

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Усынин Максим Валерьевич
Должность: Ректор
Дата подписания: 07.06.2022 19:37:47
Уникальный программный ключ:
f498e59e83f65dd7c3ce7bb8a25cbbabb33ebc58

Частное образовательное учреждение высшего образования
«Международный Институт Дизайна и Сервиса»
(ЧОУВО МИДиС)
Кафедра сервиса и туризма

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ДЛЯ ВЫПОЛНЕНИЯ ВНЕАУДИТОРНОЙ САМОСТОЯТЕЛЬНОЙ
РАБОТЫ СТУДЕНТОВ ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ
ПМ.08 РАЗРАБОТКА ДИЗАЙНА ВЕБ-ПРИЛОЖЕНИЙ

Специальность:
09.02.07 Информационные системы и программирование

Уровень базового образования обучающихся:
Среднее общее образование

Вид подготовки:
Базовый

Челябинск 2022

Методические указания для выполнения внеаудиторной самостоятельной работы студентов по профессиональному модулю ПМ.08 .Разработка дизайна веб-приложений разработаны на основе федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.07 Информационные системы и программирование, утверждено Приказом Министерства образования и науки РФ от 9 декабря 2016 года № 1547 и рабочей программы профессионального модуля ПМ.08 .Разработка дизайна веб-приложений .

Автор-составитель: Лapidус Л.М., Статных А.С., Левченко А.А.

Методические указания для выполнения внеаудиторной самостоятельной работы студентов по профессиональному модулю ПМ.08 .Разработка дизайна веб-приложений рассмотрены и одобрены на заседании кафедры сервиса и туризма, протокол № 10 от 30.05.2022 г.

Заведующий кафедрой математики и информатики

Л.Ю. Овсяницкая

СОДЕРЖАНИЕ

1.Пояснительная записка.....	4
2.Задания для выполнения внеаудиторной самостоятельной работы.....	7
3.Рекомендуемая литература.....	31

1. Пояснительная записка

Методические указания для выполнения внеаудиторной самостоятельной работы студентов составлены в соответствии с рабочей программой профессионального модуля ПМ.08 .Разработка дизайна веб-приложений, с учетом потребностей клиента и предназначены для реализации основной профессиональной образовательной программы среднего профессионального образования (далее – образовательной программы) по специальности 09.02.07 Информационные системы и программирование.

В методических указаниях для выполнения внеаудиторной самостоятельной работы студентов изложены рекомендации к темам, вынесенным на самостоятельное изучение, приведены задания и виды внеаудиторной самостоятельной деятельности обучающихся, перечень литературы, рекомендуемой для выполнения заданий.

Внеаудиторная самостоятельная работа является одним из видов учебных деятельности обучающихся, который предполагает выполнение заданий преподавателя, но без его непосредственного участия.

Организация внеаудиторной самостоятельной работы по профессиональному модулю ПМ.08 .Разработка дизайна веб-приложений направлена на формирование общих (ОК) и профессиональных (ПК) компетенций:

Общие компетенции (ОК):

Код	Содержание формируемых компетенций
ОК 01.	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам
ОК 02.	Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности
ОК 03.	Планировать и реализовывать собственное профессиональное и личностное развитие
ОК 04.	Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами
ОК 05.	Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста
ОК 06.	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарты антикоррупционного поведения.
ОК 07.	Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях
ОК 08.	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности
ОК 09.	Использовать информационные технологии в профессиональной деятельности
ОК 10.	Пользоваться профессиональной документацией на государственном и иностранном языках
ОК 11.	Использовать знания по финансовой грамотности, планировать предпринимательскую деятельность в профессиональной сфере

Профессиональные компетенции (ПК):

Код	Наименование видов деятельности и профессиональных компетенций
ВД 8	Разработка дизайна веб-приложений
ПК 8.1	Разрабатывать дизайн-концепции веб-приложений в соответствии с корпоративным стилем заказчика
ПК 8.2	Формировать требования к дизайну веб-приложений на основе анализа предметной области и целевой аудитории.
ПК 8.3	Осуществлять разработку дизайна веб-приложения с учетом современных тенденций в области веб-разработки

В результате освоения профессионального модуля ПМ.08 .Разработка дизайна веб-приложений обучающийся должен:

Иметь практический опыт	В разработке дизайна веб-приложений в соответствии со стандартами и требованиями заказчика; создании, использовании и оптимизировании изображений для веб-приложений; разработке интерфейса пользователя для веб-приложений с использованием современных стандартов
Уметь	создавать, использовать и оптимизировать изображения для веб-приложений; выбирать наиболее подходящее для целевого рынка дизайнерское решение; создавать дизайн с применением промежуточных эскизов, требований к эргономике и технической эстетике; разрабатывать интерфейс пользователя для веб-приложений с использованием современных стандартов
Знать	нормы и правила выбора стилистических решений; современные методики разработки графического интерфейса; требования и нормы подготовки и использования изображений в информационно-телекоммуникационной сети "Интернет" (далее - сеть Интернет); государственные стандарты и требования к разработке дизайна веб-приложений

Содержание, объем и виды внеаудиторной самостоятельной работы студентов определяются рабочей программой профессионального модуля ПМ.08 .Разработка дизайна веб-приложений .

Целями внеаудиторной самостоятельной работы студентов являются:

- систематизация и закрепления полученных теоретических знаний;
- углубление и расширение теоретических знаний;
- развитие познавательных способностей и активности студентов, самостоятельности, ответственности и организованности;
- формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации.

Внеаудиторная самостоятельная работа студентов представлена следующими этапами:

- определением цели внеаудиторной самостоятельной работы;
- конкретизацией познавательной (проблемного или практического) задания;
- самооценкой готовности к самостоятельной работе;
- планированием работы (самостоятельной или с помощью преподавателя) над заданием;
- самостоятельным выполнением задания и самоконтролем промежуточного и конечного результатов с целью дальнейшей корректировки выполнения задания.

Формы и виды внеаудиторной самостоятельной работы студентов определяются содержанием профессионального модуля ПМ.08 .Разработка дизайна веб-приложений .

Формы и методы контроля внеаудиторной самостоятельной работы студентов: экзамен, дифференцированный зачет.

Критерии оценки результатов внеаудиторной самостоятельной работы студентов:

- «отлично», если работа выполнена в полном объеме с соблюдением необходимой последовательности. Обучающиеся работают полностью самостоятельно: подбирают необходимые для выполнения предлагаемых работ источники знаний, показывают необходимые для проведения практической работы теоретические знания, практические умения и навыки.

– «хорошо», если работа выполняется обучающимися в полном объеме и самостоятельно. Допускаются отклонения от необходимой последовательности выполнения, не влияющие на правильность конечного результата. Работа показывает знание обучающимися основного теоретического материала и овладение умениями, необходимыми для самостоятельного выполнения работы. Могут быть неточности и небрежность в оформлении результатов работы.

– «удовлетворительно», если работа выполняется и оформляется обучающимися при сторонней помощи. На выполнение работы затрачивается много времени (можно дать возможность доделать работу дома). Обучающиеся показывают знания теоретического материала, но испытывают затруднение при самостоятельной работе.

– «неудовлетворительно» выставляется в том случае, когда обучающиеся не подготовлены к выполнению работы. Полученные результаты не позволяют сделать правильных выводов и полностью расходятся с поставленной целью. Показывается плохое знание теоретического материала и отсутствие необходимых умений.

В соответствии с учебным планом специальности 09.02.07 Информационные системы и программирование на внеаудиторную самостоятельную работу по профессиональному модулю ПМ.08 .Разработка дизайна веб-приложений предоставляется 24 часа.

Распределение часов внеаудиторной самостоятельной работы по темам, перечень видов внеаудиторной самостоятельной работы, методов и форм контроля представлен в таблице 1.

Таблица 1 Виды, формы и методы контроля внеаудиторной самостоятельной работы

Тема	Кол-во часов	Вид внеаудиторной самостоятельной работы	Методы и формы контроля формируемых компетенций
МДК.08.01 Проектирование и разработка интерфейсов пользователя			
1 семестр			
Тема 08.01.01 Основы web-технологий.	8	Создание интерфейса формы на html-странице	Проверка индивидуального задания
2 семестр			
Тема 08.01.02 Web-дизайн	4	Использование языка сценариев JavaScript при создании web-сайта	Проверка индивидуального задания
МДК.08.02. Графический дизайн и мультимедиа			
1 семестр			
Тема 08.02.02 Векторная графика	8	Создание Flash-баннера и Gif-анимации	Проверка индивидуального задания
2 семестр			
Тема 08.02.03 Растровая графика	4	Создание макета сайта, буклета	Проверка индивидуального задания
Всего	24		

2. Задания для выполнения внеаудиторной самостоятельной работы

По профессиональному модулю ПМ.08 Разработка дизайна веб-приложений обучающимися выполняются практические работы по индивидуальным заданиям.

Тема 08.01.01 Основы web-технологий

Создание интерфейса формы на html-странице

Задание:

- создать страницу регистрации с веб - формами, по образу многочисленных регистрационных веб - страниц, следующего вида:

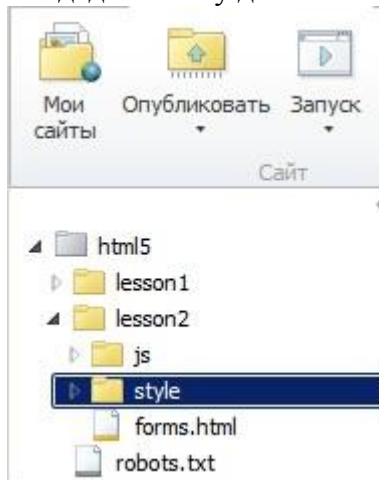
ФИО	<input type="text"/>
Email	<input type="text"/>
Телефон	<input type="text"/>
Почтовый индекс	<input type="text"/>
Адрес	<input type="text"/>
Дата	<input type="text"/>
Предпочитаемый способ получения уведомлений	<input type="radio"/> По телефону <input type="radio"/> email
<input type="button" value="Отправить данные"/>	

Необходимо также создать соответствующие стили для оформления веб - формы и валидацию вводимых значений.

