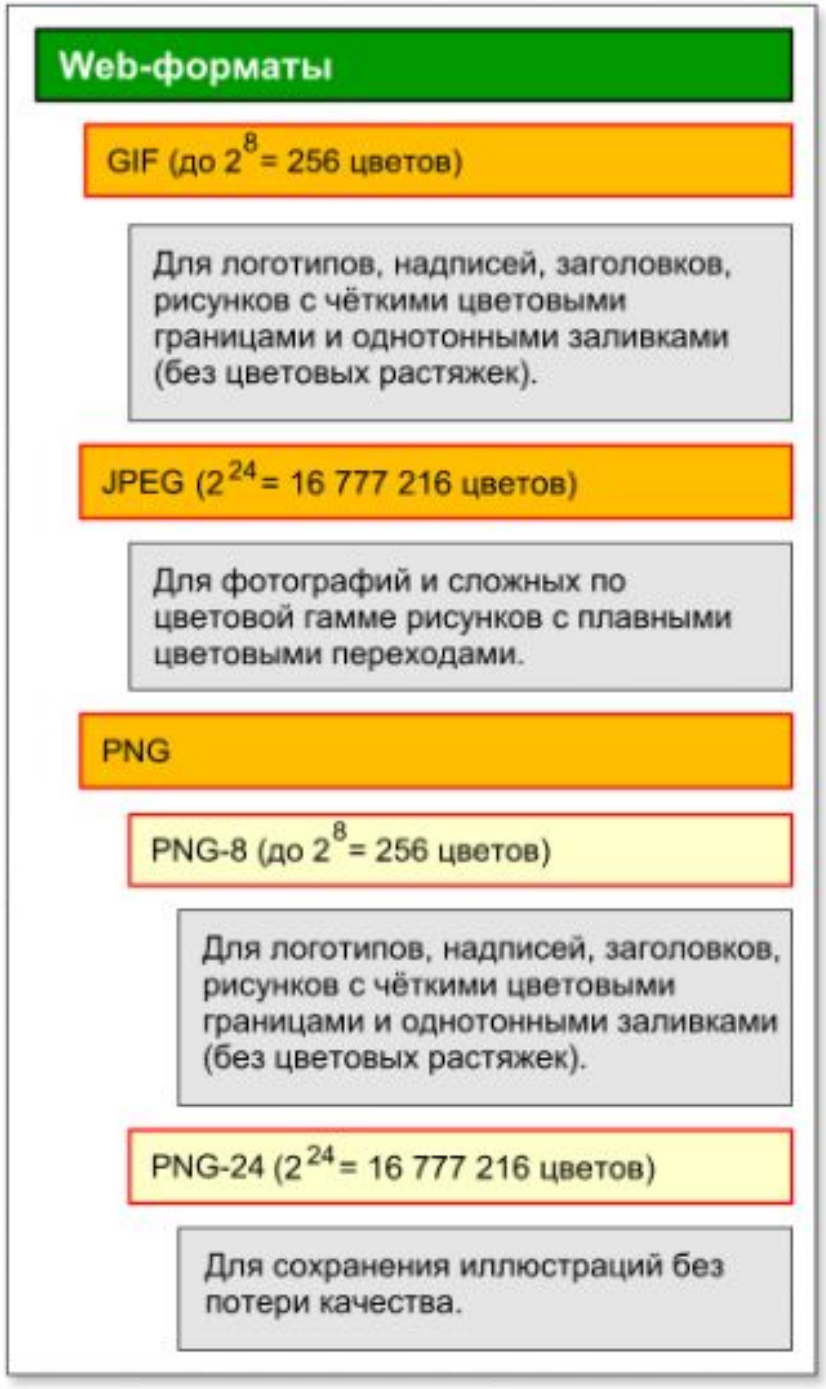


https://midis.info/

http://kremlin.ru/

Один из главных путей оптимизации Web-графики ведёт к выбору графического формата, наиболее подходящего для данной иллюстрации и настройке его параметров при сохранении графического файла.

В Web популярны три графических формата — GIF (файлы с расширением gif), JPEG (файлы с расширением jpg или jpeg) и PNG (файлы с расширением png).



Обычное применение этих графических форматов:

GIF — для логотипов, надписей, заголовков, рисунков с чёткими цветовыми границами и однотонными заливками (без цветовых растяжек);

JPEG — для фотографий и сложных по цветовой гамме рисунков с плавными цветовыми переходами;

PNG — для любых иллюстраций, когда соотношение объём/качество оказывается лучше по сравнению с форматами GIF и JPEG.

