

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Факультет Инновационных технологий
Кафедра управления инновациями

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

по дисциплине Веб-программирование

Составлены кафедрой управления инновациями для студентов, обучающихся
по направлениям подготовки «Инноватика», «Управление качеством»

Форма обучения очная

Составитель
Ст. преподаватель кафедры управления инновациями

Д.Ф. Вячистый
«11» октября 2018 г.

Томск 2018

Оглавление

Введение	3
Общие требования	3
Материально-техническое обеспечение лабораторных работ	4
Прием результатов выполнения лабораторных работ	4
Темы лабораторных работ	5
Оформление отчетов по лабораторным работам	13
Учебно-методическое и информационное обеспечение дисциплины	13
Приложение А. Образец титульного листа отчета по лабораторным работам	15

Введение

Дисциплина «Веб-программирование» играет важную роль в формировании профессиональных знаний в области Инноватики и Управления качеством. Изучение дисциплины имеет цель: овладение информационными технологиями, основными прикладными программными средствами, базовыми навыками алгоритмизации, веб-программирования; изучение основных принципов построения интернет-ориентированного программного обеспечения – веб-сервисов, сайтов, онлайн-ресурсов с использованием современных языков, методов, средств, технологий, алгоритмов веб-программирования. Полученные знания и навыки могут быть использованы в формировании системы менеджмента качества организации; организации аудита и сертификации систем менеджмента; осуществлении работ по управлению качеством ресурсов организации; осуществлении работ по управлению качеством эксплуатации продукции; организации проведения и осуществлению работ по управлению качеством эксплуатации продукции; организации проведения и осуществлению работ по управлению качеством процессов производства продукции и оказания услуг; организации проведения и осуществлению работ по управлению качеством проектирования продукции и услуг; в осуществлении процессов инновационных преобразований; в информационном, технологическом, финансовом и нормативно-правовом обеспечении инновационной деятельности; в инновационном предпринимательстве.

Лабораторные работы обеспечивают учащимся возможность получить профессиональные практические навыки, в том числе исследовательского характера и закрепить знания, полученные в лекционной части дисциплины «Алгоритмические языки и программирование».

Общие требования

Лабораторные работы выполняются студентами очной формы обучения индивидуально под контролем со стороны преподавателя. Все консультации осуществляются преподавателем. Число студентов, одновременно присутствующих на занятии не должно превышать 12 человек. Если в списочном составе группы студентов больше 12, то группа должна быть разделена на подгруппы численностью от 6 до 12 человек в каждой.

Для выполнения лабораторных работ целесообразно в учебном расписании выделять 4 академических часа подряд, без больших перерывов. Расписание также должно предусматривать раздельное проведение занятий у подгрупп, если группа была разделена.

Перед началом занятий студенты должны изучить инструкцию по охране труда, действующую в лаборатории, и в дальнейшем строго выполнять ее требования. Преподаватель должен убедиться в знании инструкции, задавая студенту вопросы по ее содержанию, после чего сделать соответствующую запись в журнале охраны труда.

Во время проведения лабораторных занятий в аудитории (лаборатории) студентам запрещается передавать друг другу файлы и другие материалы, являющиеся результатом выполнения заданий.

Студент имеет право:

- Выходить из аудитории (лаборатории) не спрашивая разрешения у преподавателя.
- Самостоятельно распределять аудиторное время, определяя необходимость перерыва или непрерывной работы.
- Просить консультации у преподавателя, если он в текущий момент не распределяет задания, не принимает выполненные работы и не консультирует другого студента.

Преподаватель, давая консультацию студенту, указывает раздел технической документации или методической литературы, в которой имеется ответ на вопрос

студента. Если необходимые сведения в документации и литературе отсутствуют, то преподаватель должен дать устные пояснения или продемонстрировать практические действия, приводящие к требуемому результату, с последующей отменой для повторения студентом.