Ход работы

Создание веб - форм

Создадим папку для стилей и основную *HTML* - страницу.



Рассмотрим по шагам создание требуемой страницы с регистрационной формой.

Шаг 1

Непосредственно создадим форму регистрации.:

```
<form id="registration">
  <fieldset>
    <legend>Форма регистрации</legend>
  </fieldset>
</form>
```

2.1.

Тег `<fieldset>` используется для логической группировки объектов формы.

Тег `<legend>` определяет заголовок.

Шаг 2

Добавим поле для ввода ФИО пользователя:

```
<fieldset>
  <legend>Форма регистрации</legend>

  <label for=fio>ФИО</label>
  <input id=fio name=fio type=text>
</fieldset>
```

На данном шаге форма будет выглядеть в браузере следующим образом:

Шаг 3

Создадим поля для ввода *email* и номера телефона, почтового индекса и города:

```
<fieldset>
  <legend>Форма регистрации</legend>

  <label for=fio>ФИО</label>
  <input id=fio name=fio type=text>

  <label for=email>Email</label>
  <input id=email name=email type=email>

  <label for=phone>Номер телефона</label>
  <input id=phone name=phone type=tel>
</fieldset>
```

Как можно видеть по следующему рисунку, результат немного иной, нежели мы ожидали:

Шаг 4

Для выравнивания элементов формы построчно и относительно друг друга необходимо создать таблицу с двумя колонками и без видимых границ. Изменим код следующим образом:

```
<fieldset>
  <legend>Форма регистрации</legend>
  <table class="alignment">
    <tr>
      <td> <label for=fio>ФИО</label> </td>
      <td> <input id=fio name=fio type=text> </td>
    </tr>
    <tr>
      <td> <label for=email>Email</label> </td>
      <td> <input id=email name=email type=email> </td>
    </tr>
    <tr>
      <td> <label for=phone>Номер телефона</label> </td>
      <td> <input id=phone name=phone type=tel> </td>
    </tr>
  </table>
</fieldset>
```


В результате получим следующее:

Форма регистрации

ФИО	<input type="text"/>
Email	<input type="text"/>
Номер телефона	<input type="text"/>

Шаг 5

Добавим поле для ввода даты рождения, для этого используем type=date (работает не со всеми браузерами). Добавим к таблице следующую строку:

```
<tr>
  <td><label for=dateofbirth>Дата рождения</label></td>
  <td><input id=dateofbirth name=dateofbirth type="date" /></td>
</tr>
```

В случае, если ваш браузер поддерживает данный тип элементов управления, получим следующее:

Форма регистрации

ФИО	<input type="text"/>
Email	<input type="text"/>
Номер телефона	<input type="text"/>
Город	<input type="text"/>
Дата рождения	Day.Month.Year ▾

<< < Май 2002 ▾ > >>

Пн	Вт	Ср	Чт	Пт	Сб	Вс
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

 Today Clear

Шаг 6

Для логической группировки, добавим еще один *fieldset*, объединяющий элементы, указывающие адрес и почтовый индекс:

```
<fieldset>
  <legend>Параметры доставки</legend>
  <table>
    <tr>
      <td><label for=address>Адрес</label></td>
      <td><textarea id=address name=address rows=5 ></textarea></td>
    </tr>
    <tr>
      <td><label for=postcode>Почтовый индекс</label></td>
      <td><input id=postcode name=postcode type=text ></td>
    </tr>
  </table>
</fieldset>
```

Поскольку написание полного адреса, скорее всего займет не одну строку, зададим атрибут rows, со значением равным 5 для соответствующего тега.

Форма регистрации	
ФИО	<input type="text"/>
Email	<input type="text"/>
Номер телефона	<input type="text"/>
Город	<input type="text"/>
Дата рождения	<input type="text"/>

Параметры доставки	
Адрес	<input type="text"/>
Почтовый индекс	<input type="text"/>

Шаг 7

Также в отдельную логическую группу выделим элемент управления, с помощью которого пользователь укажет предпочитаемый способ получения уведомлений:

```

<fieldset>
  <legend>Предпочитаемый способ получения уведомлений</legend>
  <table>
    <tr>
      <td><input id=emailmessage name=message type=radio></td>
      <td><label for=emailmessage>По Email</label></td>
    </tr>
    <tr>
      <td><input id=phonemessage name=message type=radio></td>
      <td><label for=phonemessage>По телефону</label></td>
    </tr>
    <tr>
      <td><input id=nomessage name=message type=radio></td>
      <td><label for=nomessage>Не уведомлять меня</label></td>
    </tr>
  </table>
</fieldset>

```

Форма регистрации	
ФИО	<input type="text"/>
Email	<input type="text"/>
Номер телефона	<input type="text"/>
Город	<input type="text"/>
Дата рождения	<input type="text"/>

Параметры доставки	
Адрес	<input type="text"/>
Почтовый индекс	<input type="text"/>

Предпочитаемый способ получения уведомлений	
<input type="radio"/>	По Email
<input type="radio"/>	По телефону
<input type="radio"/>	Не уведомлять меня

Шаг 8

Осталось только добавить на страницу кнопку:

```
<fieldset>
  <button type=submit>Отправить данные</button>
</fieldset>
```

Форма регистрации

ФИО

Email

Номер телефона

Город

Дата рождения

Параметры доставки

Адрес

Почтовый индекс

Предпочитаемый способ получения уведомлений

☐ По Email

☐ По телефону

☐ Не уведомлять меня

Отправить данные

На этом создание формы и размещение элементов управления заканчивается. Можно переходить к следующему пункту задания.

Стилизация форм

Шаг 1

Создадим стиль для всей формы – зададим цвет фона, ширину и т.д.:

```
form#registration {
  background: #1E90FF;
  -moz-border-radius: 5px;
  -webkit-border-radius: 5px;
  -khtml-border-radius: 5px;
  border-radius: 5px;
  counter-reset: fieldsets;
  padding: 20px;
  width: 400px;
}
```

Шаг 2

Уберем границы у *fieldset* и добавим отступ:

```
form#registration fieldset {
  border: none;
  margin-bottom: 10px;
}
```

Шаг 3

Стилизуем наши *legend*’ы сделаем их жирными и применим к ним светло синий *text-shadow* :

```
form#registration legend {
  color: #384313;
  font-size: 16px;
  font-weight: bold;
  padding-bottom: 10px;
```

```
text-shadow: 0 1px 1px #C1F7FF;
}
```

Шаг 4

Осталось стилизовать элементы label, input и button. Все label должны выглядеть одинаково, кроме label'a который прописан для элементов radio. Выровняем их по левому краю и придадим ширину:

```
form#registration label {
  float: left;
  font-size: 13px;
  width: 110px;
}
```

```
form#registration fieldset label {
  background: none no-repeat left 50%;
  line-height: 20px;
  padding: 0 0 0 30px;
  width: auto;
}
```

```
form#registration button {
  background: #87CEEB;
  border: none;
  -moz-border-radius: 20px;
  -webkit-border-radius: 20px;
  -khtml-border-radius: 20px;
  border-radius: 20px;
  color: #ffffff;
  display: block;
  font: 18px Georgia, "Times New Roman", Times, serif;
  letter-spacing: 1px;
  margin: auto;
  padding: 7px 25px;
  text-shadow: 0 1px 1px #C1F7FF;
  text-transform: uppercase;
}
```

В результате форма будет выглядеть следующим образом:

Форма регистрации

ФИО

Email

Номер телефона

Город

Дата рождения

Параметры доставки

Адрес

Почтовый индекс

Предпочитаемый способ получения уведомлений

☐ По Email

☐ По телефону

☐ Не уведомлять меня

ОТПРАВИТЬ ДАННЫЕ

Безусловно, созданные стили не являются "шедевром", дальнейшие эксперименты со стилизацией формы остаются на самостоятельное рассмотрение.

Валидация вводимых значений

Рассмотрим следующий участок нашего *HTML* кода:

```
<input id=email name=email type=email />
```

Атрибут типа равен "email", а не "text". Самое ценное в новых HTML-типах input в том, что вы можете использовать их сейчас и они будут работать на том или ином уровне в любом браузере.

Когда браузер встречает один из этих типов и не поддерживает новые input-типы, объявление типа не распознается. В таком случае браузер корректно сокращает функциональность и интерпретирует элемент как type="text".

Чтобы сделать элемент управления обязательным к заполнению, достаточно вставить в создающий его тег атрибут REQUIRED. Это атрибут тега без значения.

```
<input id=name name=name type=text required>
```

Задать минимальное и максимальное значение и шаг числовых значений можно только для полей ввода числовых величин. Другие элементы управления, в том числе и обычные поля ввода, эту возможность не поддерживают.

Для задания минимального значения числа используется атрибут тега MIN, а для задания максимального значения - атрибут тега MAX. В качестве их значений указываются числа.

```
<INPUT TYPE=" number" ID=" txtAge" REQUIRED MIN=" 1" MAX=" 100" >
```

Задание маски ввода

Маска ввода задаёт формат, которому должно соответствовать вводимое значение. Как правило, она указывается для обычных полей ввода.