Пример:

Дизайн картинок

Рамка

```
<IMG src="pic/1289.gif" border=1
     width=140 height=101
     alt="Задумчивый портрет">
```

В коде картинки, приведённой выше, записано border=1. С толщиной рамки главное не переборщить, а то она будет навевать грустные мысли. Ниже приводится та же картинка, но для неё записано border=10. Выглядит мрачновато:

```
<IMG src="pic/1289.gif" border=10
     width=140 height=101
     alt="Траурный портрет">
```

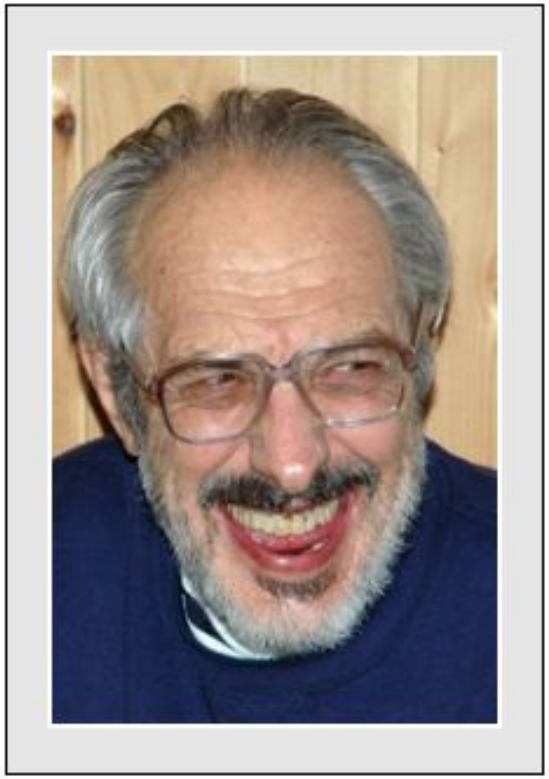
```
<IMG src="pic/1289.gif"
     style="border:2px solid olive"
     width=140 height=101
     alt="Задумчивый портрет">
```

Хорошую рамку можно построить, если вместо значения solid (сплошная) использовать значение ridge (рамка со складкой):

```
<IMG src="pic/1289.gif"
     style="border:5px ridge #ccbbaa"
     width=140 height=101
     alt="Задумчивый портрет">
```

Зелёная рамка со “складкой” хорошо подходит для оформления пейзажа:

```
<IMG src="pic/1290.jpg"
     style="border:5px ridge lime"
     width=200 height=143
     alt="Озеро Плещеево">
```



Разновидностью рамки является *паспарту* — картонная рамка. Паспарту можно смоделировать следующими стилевыми определениями:

```
.frame
{
  padding-top: 20px;
  padding-bottom: 20px;
  width:240px;
  background:#e5e5e5;
  border:1px solid black;
  text-align:center;
}
.frame IMG
{
  border:2px solid white;
}
```

Задание 1.

Изготовить для иллюстраций рамки, используя средства HTML и CSS (без графического редактора).

Обтекание

Обтекание картинки другими элементами можно задавать при помощи атрибута align со значениями:

- left — картинка выравнивается по левому краю, остальные элементы обтекают справа;
- right — картинка выравнивается по правому краю, остальные элементы обтекают слева.

Обтекание прекращает элемент **BR** с атрибутом clear:

- <**BR** clear=left> — прекращение обтекания, заданного как align=left;
- <**BR** clear=right> — прекращение обтекания, заданного как align=right;
- <**BR** clear=all> — прекращение любого обтекания.

Обтекание можно задать и при помощи стилевого свойства float (плавающий элемент):

- float:left — элемент выравнивается по левому краю, остальные элементы обтекают справа;
- float:right — элемент выравнивается по правому краю, остальные элементы обтекают слева.

Обтекание прекращает элемент, для которого задано стилевое свойство clear:

- clear:left — прекращение обтекания, заданного как float:left;
- clear:right — прекращение обтекания, заданного как float:right;
- clear:both — прекращение любого обтекания.

Примеры:

Обтекание справа:

Текст. Текст. Текст. Текст.
Текст. Текст. Текст. Текст.
Текст. Текст. Текст. Текст.
Текст. Текст. Текст. Текст.
Текст. Текст. Текст. Текст.
Текст. Текст. Текст. Текст.
Текст. Текст. Текст. Текст.
Текст. Текст. Текст. Текст. Текст. Текст.
Текст. Текст.

float:left
или
align=left

Обтекание слева:

Текст. Текст. Текст. Текст.
Текст. Текст. Текст. Текст.
Текст. Текст. Текст. Текст.
Текст. Текст. Текст. Текст.
Текст. Текст. Текст. Текст.
Текст. Текст. Текст. Текст.
Текст. Текст. Текст. Текст.
Текст. Текст. Текст. Текст. Текст. Текст.
Текст. Текст.