Самостоятельная работа студентов над лабораторными заданиями, связанными с техническими измерениями, с использованием электронных устройств, приборов, другой техники, может осуществляться в той же аудитории (лаборатории), где проводятся лабораторные занятия. В случае компьютерных лабораторных работ разрешается домашняя самостоятельная работа по материалам, предоставленным преподавателем. Преподаватель должен согласовать со студентами расписание самостоятельной работы - не менее 2 академических часов в неделю. В указанное время по учебному расписанию студентов и в аудитории (лаборатории) не должны проводиться другие занятия. Преподаватель должен обеспечить доступ студентов в аудиторию (лабораторию) в указанные часы.

Консультации, выдача лабораторных заданий и прием результатов выполнения осуществляется только во время аудиторных занятий. Задания выполняются последовательно. Правильное выполнение некоторых заданий возможно только, если студент корректно выполнил предыдущие задания. Поэтому приступать к следующему заданию студент может, только сдав преподавателю результат выполнения предыдущего.

Материально-техническое обеспечение лабораторных работ

Учебная аудитория:

учебная аудитория для проведения занятий лекционного типа, учебная аудитория для проведения занятий практического типа, учебная аудитория для проведения занятий семинарского типа, помещение для проведения групповых и индивидуальных консультаций.

634034, Томская область, г. Томск, Вершинина улица, д. 74, 220 ауд.

Описание имеющегося оборудования:

- Нетбук Lenovo ideaPad S10-3;
- Компьютер;
- Проектор Nec v260x;
- Экран проекторный;
- Доска маркерная;
- Компьютер (13 шт.);
- Комплект специализированной учебной мебели;
- Рабочее место преподавателя.

Программное обеспечение:

- Microsoft Windows 7 Pro
- OpenOffice.

Размещение и освещенность рабочих мест в учебной аудитории (лаборатории) должно удовлетворять действующим требованиям санитарных правил и норм (СанПиН).

Прием результатов выполнения лабораторных работ

Результаты выполнения лабораторных работ представляются преподавателю в виде отчета, содержащего функциональную и структурную схему созданной системы управления, запрограммированные алгоритмы работы, результаты испытаний, графики полученных закономерностей и зависимостей физических величин, файлы проектов, выполненных по проектной методологии PMI/PMBOK в письменном и/или электронном виде.

Во время приема выполненной работы преподаватель вправе:

- Требовать демонстрации работы созданной системы.

- Демонстрировать работу с лабораторной установкой, с созданной системой, с выполненным программным проектом.
- Самостоятельно производить манипуляции с программным обеспечением без его изменения, если оно разработано в ходе лабораторной работы.
- Требовать у студента пояснений по алгоритмам работы и способам взаимодействия элементов, по организации и назначению работ по проекту, по ресурсной модели и по результатам проекта с критическим анализом и выводами.

Задание считается выполненным и принимается преподавателем только в том случае, если реализованы все задачи, предусмотренные заданием. Если эти условия не выполняются, то результат выполнения подлежит доработке. Студент должен работать над заданием максимально самостоятельно, использовать все предусмотренные в лабораторной работе средства.

До конца семестра студент должен сдать результаты выполнения всех лабораторных работ, предусмотренных настоящими указаниями. В противном случае студент к сдаче зачета не допускается.

Темы лабораторных работ

1. ЛР1 – Основы языка Python. Установка и запуск веб-фреймворка Django. Исследование административного интерфейса Django.

Трудоемкость – 4 часа.

Цель работы: Изучить основы языка Python. Установить и запустить веб-фреймворк Django. Исследовать административный интерфейс Django.

Теоретический материал для выполнения этой лабораторной работы в лекции «Клиент-серверные приложения и основы языка Python» и «Описание Django и использование команд из django-admin.py», а также в главах 1,2 учебного пособия: Титков А.В. Создание веб-приложений / А.В. Титков, С.А. Черепанов. - 2014.

Исходные данные: задание на ЛР предоставляется преподавателем.

Задачи:

1. С официального сайта скачайте и установите интерпретатор языка Python. Проверьте его работоспособность.

2. Создайте файл с текстом `print «Hello, world!»`. Запустите Python и выведите на экран это сообщение.