Для указания маски используется атрибут стиля PATTERN. В качестве его значения указывается регулярное выражение, собственно, и задающее маску ввода.

```
<input id=phone name=phone type=tel pattern=" [()\\d{3,4}()]\s\d{2,3}[-]\d{2}[-]\d{2}" >
```

Эта веб-форма содержит поле ввода, в котором указывается номер телефона в формате ([x]xxx) [x]xx-xx-xx, где x - цифра.

Если посетитель введёт в поле ввода с указанной маской значение, не соответствующее данной маске, то увидит всплывающее сообщение с предупреждением о неверном формате введенного значения.

Для успешного завершения выполнения задания самостоятельно необходимо сделать следующее:

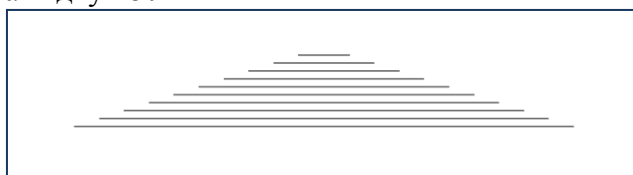
- Поля ФИО, email и Номер телефона, определить как обязательные к заполнению.
- Создать маску ввода для поля Номер телефона.
- Осуществить проверку вводимых значений поля Почтовый индекс.
- Ознакомиться с материалами для самостоятельного изучения.

Тема 08.01.02 Web-дизайн

Использование языка сценариев JavaScript при создании web-сайта

Данная работа предполагает знакомство с размещением кода JavaScript в html-документе, методами document.write, циклическими и условными конструкциями.

1.Нарисовать пирамидку из линий



```
<script>
  for(i=1;i<=10;i++)
    document. Write("<hr width=\""+i*10+\"%\">")
</script>
```

Постройте треугольник из убывающих линий, т.е. самая длинная сверху, короткая внизу.



2. Постройте таблицу умножения для числа 5 на числа от 1 до 9.

a=5

```
for (i = 1; i <=9; i++)
```

```
{
document.write(i+ "*" +a+ "=" + (i*a) + "<br>");
}
```

Постройте таблицу умножения для любого вводимого числа на числа от 20 до 1.

3. Модифицируйте предыдущий пример, взяв каждое значение в ячейку таблицы.

```
for (i = 1; i <=100; i++)
```

```
{
document.write ("<tr>");
document.write("<td>" +i+"</td>");
document.write ("</tr>");
}
```

4. Создайте полную таблицу умножения, как показано на рисунке ниже.

1*1=1	2*1=2	3*1=3	4*1=4	5*1=5	6*1=6	7*1=7	8*1=8	9*1=9	10*1=10
1*2=2	2*2=4	3*2=6	4*2=8	5*2=10	6*2=12	7*2=14	8*2=16	9*2=18	10*2=20
1*3=3	2*3=6	3*3=9	4*3=12	5*3=15	6*3=18	7*3=21	8*3=24	9*3=27	10*3=30
1*4=4	2*4=8	3*4=12	4*4=16	5*4=20	6*4=24	7*4=28	8*4=32	9*4=36	10*4=40
1*5=5	2*5=10	3*5=15	4*5=20	5*5=25	6*5=30	7*5=35	8*5=40	9*5=45	10*5=50
1*6=6	2*6=12	3*6=18	4*6=24	5*6=30	6*6=36	7*6=42	8*6=48	9*6=54	10*6=60
1*7=7	2*7=14	3*7=21	4*7=28	5*7=35	6*7=42	7*7=49	8*7=56	9*7=63	10*7=70
1*8=8	2*8=16	3*8=24	4*8=32	5*8=40	6*8=48	7*8=56	8*8=64	9*8=72	10*8=80
1*9=9	2*9=18	3*9=27	4*9=36	5*9=45	6*9=54	7*9=63	8*9=72	9*9=81	10*9=90
1*10=10	2*10=20	3*10=30	4*10=40	5*10=50	6*10=60	7*10=70	8*10=80	9*10=90	10*10=100

5. Организуйте объединение двух ячеек по горизонтали и двух по вертикали, как это показано на рисунке. (Цель данной части задания исключительно учебная. Необходимо использовать условный оператор.)

1*1=1	2*1=2	3*1=3	4*1=4	5*1=5	6*1=6	7*1=7	8*1=8	9*1=9	10*1=10
1*2=2	2*2=4	3*2=6	4*2=8	5*2=10	6*2=12	7*2=14	8*2=16	9*2=18	10*2=20
1*3=3	2*3=6	3*3=9	4*3=12	5*3=15	6*3=18	7*3=21	8*3=24	9*3=27	10*3=30
1*4=4	2*4=8	3*4=12	4*4=16	5*4=20	6*4=24	7*4=28	8*4=32		10*4=40
1*5=5	2*5=10	3*5=15	4*5=20		6*5=30	7*5=35	8*5=40	9*5=45	10*5=50
1*6=6	2*6=12	3*6=18	4*6=24	5*6=30	6*6=36	7*6=42	8*6=48	9*6=54	10*6=60
1*7=7	2*7=14	3*7=21	4*7=28	5*7=35	6*7=42	7*7=49	8*7=56	9*7=63	10*7=70
1*8=8	2*8=16	3*8=24	4*8=32	5*8=40	6*8=48	7*8=56	8*8=64	9*8=72	10*8=80
1*9=9	2*9=18	3*9=27	4*9=36	5*9=45	6*9=54	7*9=63	8*9=72	9*9=81	10*9=90
1*10=10	2*10=20	3*10=30	4*10=40	5*10=50	6*10=60	7*10=70	8*10=80	9*10=90	10*10=100

МДК.08.02. Графический дизайн и мультимедиа

Тема 08.02.02 Векторная графика

Создание Flash-баннера и Gif-анимации

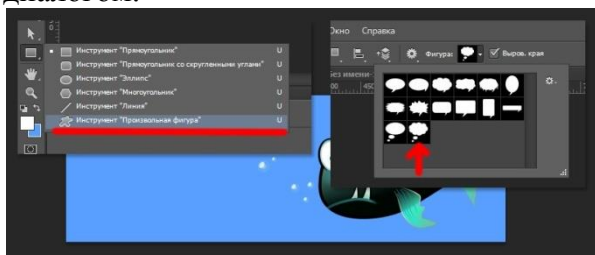
Создадим новый документ размером 600 на 200 px.

Фон — #589fff

Для начала соберем баннер в статичном виде. Расположим на слое нашего главного персонажа – рыбу, где можно взять бесплатные ресурсы для своих работ я писала в статье «ТОП 8 сайтов с бесплатными ресурсами для дизайнера»:



С помощью инструмента «Произвольная фигура» расположим на слое выноску с диалогом.



Напомню, что отредактировать фигуру (векторный объект) можно с помощью инструмента «Выделение узла».



удаляем ненужные элементы в слое с первой фигурой и добавила еще 3 Эллипса, чтобы сделать так, как-будто это облако диалога идет изо рта рыбы.

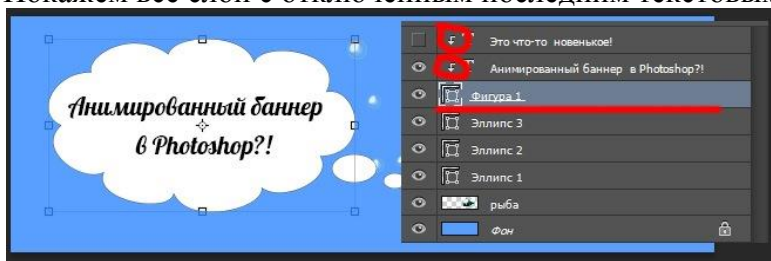
Получилось следующее.



Теперь поместим текст на слоях документа, в порядке, в котором он будет появляться.

Так как текст у нас будет появляться поочередно, то, естественно, что в статичном варианте слои будут налезать друг на друга.

Покажем все слои с отключенным последним текстовым слоем.



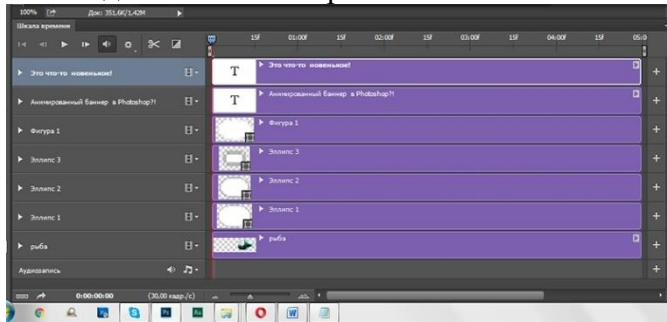
текстовые слои должны быть кадрированы по размерам слоя с фигурой диалогового облака, чтобы оно служило рамкой, из-за которой текст уже не будет видно.

Итак, все слои готовы. Расположены в порядке их появления (снизу вверх). Можем приступить к анимации.

Выберите пункт Окно – Шкала времени.

В появившейся панели выберите кнопку Создать шкалу времени для видео.

В результате создастся временная шкала, со всеми нашими слоями, которые мы уже можем двигать и анимировать.

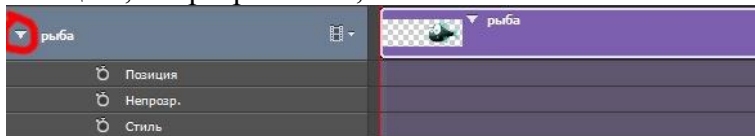


Так как слой Фон с замочком, то отображаться он не будет на временной шкале, он будет присутствовать всегда неизменно.