float:right
или
align=right

Обтекание слева и справа:

Текст. Текст. Текст. Текст. Текст. Текст.
Текст. Текст. Текст. Текст. Текст. Текст.
Текст. Текст. Текст. Текст. Текст. Текст.
Текст. Текст. Текст. Текст. Текст. Текст.
Текст. Текст. Текст. Текст. Текст. Текст.
Текст. Текст. Текст. Текст. Текст. Текст.
Текст. Текст. Текст. Текст. Текст. Текст. Текст. Текст. Текст.

```
.left
{
  float:left;
  margin-right:10px;
  margin-bottom:5px;
}
.right
{
  float:right;
  margin-left:10px;
  margin-bottom:5px;
}
```

```
<IMG src="pic/1294.jpg" class=left
     width=79 height=104 alt="Светлана">
<IMG src="pic/1295.jpg" class=right
     width=79 height=104 alt="Светлана">
<P align=justify>
Текст. Текст. Текст. Текст. Текст. Текст. Текст.
Текст. Текст. Текст. Текст. Текст. Текст. Текст.
Текст. Текст. Текст. Текст. Текст. Текст. Текст.
Текст. Текст. Текст. Текст. Текст. Текст. Текст.
Текст. Текст. Текст. Текст. Текст. Текст. Текст.
Текст. Текст. Текст. Текст. Текст. Текст. Текст.
Текст. Текст. Текст.
```

Задание 2.

Создать страницу с описанием картинки по образцу.

Животный мир

Лиза

Текст. Текст. Текст. Текст. Текст. Текст. Текст. Текст. Текст. Текст. Текст.

Текст. Текст.

Вася

Текст. Текст. Текст. Текст. Текст. Текст. Текст. Текст. Текст. Текст. Текст.

Текст. Текст.

Конструирование

При подготовке иллюстрации для Web-страницы её часто разрезают на несколько частей.

Уменьшение килобайтного объёма иллюстрации. Небольшие фрагменты легче оптимизировать. Однотонные участки, участки с чёткими границами, текстовые надписи можно сохранять в формате GIF (или PNG-8), минимизируя число цветов в палитре отдельно для каждого фрагмента. При сохранении фрагментов в формате JPEG можно управлять параметром качество/объём отдельно для каждой части. В итоге килобайтный объём оптимизированных кусочков оказывается меньше объема полной оптимизированной картинки.

Возможность создания карты ссылок. Отдельные фрагменты общей иллюстрации можно оформить как гипертекстовые ссылки, что особенно полезно, если изображение используется как навигационное меню.

Кажущаяся быстрота загрузки. Иллюстрация появляется на экране по мере поступления её частей по сети, и пользователю есть чем занять себя, в ожидании полной загрузки страницы.

Как из отдельных частей собрать единое изображение? Запустим двух козочек попастись, поместив их в абзац:

Вот код этого абзаца:

```
<P style="white-space:nowrap">
  <IMG src="pic/12100.jpg" width=80 height=70
       hspace=0 vspace=0
       alt="Коза" border=0 align=top>
  <IMG src="pic/12100.jpg" width=80 height=70
       hspace=0 vspace=0
       alt="Коза" border=0 align=top>
</P>
```

Свойство white-space со значением nowrap заставляет браузер выводить элементы в одну строку. При конструировании картинки из фрагментов это необходимо, иначе в узких окнах изображение “развалится”.

Как удалить промежуток между картинками?

```
<P style="white-space:nowrap">
  <IMG src="pic/12100.jpg" width=80 height=70
       hspace=0 vspace=0
       alt="Коза" border=0 align=top
  ><IMG src="pic/12100.jpg" width=80 height=70
       hspace=0 vspace=0
       alt="Коза" border=0 align=top>
</P>
```

Получилось:

сомкнули границы тегов, ведь конец строки для браузера равнозначен пробелу!!!!!

Как бы мы ни уменьшали окно, браузер включит протяжку, но не развалит рисунок на части.

Как построить заголовочную часть сайта из картинки?

Эту картинку удобно разрезать на три части: логотип, заголовок и декоративную иллюстрацию:

Для показа сложно-разрезанных картинок используют таблицы

Оптимизация частей привела к следующему результату:

Фрагмент	Формат	Размер
Логотип	JPEG	8619 КБ
Заголовок	GIF	1838 КБ
Декорация	JPEG	3858 КБ
Итого		14315 КБ

Исходная картинка после оптимизации имела размер 23 КБ. Экономия составила почти 9 КБ (38%).

```
<P style="white-space:nowrap">
  <A href="index.htm"><IMG src="pic/1297.jpg"
       width=209 height=181 border=0
       hspace=0 vspace=0 alt="Логотип"
       title="Переход на главную"
  ></A><IMG src="pic/1298.gif" width=176 height=181
       border=0 hspace=0 vspace=0 alt="Цифровое фото"
  ><IMG src="pic/1299.jpg" width=116 height=181
       border=0 hspace=0 vspace=0 alt="">
</P>
```

Задание 3.