3. Создайте новый файл и скопируйте в него следующий код программы:

```
#coding:utf-8
groupmates = [
    {
        "name": u"Василий",
        "group": "912-2",
        "age": 19,
        "marks": [4, 3, 5, 5, 4]
    },
    {
        "name": u"Анна",
        "group": "912-1",
        "age": 18,
        "marks": [3, 2, 3, 4, 3]
    },
]
def print_students(students):
    print u"Имя студента".ljust(15), \
          u"Группа".ljust(8), \
```

```

    u"Возраст".ljust(8), \
    u"Оценки".ljust(20)
for student in students:
    print student["name"].ljust(15), \
          student["group"].ljust(8), \
          str(student["age"]).ljust(8), \
          str(student["marks"]).ljust(20)
print "\n"
print_students(groupmates)

```

4. Запустите Python и выведите на экран форматированный список студентов.
5. Напишите функцию фильтрации студентов по средней оценке (так чтобы функция возвращала всех студентов выше заданного в параметрах функции среднего балла).
6. С официального сайта проекта django скачайте дистрибутив фреймворка Django и установите его. Проверьте работоспособность Django.
7. Создайте первый проект на Django.
8. С помощью команды «python manage.py runserver» запустите сервер разработки и посмотрите созданное приложение в действии в окне браузера.
9. Сконфигурируйте Django для входа в административное приложение.
10. Изучите интерфейс административного приложения Django.
11. Через интерфейс административного приложения создайте нового пользователя с правами суперюзера и нового пользователя без прав суперюзера, сделайте пользователя «неактивным».

Отчет о работе должен включать:

1. Цель и постановку задачи.
 2. Ход работы.
- Результаты выполнения программы (разрешается вставить в отчет скриншоты экрана).
3. Заключение и выводы.

2. ЛР2 – Создание веб-страницы с простым текстом. Создание html-шаблона. Настройка обработки статичных файлов для Django.

Трудоемкость – 4 часа.

Цель работы: Создать веб-страницу с простым текстом. Создать html-шаблона. Настроить обработку статичных файлов для Django.

Теоретический материал для выполнения этой лабораторной работы приведен в лекции «Работа с шаблонами в Django», а также в главе 3 учебного пособия: Титков А.В. Создание веб-приложений / А.В. Титков, С.А. Черепанов. - 2014.

Исходные данные: задание на ЛР предоставляется преподавателем.

Задачи:

1. Создайте новый проект на Django. Создайте внутри проекта новое приложение flatpages.

2. Проведите конфигурирование проекта Django.

3. Создайте новый адрес в файле urls.py:

```

urlpatterns = patterns("",
    url(r'^$', 'flatpages.views.home', name='home'),
    url(r'^admin/', include(admin.site.urls)),
)

```

4. В файле flatpages/views.py создайте функцию home:

```

#coding:utf-8
from django.http import HttpResponse
def home(request):

```

```
return HttpResponse(u'Привет, Мир!', mimetype="text/plain")
```

5. В браузере на страницу по адресу `http://127.0.0.1:8000/` посмотрите результат.

6. Сделайте так, чтобы по адресу `http://127.0.0.1:8000/hello/` тоже возвращался тот же самый текст.

7. Создайте папку `templates` внутри вашего приложения `flatpages`, внутри которого создайте файл `index.html`. Добавьте внутрь этого файла html-код.

8. Подключите html-файл к функции-представлению в файле `views.py`.

9. В html-файл внутрь тега `<head>` вставьте тег подключения css-скрипта `<link rel="stylesheet" href="{{ STATIC_URL }}css/index.css">`

10. Создайте css-файл и укажите в нем несколько стилей.

11. Произведите изменения в каскадной таблице стилей так, чтобы; для заголовка первого уровня был установлен шрифт с засечками, высота картинка стала 30px, ширина таблицы занимала 100% экрана.

Отчет о работе должен включать:

1. Цель и постановку задачи.

2. Ход работы.

Результаты выполнения программы (разрешается вставить в отчет скриншоты экрана).

3. Заключение и выводы.

3. ЛР3 – Создание первой модели данных и её регистрация в административном приложении Django. Динамическое генерирование шаблона для вывода экземпляров модели.

Трудоемкость – 4 часа.

Цель работы: Создать первую модель данных и зарегистрировать её в административном приложении Django. Научиться динамически генерировать шаблоны для вывода экземпляров модели.