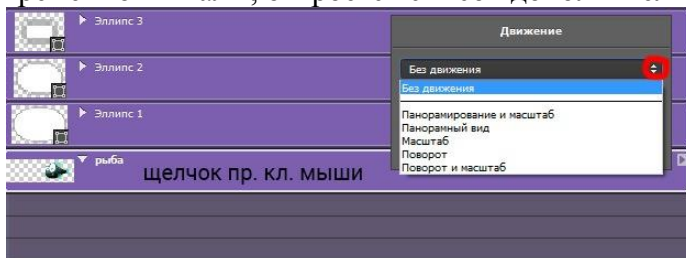
Если хотите анимировать фон, то снимите блокировку на его слое, щелкнув 2 раза левой клавишей мыши.

Для каждого слоя на временной шкале предусмотрены атрибуты для анимации.

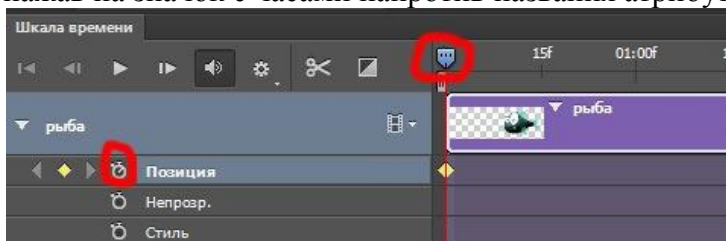
Их можно увидеть, если раскрыть список слева от названия слоя. Мы можем изменять позицию, непрозрачность, стиль.



Так же, если щелкнуть правой клавишей мыши по самому названию слоя на дорожке временной шкалы, откроется список дополнительных анимационных действий.



Активировать создание ключевых кадров для конкретной дорожки со слоем, можно, нажав на значок с часами напротив названия атрибута анимации.



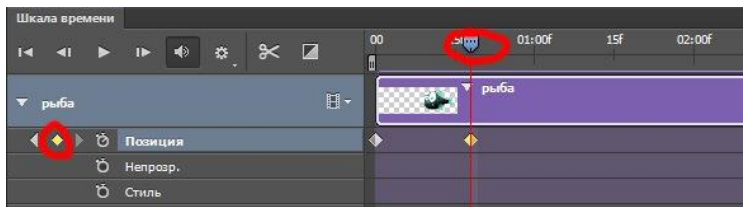
Желтый ромб (пиктограмма ключевого кадра) появится там, где стоит временной указатель на шкале времени. Например, в начале.

Вы можете изменить значение атрибута объекта, например, позиции и ее значение будет сохранено для данного ключевого кадра.

Затем, чтобы задать новое значение и новый ключевой кадр, нужно переместить указатель времени на новое место.

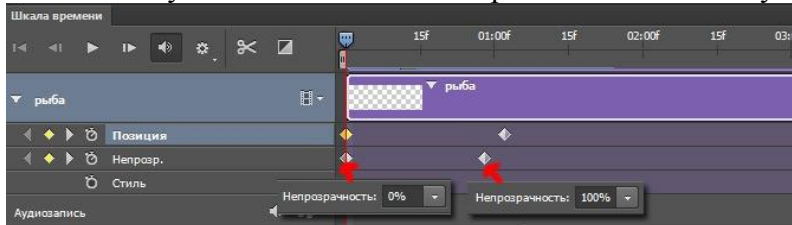
Щелкнуть по значку с ромбом напротив названия атрибута (на временной шкале появится новая пиктограмма ключевого кадра) и задать новое значение атрибута.

Например, переместить объект в новое место.



Теперь попробуем применить полученные знания.

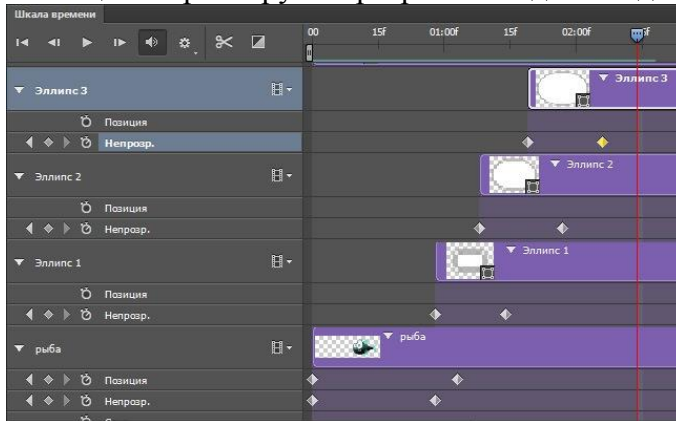
Наша рыба должна появиться слева направо и ее прозрачность должна поменяться с 0 до 100%. Так будет выглядеть шкала времени в данном случае.



После того, как рыба появилась, появится диалоговое облако по частям.

Сначала эллипсы, потом сама фигура.

Дорожки каждого слоя мы можем двигать на временной шкале, поэтому распределим их позиции так, чтобы они появлялись с нужным временным интервалом и зададим анимацию параметру Непрозрачность для каждого слоя.



Для фигуры применим те же действия.

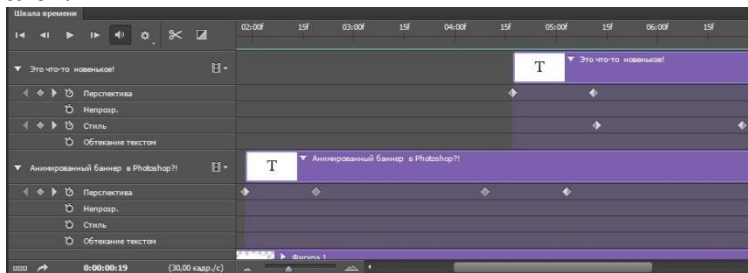
Теперь создадим анимацию для текста, используя тот же алгоритм.

Только у текста будет параметр не Позиция, а Перспектива, при этом мы сможем не только изменять позицию, но и изменять масштаб от меньшего к большему, при начале движения текста.

Для этого используется стандартный инструмент трансформации объекта (**Ctrl + T**).

Так как изначально слои с текстом у нас кадрированы по форме фигуры, то они будут видимы только в ее пределах.

У второго текста я так же изменила стиль. Добавила градиент в параметрах наложения слоя.



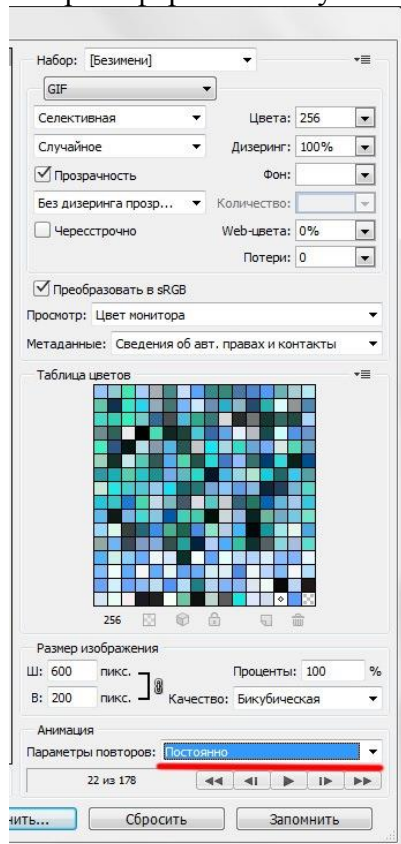
Теперь сравниваем все края дорожек в конце.

Для этого нужно потянуть за правый край каждой дорожки и подогнать все под одно время. Чтобы резко не пропадали все объекты.

Для каждой дорожки можно задать изменение непрозрачности в 0. Пройграть то, что получилось, можно нажав на клавишу пробел.

Чтобы сохранить результат, необходимо выбрать Файл – Сохранить для Web.

Выбрать формат Gif и указать, чтобы анимация повторялась постоянно.



Хоть Photoshop и не создан для какой-то сложной анимации (и это резонно), но с помощью вот таких нехитрых действий можно добиться достойного результата при создании gif баннера

Тема 08.02.03 Растровая графика

Создание макета, буклета

Создание буклета

Работа над подобными изданиями делится на два больших этапа – проектирование макета и дизайн документа.

Макет

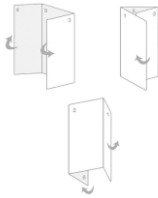
Как известно, буклет состоит из трех отдельных частей или из двух разворотов, с информацией на лицевой и тыльной стороне. Исходя из этого, нам понадобится два отдельных документа.

Каждая сторона делится на три части.



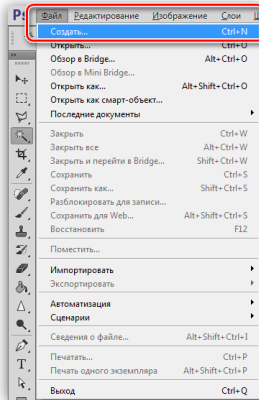
Далее необходимо определиться, какие данные будут располагаться на каждой стороне. Для этого лучше всего подойдет обычный лист бумаги. Именно этот «дедовский» способ позволит Вам понять, каким образом должен выглядеть конечный результат.

Лист сворачивается, как буклет, а потом наносится информация.

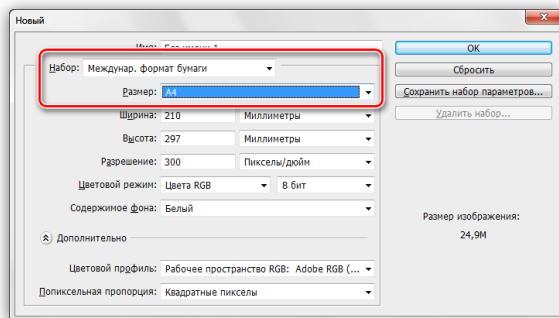


Когда концепт готов, можно приступить к работе в Фотошопе. При проектировании макета не бывает маловажных моментов, поэтому будьте максимально внимательны.