Создать страницу с описанием картинки по образцу.

Первый фрагмент (логотип) оформлен как ссылка на главную страницу.

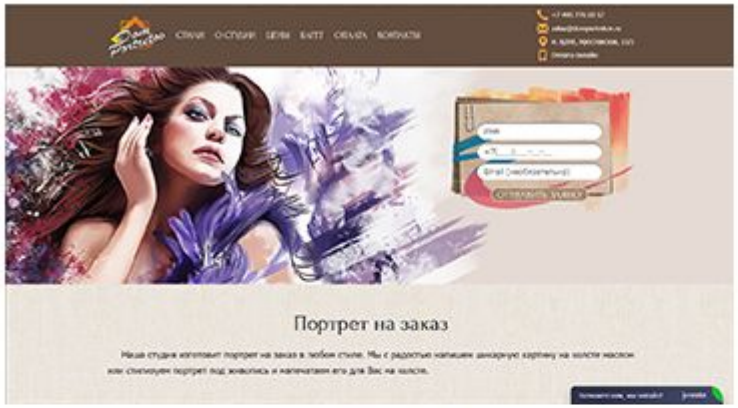
Адаптивная верстка

В настоящее время посетители заходят на сайты не только с настольного компьютера, но и с ноутбука, планшета, смартфона. При этом на устройствах с маленьким экраном часто текст слишком мелкий, ссылки практически некликабельны, элементы управления расположены слишком близко друг от друга.

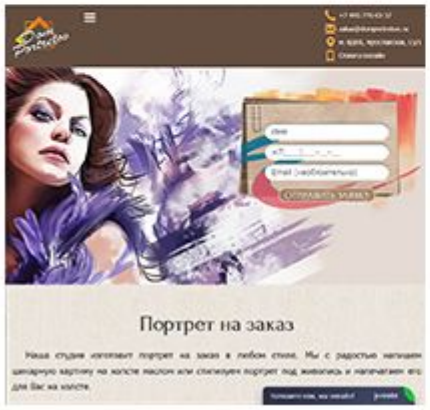
Чтобы устранить эти недостатки стали разрабатывать отдельные мобильные версии сайтов. Но это долго, дорого и неудобно в поддержке.

Решение возникшей проблемы – **адаптивная верстка**, при которой CSS-стили меняются динамически для разной ширины окна браузера.

Пример



Разработку адаптивной верстки чаще всего проводят от уже существующего сайта для большого экрана к экранам меньшего размера. Проверять результат работы можно разными способами:



- просто менять размер экрана браузера,
- расположить справа панель инспектора компонентов и менять ее ширину,
- использовать адаптивный дизайн браузера, нажав **Ctrl+Shift+M** в Firefox или Google Chrome.

Что используют для оптимизации сайта под мобильные устройства?

Метатег viewport

Мобильные браузеры по умолчанию принимают страницу сайта за страницу для обычного компьютера и масштабирует ее по ширине экрана телефона. В результате текст становится мелким, и посетителю, чтобы его прочесть, приходится увеличивать масштаб страницы.

Для корректировки размеров и масштабирования страницы с учетом ширины экрана устройства используют **метатег viewport**, который содержит инструкции для браузера.

Чтобы сообщить браузеру, что страница адаптируется к любым устройствам, в заголовок документа добавляют метатег viewport.

Пример

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Основные свойства метатега <viewport>

width	Ширина видимой области. Рекомендуемое значение: device-width.
height	Высота видимой области. Рекомендуемое значение: device-height.
initial-scale	Первоначальный масштаб страницы. Принимает значение от 1 до 5. Рекомендуемое значение: 1.
minimum-scale	Минимальный масштаб страницы. Принимает значение, которое должно быть меньше или равным initial-scale. Значение 1 запрещает уменьшение масштаба страницы.
maximum-scale	Максимальный масштаб страницы. Принимает значение, которое должно быть больше или равным initial-scale. Значение 1 запрещает увеличение масштаба страницы.
user-scalable	Разрешает или запрещает возможность масштабирования страницы. Принимает значение true или false.

Контент шире экрана – часто возникающая проблема, как только задан viewport.

Это происходит, если каким-то элементам задана большая фиксированная ширина.

Чтобы не появлялась горизонтальная прокрутка, ширину страницы задают на весь экран, при необходимости ограничивая ее свойством max-width.

Пример

```
.content { width: 100%; max-width: 1200px; }
```