Теоретический материал для выполнения этой лабораторной работы приведен в лекции «Модели, представления и конфигурация URL в Django», а также в главе 5 учебного пособия: Титков А.В. Создание веб-приложений / А.В. Титков, С.А. Черепанов. - 2014.

Исходные данные: задание на ЛР предоставляется преподавателем.

Задачи:

1. Создайте новый проект на Django. Создайте внутри проекта новое приложение `blog`. В этом приложении будет выводиться блог статей.

2. Проведите конфигурирование проекта Django.

3. Откройте файл `articles/models.py`, в котором будут храниться будущие модели статей. Напишите в этом файле следующее:

```
from django.db import models
from django.contrib.auth.models import User
class Article(models.Model):
    title = models.CharField(max_length=200)
    author = models.ForeignKey(User)
    text = models.TextField()
    created_date = models.DateField(auto_now_add=True)
def __unicode__(self):
    return "%s: %s" % (self.author.username, self.title)
def get_excerpt(self):
    return self.text[:140] + if len(self.text) > 140 else self.text
```

4. Создайте несколько новых статей через административный интерфейс.

5. Создайте файл `html`, в который включите следующий код для динамического генерирования вывода всех экземпляров модели:

```
{% for post in posts %}
  <div class="one-post">
    <h2 class="post-title">{{ post.title }}</h2>
    <div class="article-info">
      <div class="article-author">{{ post.author.username }}</div>
      <div class="article-created-date">{{ post.created_date }}</div>
    </div>
    <p class="article-text">{{ post.get_excerpt }}</p>
  </div>
{% endfor %}
```

6. В файле `urls.py` настройте `url`, по которому будут отображаться все статьи проекта.

Отчет о работе должен включать:

1. Цель и постановку задачи.
2. Ход работы.

Результаты выполнения программы (разрешается вставить в отчет скриншоты экрана).

3. Заключение и выводы.

4. ЛР4 – Создание страницы определенной записи. Использование CSS для верстки страниц.

Трудоемкость – 4 часа.

Цель работы: Создать страницу определенной записи. Изучить использование каскадных таблиц стилей CSS для верстки страниц.

Теоретический материал для выполнения этой лабораторной работы приведен в лекции «Статичные файлы в Django-проектах и работа с CSS», а также в главе 4 учебного пособия: Титков А.В. Создание веб-приложений / А.В. Титков, С.А. Черепанов. - 2014.

Исходные данные: задание на ЛР предоставляется преподавателем; проект, разработанный в предыдущей лабораторной работе.

Задачи:

1. Сконфигурируйте `url` в файле `urls.py`:

```
url( r'^article/(?P<article_id>\d+)$', 'articles.views.get_article', name='get_article' ),
```

2. В файле `articles/views.py` создайте новую функцию:

```
from django.http import Http404
```

```
def get_article(request, article_id):
```

```
    try:
```

```
        post = Article.objects.get(id=article_id)
```

```
        return render(request, 'article.html', {"post": post})
```

```
    except Article.DoesNotExist:
```

```
        raise Http404
```

3. Создайте шаблон для вывода страницы определенной записи.

4. На странице списка постов каждый заголовок сделать ссылкой. Сделайте так, чтобы при клике по названию происходил переход на страницу указанной определенной записи.

5. Произведите верстку обеих страниц в соответствии с макетом, для чего создайте и отредактируйте файл `archive.css`.

Отчет о работе должен включать:

1. Цель и постановку задачи.
2. Ход работы.

Результаты выполнения программы (разрешается вставить в отчет скриншоты экрана).

3. Заключение и выводы.

5. ЛР5 – Создание формы и представления для нового поста.

Трудоемкость – 4 часа.

Цель работы: Создать форму, представления и шаблон для нового поста.

Теоретический материал для выполнения этой лабораторной работы приведен в лекции «Формы в Django», а также в главе 6.1 учебного пособия: Титков А.В. Создание веб-приложений / А.В. Титков, С.А. Черепанов. - 2014.

Исходные данные: задание на ЛР предоставляется преподавателем; проект, разработанный в предыдущей лабораторной работе.