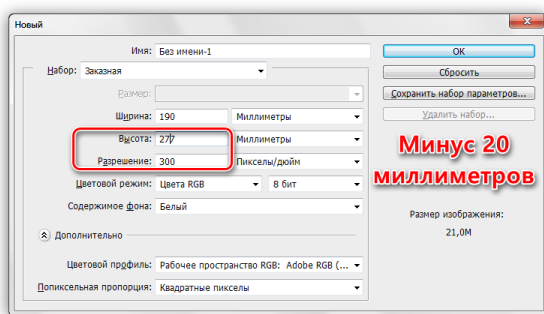
1. Создаем новый документ в меню «Файл».



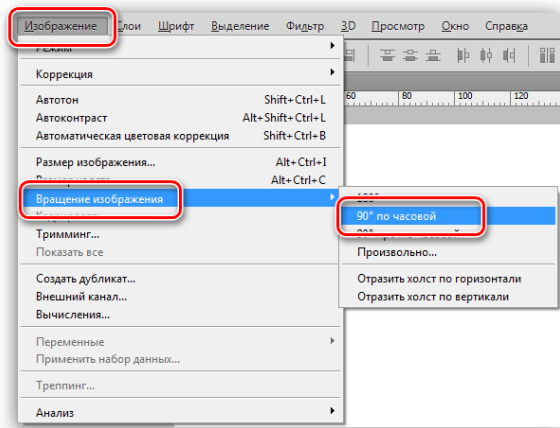
2. В настройках указываем «Международный формат бумаги», размер A4.



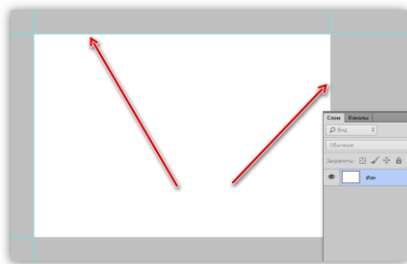
3. От ширины и высоты отнимаем по 20 миллиметров. Впоследствии мы их добавим к документу, но при печати они будут пустыми. Остальные настройки не трогаем.



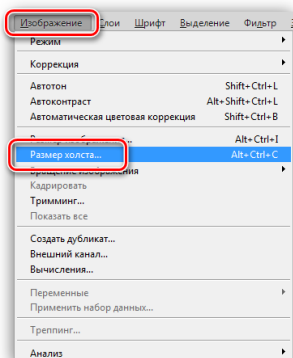
4. После создания файла идем в меню «Изображение» и ищем пункт «Вращение изображения». Повернем холст на 90 градусов в любую сторону.



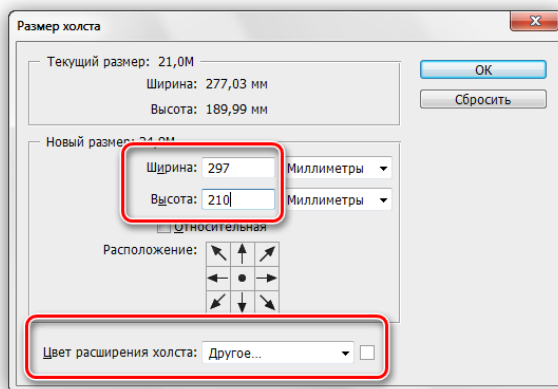
5. Далее нам необходимо определить линии, ограничивающие рабочую область, то есть, поле для размещения контента. Выставляем направляющие по границам холста.



6. Обращаемся к меню «Изображение – Размер холста».

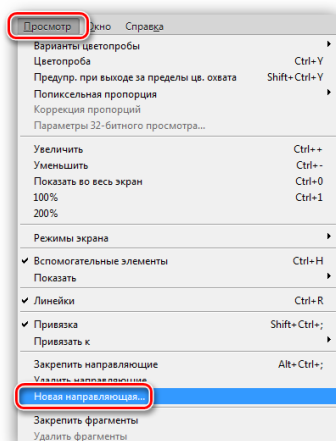


7. Добавляем ранее отнятые миллиметры к высоте и ширине. Цвет расширения холста должен быть белым. Обратите внимание, что значения размеров могут оказаться дробными. В этом случае просто возвращаем исходные значения формата A4.

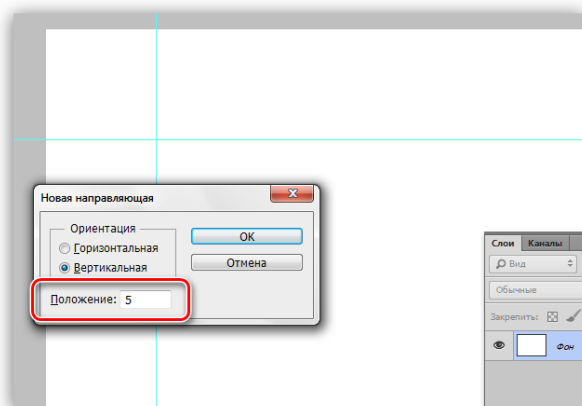


8. Имеющиеся в данный момент направляющие будут играть роль линий отреза. Для наилучшего результата фоновое изображение должно немного выходить за эти границы. Достаточно будет 5 миллиметров.

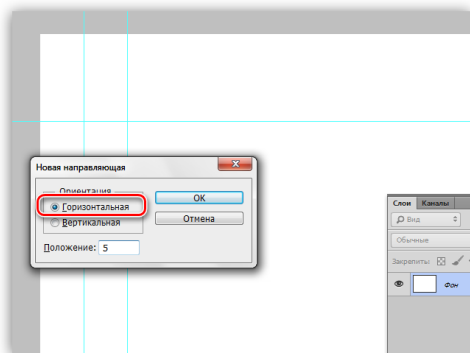
- Идем в меню «Просмотр – Новая направляющая».



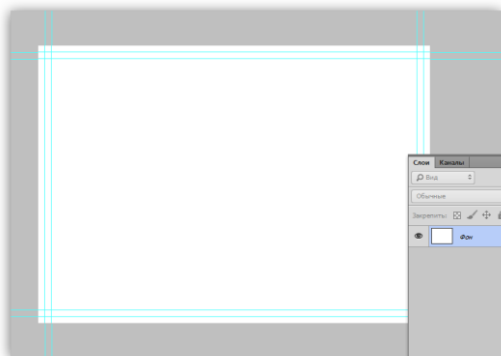
- Первую вертикальную линию проводим в 5 миллиметрах от левого края.



- Таким же способом создаем и горизонтальную направляющую.

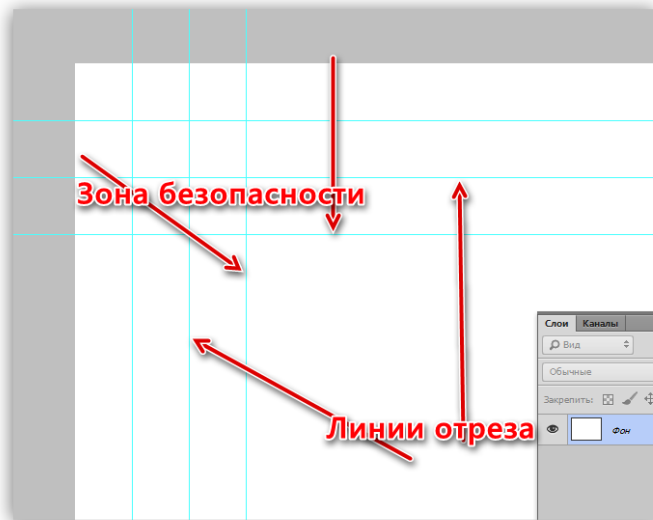


- Путем нехитрых вычислений определяем положение остальных линий ($210-5=205$ мм, $297-5=292$ мм).



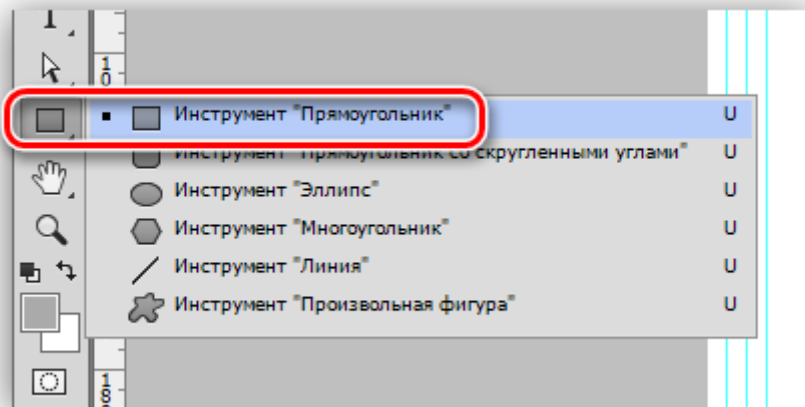
9. При обрезке печатной продукции могут быть допущены ошибки в силу различных причин, что может повредить контент на нашем буклете. Для того, чтобы избежать таких неприятностей, нужно создать так называемую «зону безопасности», за границами

которой не располагаются никакие элементы. Фонового изображения это не касается. Размер зоны также определим в 5 миллиметров.

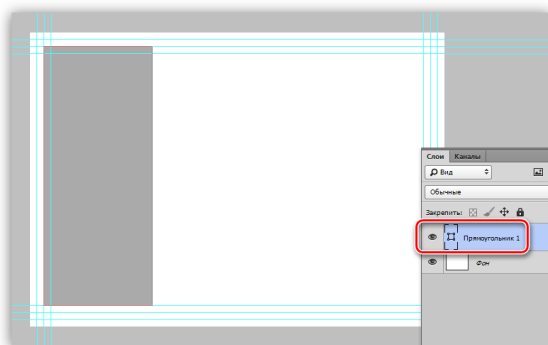


10. Как мы помним, наш буклет состоит из трех равных частей, и перед нами стоит задача создать три равные зоны для контента. Можно, конечно, вооружиться калькулятором и высчитать точные размеры, но это долго и неудобно. Существует прием, который позволяет быстро поделить рабочее пространство на равные по размеру участки.

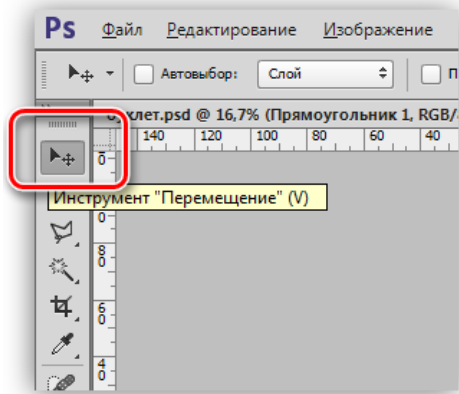
- Выбираем на левой панели инструмент «Прямоугольник».



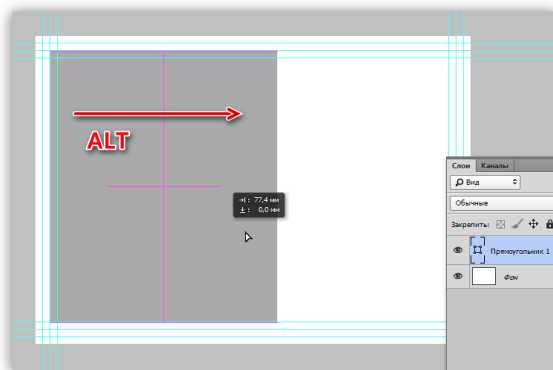
- Создаем фигуру на холсте. Размер прямоугольника не имеет значения, главное, чтобы суммарная ширина трех элементов была меньше ширины рабочей области.



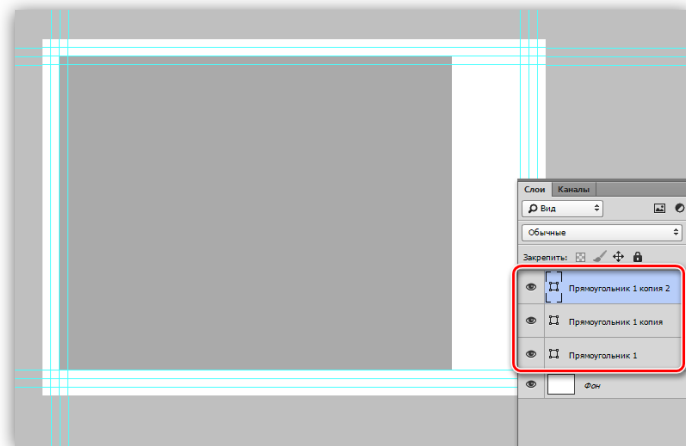
- Выбираем инструмент «Перемещение».



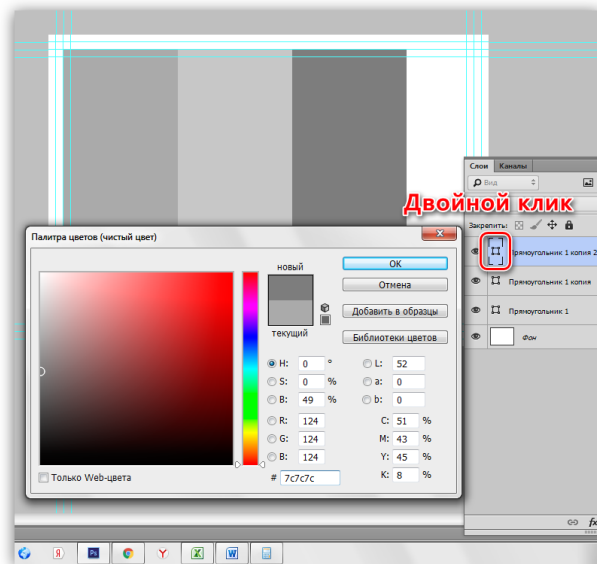
- Зажимаем клавишу *ALT* на клавиатуре и перетягиваем прямоугольник вправо. Вместе с перемещением создастся его копия. Следим, чтобы между объектами не было зазора и нахлеста.



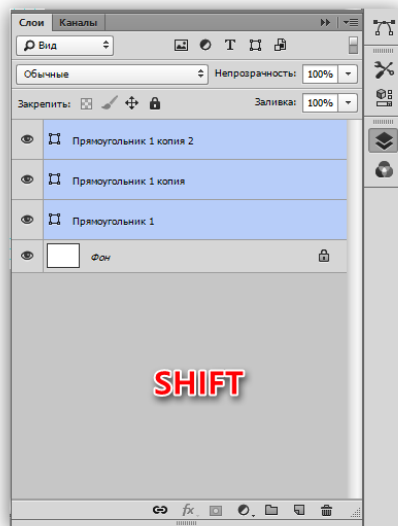
- Таким же способом делаем еще одну копию.



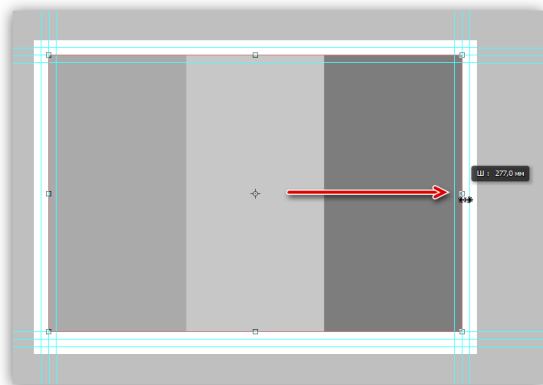
- Для удобства поменяем цвет каждой копии. Делается при помощи двойного клика по миниатюре слоя с прямоугольником.



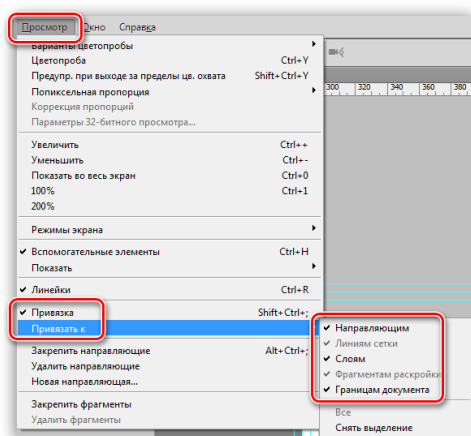
- Выделяем все фигуры в палитре с зажатой клавишей *SHIFT* (клик по верхнему слою, *SHIFT* и клик по нижнему).



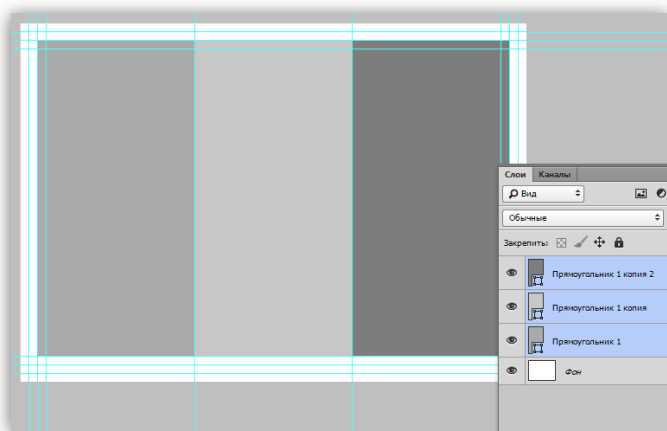
- Нажав горячие клавиши *CTRL+T*, применяем функцию «Свободное трансформирование». Беремся за правый маркер и растягиваем прямоугольники вправо.



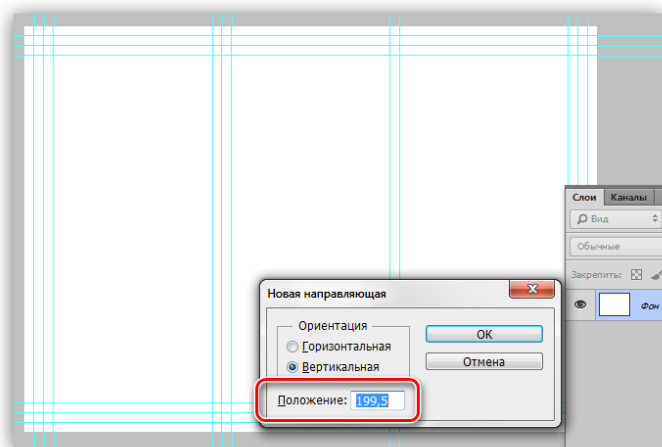
- После нажатия клавиши *ENTER* у нас получится три равные фигуры.
11. Для точного проведения направляющих, которые будут делить рабочую область буклета на части, необходимо включить привязку в меню «Просмотр».



12. Теперь новые направляющие «прилипнут» к границам прямоугольников. Вспомогательные фигуры нам больше не нужны, можно их удалить.