Задачи:

1. Создайте функцию-представление для ввода нового поста:

```
from django.shortcuts import render, redirect
if request.method == "POST":
    form = {
        'title': request.POST["title"]
        'text': request.POST["text"],
    }
    if form["text"] and form["title"]:
        Article.objects.create (text=form["text"],
                                title=form["title"], author=request.user)
        article = Article.objects.get(title=form['title'])
        return redirect('get_article', article_id=article.id)
    else:
        form['errors'] = u"Не все поля заполнены"
        return render(request, 'create_post.html', {'form': form})
else:
    return render(request, 'create_post.html', {})
```

2. Создайте html шаблон:

```
<html>
<head>
    <title>Создать статью</title >
</head>
<body>
    <div class="content">
        <h1>Написать статью</h1>
        <form method="POST"> {% csrf_token %}
            <input type="text" name="title" placeholder="Название статьи"
                value="{{ form.title }}">
            <textarea name="text" placeholder="Текст статьи">
                {{ form.text }} </textarea>
            <input type="submit" value="Сохранить">
        </form>
        {{ form.errors }}
    </div>
</body>
</html>
```

3. Используйте методы запроса GET и POST.
4. Создайте стили, подключив css-файл к шаблону, и оформите новую страницу аналогично предыдущим.
5. Добавьте проверку на то, что введенное для нового поста название уникально. Отчет о работе должен включать:
 1. Цель и постановку задачи.
 2. Ход работы.
 Результаты выполнения программы (разрешается вставить в отчет скриншоты экрана).
3. Заключение и выводы.

6. ЛР6 – Создание формы, шаблона и представления для авторизации и регистрации пользователей.

Трудоемкость – 4 часа.

Цель работы: Создать форму, представление и шаблон для авторизации и регистрации пользователей.

Теоретический материал для выполнения этой лабораторной работы приведен в лекции «Формы в Django. Система авторизации и регистрации», а также в главе 6 учебного пособия: Титков А.В. Создание веб-приложений / А.В. Титков, С.А. Черепанов. - 2014.

Исходные данные: задание на ЛР предоставляется преподавателем; проект, разработанный в предыдущей лабораторной работе.

Задачи:

1. Создайте шаблон и настройте адрес для отображения страницы регистрации.
 2. Создайте представление, которое обрабатывает поступающие запросы и регистрирует новых пользователей. Не забудьте сделать проверку на то, что отправленные поля не пусты, а введенное имя пользователя уникально.
 3. Сверстайте страницу и оформите ее аналогично предыдущим.
 4. Добавьте в шапку страниц всех записей и страниц для определенных статей ссылку на регистрацию в верхнем правом углу (стиль ссылке сделать точно такой же, как у ссылки “Все статьи” на собственных страницах постов).
 5. Создайте шаблон и настройте адрес для отображения страницы авторизации.
 6. Создайте представление, которое обрабатывает поступающие запросы и авторизует пользователей. Не забудьте сделать проверку на то, что отправленные поля не пусты, логин и пароль соответствуют одному из аккаунтов вашего проекта.
 7. Сверстайте страницу и оформите ее аналогично предыдущим.
- Отчет о работе должен включать:
1. Цель и постановку задачи.
 2. Ход работы.
- Результаты выполнения программы (разрешается вставить в отчет скриншоты экрана).
3. Заключение и выводы.

7. ЛР7 – Изучение основ JavaScript, создание простейших функций и использование базовых операторов.

Трудоемкость – 4 часа.

Цель работы: Изучить основы JavaScript, научиться создавать простейшие функции на JavaScript и использовать базовые операторы.

Теоретический материал для выполнения этой лабораторной работы приведен в лекции «Основы языка сценариев JavaScript», а также в главе 7 учебного пособия: Титков А.В. Создание веб-приложений / А.В. Титков, С.А. Черепанов. - 2014.

Исходные данные: задание на ЛР предоставляется преподавателем; проект разработанный в предыдущей лабораторной работе.