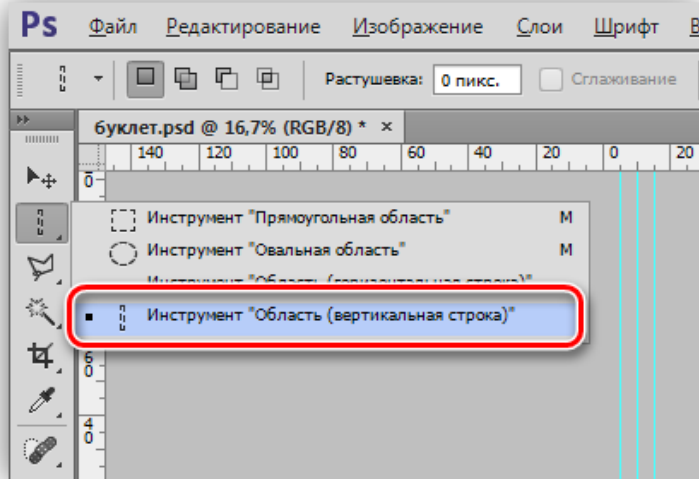


13. Как мы уже говорили ранее, для контента необходима зона безопасности. Поскольку буклет будет сгибаться по линиям, которые мы только что определили, то на этих участках объектов быть не должно. Отступим от каждой направляющей по 5 миллиметров с каждой стороны. Если значение дробное, то разделителем должна быть запятая.

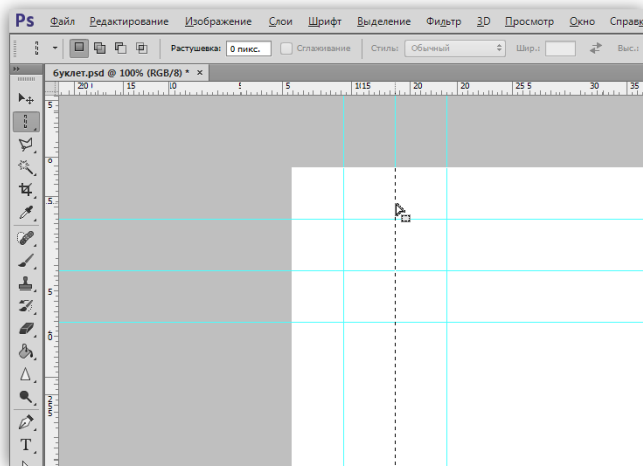


14. Последним шагом будет проведение линий отреза.

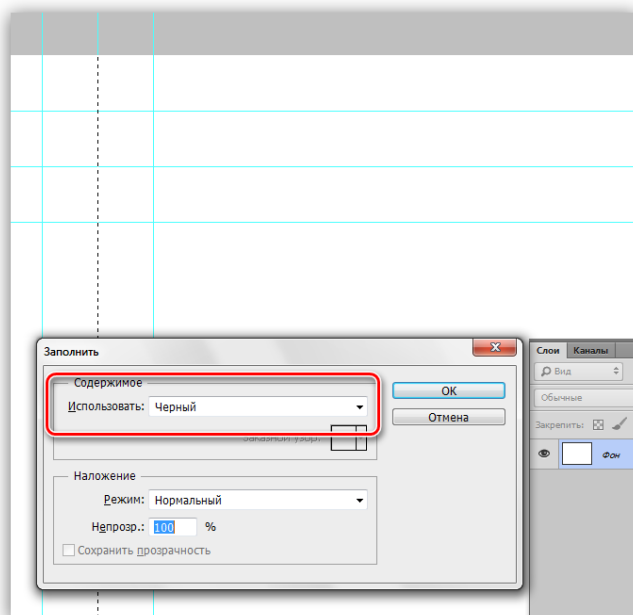
- Берем инструмент «Вертикальная строка».



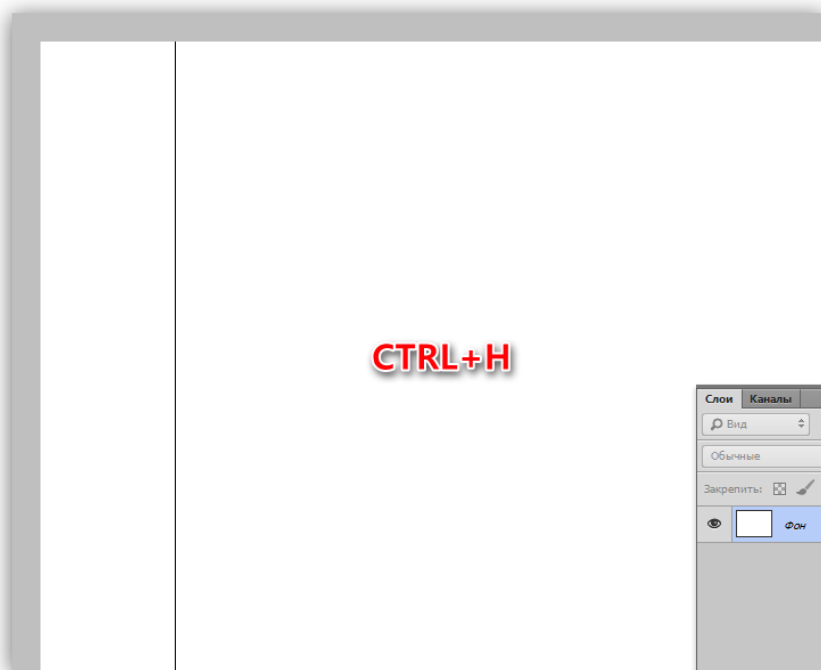
- Кликаем по средней направляющей, после чего появится вот такое выделение толщиной 1 пиксель:



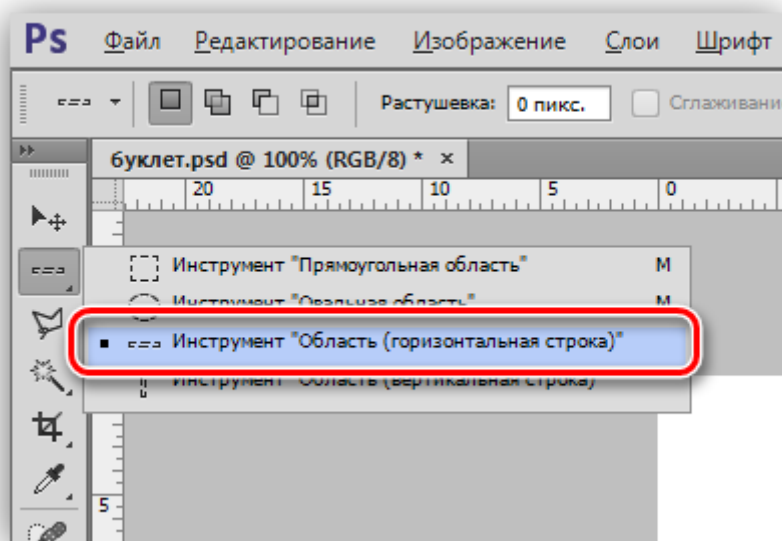
- Вызываем окно настройки заливки горячими клавишами **SHIFT+F5**, выбираем в выпадающем списке черный цвет и жмем **OK**. Выделение убирается комбинацией **CTRL+D**.



- Для просмотра результата можно временно скрыть направляющие сочетанием клавиш **CTRL+H**.



- Горизонтальные линии проводятся при помощи инструмента «Горизонтальная строка».



На этом создание макета буклета завершено. Его можно сохранить и использовать в дальнейшем, как шаблон.

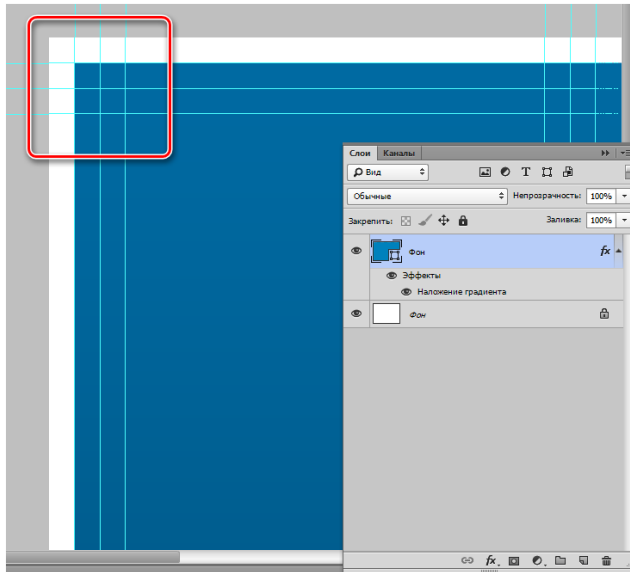
Дизайн

Дизайн буклета – дело индивидуальное. Все составляющие дизайна обусловлены либо вкусом, либо техническим заданием. В данном уроке обсудим лишь несколько моментов, на которые следует обратить внимание.

1. Фоновое

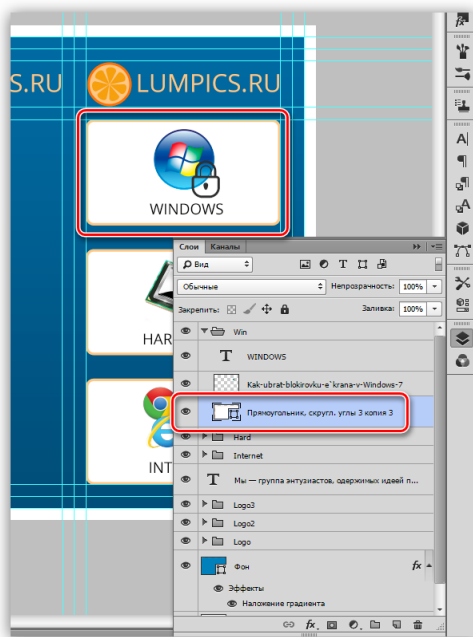
изображение.

Ранее, при создании шаблона, мы предусмотрели отступ от линии отреза. Это нужно для того, чтобы при обрезке бумажного документа не оставались белые участки по периметру. Фон должен доходить именно до линий, определяющих этот отступ.



2. Графика.

Все создаваемые графические элементы необходимо изображать при помощи фигур, так как залитая цветом выделенная область на бумаге может иметь рваные края и лесенки.



3. При работе над дизайном буклета, не перепутайте информационные блоки: лицевой – справа, второй – тыльная сторона, третий блок будет первым, что увидит читатель, открыв буклет.

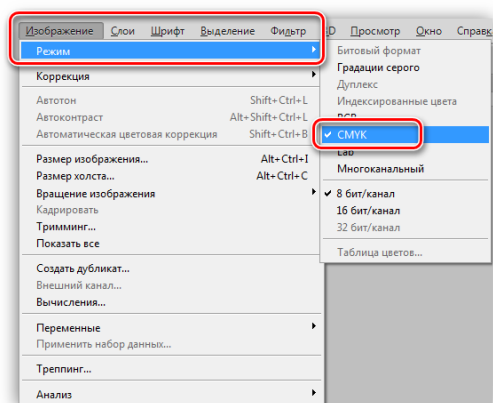


4. Данный пункт – следствие предыдущего. На первом блоке лучше расположить информацию, которая максимально четко отражает главную мысль буклета. Если это компания или, в нашем случае, сайт, то это могут быть основные направления деятельности. Желательно сопроводить надписи изображениями для большей наглядности.

В третьем блоке уже можно написать более подробно, чем мы занимаемся, а информация внутри буклета может, в зависимости от направленности, иметь как рекламный, так и общий характер.

Цветовая схема

Перед печатью настоятельно рекомендуется перевести цветовую схему документа в *CMYK*, поскольку большинство принтеров не в состоянии в полной мере отобразить цвета *RGB*.



Это можно сделать и в начале работы, так как цвета могут отображаться немного по-другому.

Сохранение

Сохранять такие документы можно как в формате *JPEG*, так и в *PDF*.

На этом урок о том, как создать буклет в Фотошопе, завершен. Неукоснительно следуйте инструкциям по проектированию макета и на выходе получите качественную полиграфию.

По профессиональному модулю ПМ.08 Разработка дизайна веб-приложений обучающимися выполняются практические работы по индивидуальным заданиям. К каждой практической работе обучающийся оформляет отчет.

Отчет по практической работе описывает проведенный эксперимент, обучающийся приводит расчеты и результаты, делает выводы. Оценивается такой отчет не только по содержанию, но и по оформлению. Особое внимание уделяется иллюстрациям, таблицам, формулам, программному коду.

Структура отчета

1. Титульный лист

Первая страница отчета — это титульный лист. Верх страницы начинается с названия учебного заведения и кафедры. Затем идет заголовок «Отчет по практической работе по дисциплине ...» и тема. В правой части страницы указывается автор (студент), его группа, преподаватель. В нижней части титульного листа пишется город и год составления работы.

2. Введение

Сразу после титульного листа начинается вводный раздел. Считается, что отчет по практической работе имеет недостаточно большой объем для того, чтобы в нем требовалось оглавление. Во введении автор обозначает цель и перечисляет решаемые в соответствии с ней задачи.

Приводится информация об объекте и предмете исследования, то есть изучаемой научной области и рассматриваемой в ней проблеме. Введение должно погрузить читателя отчета в суть работы.

3. Теоретическая часть

Теоретическая часть зависит от выполняемой работы. В части практических работ требуется описание используемых определений и правил или необходимых формул. Также тут приводятся теоремы, законы, фрагменты программного кода. Требуемые сведения излагаются в сжатом виде, тезисно.

4. Практическая часть

После основных теоретических сведений, которые нужны для понимания темы практической работы, начинается описание эксперимента. Здесь требуется перечислить следующее:

- техническое оснащение, используемые программы;
- метод эксперимента (как правило, он приводится в задании к работе);
- ход практической работы, которую выполнял студент.

Практическая часть составляется так, чтобы преподаватель мог по ней понять, что автор самостоятельно выполнял все действия и воспроизвести его действия. Приводятся коды выполненных программ.

В конце практической части обычно составляется таблица, в которой наглядно отображены все полученные результаты (и иногда — начальные условия). Если в практической работе требовались промежуточные результаты, то они тоже фиксируются.

5. Заключение

В заключительной части обучающийся должен подытожить выполненные задачи, кратко перечисляет полученные данные, на их основе делает собственные выводы.

Общие требования к оформлению

Текст печатается на листах формата А4.

Цвет шрифта – черный. Размер шрифта (кегель) 14. Тип шрифта – Times New Roman. Шрифт печати должен быть прямым, четким, черного цвета, одинаковым по всему объему текста. При выделении заголовков структурных частей используется полужирный шрифт (введение, название главы, заключение и т.д.).

Текст обязательно выравнивается по ширине.

Размер абзацного отступа – 1,25 см, межстрочный интервал – 1,5.

Поля: левое – 25 мм, правое – 15 мм, верхнее и нижнее 20 мм.

Нумерация страниц работы должна быть сквозной, включая список использованных источников и приложения. Нумерация начинается со страницы 2 (введение), первой страницей является титульный лист, второй – введение и так далее, последней – первая страница приложения. Номер страницы проставляют арабскими цифрами, шрифт Times New Roman, размер шрифта 12 в центре нижней части листа без точки. На титульном листе номер страницы не ставится.

Требования к оформлению отчетов:

Отчёт оформляется на листах формата А4 шрифтом Times 14 (13) одиночным интервалом.

Каждая задача начинается с нового листа.

Рукописные тексты и рисунки в отчётах не допускаются.

Скриншоты экранов не допускаются.

Диаграммы только в печатном виде с помощью фигур MS Word или Visio.

Формулы и уравнения выполняются с помощью MS Office Equation Editor.

Требования к оформлению листингов программ:

Листинги программ оформляются шрифтом Courier 11 одиночным интервалом.

Перед каждой разработанной функцией должен быть комментарий, рассказывающий, что эта функция делает, что возвращает и какие аргументы принимает.

В теле функции должны быть поясняющие комментарии, если части алгоритма не являются очевидными.

При выполнении заданий допускается использование языков программирования C#, C++, Java.

3. Рекомендуемая литература

Печатные издания

1. Графический дизайн. Современные концепции [Текст] : учеб. пособие / отв. ред. Е.Э. Павловская. - 2-е изд, перераб. и доп. - М. : Юрайт, 2017. - 183 с. - (Университеты России).

2. Хеллер, С. IDEA BOOK. Графический дизайн [Текст] / С.Хеллер, Г.Андерсон. - СПб : Питер, 2017. - 120с.: ил. - (Современный дизайн).

Электронные издания (электронные ресурсы)

1. Графический дизайн. Современные концепции : учебное пособие / Е.Э. Павловская [и др.]; отв. ред. Е.Э. Павловская. — 2-е изд., перераб. и доп. — Москва: Юрайт, 2020. — 119 с. — Текст: электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/454541> (дата обращения: 14.05.2020).

2. Основы дизайна и композиции: современные концепции: учебное пособие для спо / Е.Э. Павловская [и др.]; ответственный редактор Е.Э. Павловская. — 2-е изд., перераб. и доп. — Москва: Юрайт, 2020. — 119 с. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/456065> (дата обращения: 14.05.2020).

3. Цифровые технологии в дизайне. История, теория, практика: учебник и практикум / А.Н. Лаврентьев [и др.]; под редакцией А.Н. Лаврентьева. — 2-е изд., испр. и доп. — Москва: Юрайт, 2020. — 208 с. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/454519> (дата обращения: 14.05.2020).

4. Полуэктова, Н.Р. Разработка веб-приложений : учебное пособие / Н. Р. Полуэктова. — Москва: Юрайт, 2021. — 204 с. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/466449> (дата обращения: 14.05.2020).

5. Сысолетин, Е. Г. Разработка интернет-приложений : учебное пособие / Е. Г. Сысолетин, С. Д. Ростунцев; под научной редакцией Л. Г. Доросинского. — Москва: Юрайт, 2020. — 90 с. — Текст: электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/453345> (дата обращения: 14.05.2020).

6. Тузовский, А.Ф. Проектирование и разработка web-приложений: учебное пособие для спо / А.Ф. Тузовский. — Москва: Юрайт, 2020. — 218 с. — Текст: электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/456394> (дата обращения: 14.05.2020).

Дополнительные источники (при необходимости)

1. Зельдман,Д. Web-дизайн по стандартам / Дж.Зельдман. - М.: НТ Пресс,2005. - 440с. - (ШколаWeb-мастерства).

2. Томас,Д. Гибкая разработка веб-приложений в среде Rails [Текст] / Д.Томас,Д.Х.Хэнссон. - СПб : Питер, 2008. - 716с.: ил. - (Библиотека программиста).

3. Якобсон,Й. Концепция разработки Web-сайтов: Как успешно разработать web-сайт с применением мультимедийных технологий / Й.Якобсон. - М.: НТ Пресс, 2006. - 512с. - (Школа Web-мастерства).

4. Лесняк, Владимир Графический дизайн: основы профессии [Текст] / В.Лесняк. - М. : ИндексМаркет, 2011. - 416с. : ил.

5. Луптон, Э. Графический дизайн от идеи до воплощения [Текст] / Э.Луптон; пер. с англ. В.Иванов. - СПб: Питер, 2014. - 184с. : ил.

6. Фрейзер, Том Графический дизайн. Мастер -класс [Текст] / Том Фрейзер, Адам Бэнкс. - РИП-холдинг: М., 2012. - 256с.: ил. - (Мастер-класс).

7. Хембри, Р. Самый полный справочник [Текст] Графический дизайн : Как научиться понимать графику и визуальные образы / Р.Хембри. - М. : АСТ, 2008. - 192с. : ил.