Задачи:

1. Создайте файл helloworld.js в папке js. Подключите файл helloworld.js к html-шаблону archive.html.

```
<script src="{{ STATIC_URL }}js/helloworld.js"></script>
```

3. Создайте список студентов (аналогично ЛР № 1) и функцию, которая будет выводить в виде таблицы содержимое этого списка:

```
var printStudents = function(students){
    console.log(
        grad("Имя студента", 15),
        grad("Группа", 8),
        grad("Возраст", 8),
        grad("Оценки", 20)
    );
    // был выведен заголовок таблицы
    for (var i = 0; i<=students.length-1; i++){
        console.log(
            grad(students[i]['name'], 15),
            grad(students[i]['group'], 8),
            grad(students[i]['age'], 8),
            grad(students[i]['marks'], 20)
        );
    }
    console.log("\n"); // добавляется пустая строка в конце вывода
};
printStudents(groupmates);
```

4. На языке JavaScript напишите функцию, которая фильтрует студентов по группе. Функция должна возвращать только тех студентов, которые учатся в указанной группе.

Отчет о работе должен включать:

1. Цель и постановку задачи.
2. Ход работы.

Результаты выполнения программы (разрешается вставить в отчет скриншоты экрана).

3. Заключение и выводы.

8. ЛР8 – Работа с элементами DOM с помощью JavaScript. Добавление обработчиков на пользовательские события.

Трудоемкость – 4 часа.

Цель работы: Изучить работу с элементами DOM с помощью JavaScript. Научиться добавлять обработчики на пользовательские события.

Теоретический материал для выполнения этой лабораторной работы приведен в лекции «Принципы работы с DOM при помощи JavaScript», а также в главе 8 учебного пособия: Титков А.В. Создание веб-приложений / А.В. Титков, С.А. Черепанов. - 2014.

Исходные данные: задание на ЛР предоставляется преподавателем; проект разработанный в предыдущей лабораторной работе.

Задачи:

1. Подключите файл fold-post.js к своему проекту, для этого перед закрывающим тегом </body> добавьте скрипт:

```
<script src="{{ STATIC_URL }}js/fold-post.js"></script>
```

2. Пропишите классы у тех элементов, которые будут “сворачиваться”.

3. С помощью JavaScript добавьте к вашей главной странице возможность скрыть информацию по каждому посту после нажатия на клавишу “свернуть”:

```
var foldBtns = document.getElementsByClassName("fold-button");
for (var i = 0; i < foldBtns.length; i++){
foldBtns[i].addEventListener("click", function(e) {
    if (e.target.parentElement.parentElement.classList.contains("folded")){
        e.target.innerHTML = "свернуть";
        e.target.parentElement.parentElement.classList.remove("folded");
        e.target.parentElement.parentElement.classList.add("fold");
    }
    else{
        e.target.innerHTML = "развернуть";
        e.target.parentElement.parentElement.classList.remove("fold");
        e.target.parentElement.parentElement.classList.add("folded");
    }
});
foldBtns[i].addEventListener("click", function(event) {
console.log("you clicked ", event.target);
});
}
```

4. Через css-стили установите исчезновение элементов поста.

Отчет о работе должен включать:

1. Цель и постановку задачи.

2. Ход работы.

Результаты выполнения программы (разрешается вставить в отчет скриншоты экрана).

3. Заключение и выводы.

9. ЛР9 – Изучение библиотеки jQuery, добавление эффектов на HTML страницу.

Трудоемкость – 4 часа.

Цель работы: Изучить возможности библиотеки jQuery, научиться добавлять эффекты на HTML страницу.

Теоретический материал для выполнения этой лабораторной работы приведен в лекции «Основы языка сценариев JavaScript», а также в главе 9 учебного пособия: Титков А.В. Создание веб-приложений / А.В. Титков, С.А. Черепанов. - 2014.

Исходные данные: задание на ЛР предоставляется преподавателем; проект разработанный в предыдущей лабораторной работе.

Задачи:

1. С официального сайта <http://jquery.com/> скачайте и установите библиотеку jQuery.

2. Создайте эффект подсветки поста, для чего в файл highlight-post.js введите:

```
$(document).ready(function(){
    $('.one-post').hover(function(event){
        $(event.currentTarget).find('.one-post-shadow').animate({opacity:'0.1'}, 300); },
    function(event){
        $(event.currentTarget).find('.one-post-shadow').animate({opacity: '0'},300);
    })
});
```

3. Создайте эффект изменения размера картинки, для чего в файл picture.js введите:

```
$(document).ready(function(){
    $('.header').hover(function(event){
```

```

$(event.currentTarget).find('.picture').animate({width:'320px'}, 300);
}, function(event){
$(event.currentTarget).find('.picture').animate({width: '300px'},300);
})
});

```

4. Ширина картинки должна становиться больше на 20px, а высота увеличиваться пропорционально.

Отчет о работе должен включать:

1. Цель и постановку задачи.
2. Ход работы.

Результаты выполнения программы (разрешается вставить в отчет скриншоты экрана).

3. Заключение и выводы.

Оформление отчетов по лабораторным работам

Отчет по лабораторной работе должен включать:

1. Титульный лист, оформленный в соответствии с приложением А.
2. Введение, в котором указывается цель работы, схема лабораторной установки и описываются полученные исходные данные.
3. Ход работы, в которой описывается выполнение каждой задачи.
4. Заключение и выводы.

В целях завершения лабораторной работы в аудитории по решению преподавателя допускается сдача аккуратно оформленного рукописного отчета, включая титульный лист, со вставкой и вклейкой скриншотов, прочих рисунков и изображений графиков.

Учебно-методическое и информационное обеспечение дисциплины

Основная литература

1. Моррисон М. HTML и XML: Практические знания необходимые для самостоятельного создания веб-страниц: Пер. с англ. / М. Моррисон; пер. К. Коваль, пер. А. Кузнецов. – СПб.: Питер, 2005. – 302 с.: ил. (наличие в библиотеке ТУСУР - 20 экз.)
2. Баранов Д.В. Построение эффективного взаимодействия с web-сайтом. HTML. CSS: Учебное пособие / Д.В. Баранов; Министерство образования Российской Федерации, Томский государственный университет систем управления и радиоэлектроники, Институт дополнительного образования. – Томск: ТУСУР, 2004. – 291 с.: ил. (наличие в библиотеке ТУСУР - 14 экз.)
3. Дунаев В.В. Самоучитель JavaScript: самоучитель / В.В. Дунаев. – 2-е изд. – СПб.: Питер, 2005. – 394 с.: ил. (наличие в библиотеке ТУСУР - 20 экз.)

Дополнительная литература

1. Губин И.Г. Технология создания интернет-приложений: учебное пособие: в 4 разделах / И.Г. Губин; Федеральное агентство по образованию, Томский государственный университет систем управления и радиоэлектроники, Кафедра компьютерных систем в управлении и проектировании. – Томск: ТМЦДО, 2007. Раздел 3: Основы PHP и MySQL. – Томск: ТМЦДО, 2007. – 144 с.: ил. (наличие в библиотеке ТУСУР - 8 экз.)
2. Губин И.Г. Технология создания интернет-приложений: учебное пособие: в 4 разделах / И.Г. Губин; Федеральное агентство по образованию, Томский государственный университет систем управления и радиоэлектроники, Кафедра компьютерных систем в управлении и проектировании. – Томск: ТМЦДО, 2007. Раздел 4: Основы PHP и MySQL. – Томск: ТМЦДО, 2007. – 142 с.: ил., табл. (наличие в библиотеке ТУСУР - 8 экз.)

Профессиональные базы данных и информационные справочные системы

1. При изучении дисциплины рекомендуется обращаться к базам данных, информационно-справочным и поисковым системам, к которым у ТУСУРа открыт доступ: <https://lib.tusur.ru/ru/resursy/bazy-dannyh>

Приложение А

Образец титульного листа отчета по лабораторным работам

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Факультет Инновационных технологий

Кафедра управления инновациями

ОТЧЁТ

по лабораторной работе по дисциплине

Алгоритмические языки и программирование

Тема лабораторной работы

Студент гр. 0 ____

_____ И.О. Фамилия

« ____ » _____ 201_ г.

Преподаватель

Должность, ученая степень

_____ И.О. Фамилия

« ____ » _____ 201_ г.

_____ оценка