


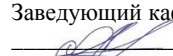


МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования

**«Дальневосточный федеральный университет»
(ДФУ)**

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ (ИКОТ)

СОГЛАСОВАНО
Руководитель ОП

(подпись) Чеботарев А.Ю.
(ФИО)

УТВЕРЖДАЮ
Заведующий кафедрой

(подпись) Чеботарев А.Ю.
(ФИО.)
«15» июля 2021 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Web-программирование

Направление подготовки 01.03.02 Прикладная математика и информатика

(Системное программирование)

Форма подготовки очная

курс 2 семестр 3

лекции 0 час.

практические занятия 0 час.

лабораторные работы 32 час.

в том числе с использованием МАО лек. 14 / пр. 0 / лаб. 14 час.

всего часов аудиторной нагрузки 32 час.

в том числе с использованием МАО 28 час.

самостоятельная работа 40 час.

в том числе на подготовку к экзамену 36 час.

контрольные работы (количество) 1

курсовая работа / курсовой проект - / ±

зачет не предусмотрен

экзамен 3 семестр

Рабочая программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта высшего образования по направлению подготовки 01.03.02 Прикладная математика и информатика, утвержденного приказом Министерства образования и науки Российской Федерации от 10 января 2018 г. № 9 (с изменениями и дополнениями).

Рабочая программа обсуждена на заседании кафедры информатики, математического и компьютерного моделирования протокол № 10 от «09» июля 2021 г.

Заведующий кафедрой Чеботарев А.Ю.

Составители: Сущенко А.А.

Владивосток
2021

Оборотная сторона титульного листа РПД

I. Рабочая программа пересмотрена на заседании кафедры/департамента:

Протокол от «_____» _____ 20__ г. № _____

Заведующий *кафедрой* _____
(подпись) (И.О. Фамилия)

II. Рабочая программа пересмотрена на заседании кафедры/департамента:

Протокол от «_____» _____ 20__ г. № _____

Заведующий *кафедрой* _____
(подпись) (И.О. Фамилия)

III. Рабочая программа пересмотрена на заседании кафедры/департамента:

Протокол от «_____» _____ 20__ г. № _____

Заведующий *кафедрой* _____
(подпись) (И.О. Фамилия)

IV. Рабочая программа пересмотрена на заседании кафедры/департамента:

Протокол от «_____» _____ 20__ г. № _____

Заведующий *кафедрой* _____
(подпись) (И.О. Фамилия)

1. Цели и задачи освоения дисциплины:

Цель: является освоение современных интернет-технологий и сопутствующих областей знаний, методов и средств создания web-ресурсов, их продвижения и применения в различных видах деятельности

Задачи:

- Дать целостное представление о возможностях и структуре глобальной сети Internet.
- Дать представление о развитии и применении Internet технологий в профессиональной деятельности.
- Изучить методы и средства разработки web-приложений;
- Сформировать навыки практической работы по созданию сайтов.
- Развить логическое и алгоритмическое мышление.

Для успешного изучения дисциплины «Web-программирование» у обучающихся должны быть сформированы следующие предварительные компетенции:

- УК-2. Способен определять круг задач в рамках поставленной задачи и выбирать эффективные методы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений

- ОПК-1 способен применять теоретические знания, полученные в области математических и естественных наук, и применять их в профессиональной деятельности

- ОПК-4 способен решать задачи профессиональной деятельности с применением существующих информационно-коммуникационных технологий и с учетом основных требований информационной безопасности

Профессиональные компетенции выпускников и индикаторы их достижения:

Тип задач	Код и наименование профессиональной компетенции (результат освоения)	Код и наименование индикатора достижения компетенции
Организационно-управленческий	ПК-7 Способен составлять и контролировать план выполняемой работы, планировать необходимые для выполнения работы ресурсы, оценивать результаты собственной работы	ПК-7.1 использует принципы разработки и отладки программного кода, методы тестирования программного обеспечения
		ПК-7.2 использует методы по выявлению и устранению сбоев и отказов в работе программного обеспечения, ликвидации их последствия и восстановления работоспособности
		ПК-7.3 разрабатывает программный код, проводит его отладку и тестирование, своевременно принимает меры по

Тип задач	Код и наименование профессиональной компетенции (результат освоения)	Код и наименование индикатора достижения компетенции
		выявлению и устранению сбоев и отказов в работе программного обеспечения, ликвидации их последствий и восстановлению работоспособности
Организационно-управленческий	ПК-8 Способен к формированию технической отчетной документации и разработке технических документов	ПК-8.1 определяет основные стандарты, нормы и правила разработки технической документации программных продуктов и программных комплексов
		ПК-8.2 составляет техническую документацию на различных этапах жизненного цикла информационных систем, программных продуктов
		ПК-8.3 разрабатывает тестовые документы, включая план тестирования, стратегии тестирования и управления процессом тестирования

Код и наименование индикатора достижения компетенции	Наименование показателя оценивания (результата обучения по дисциплине)
ПК-7.1 использует принципы разработки и отладки программного кода, методы тестирования программного обеспечения	Знает основы составления планов работы с учетом ресурсов.
ПК-7.2 использует методы по выявлению и устранению сбоев и отказов в работе программного обеспечения, ликвидации их последствия и восстановления работоспособности	Умеет составлять и контролировать план выполняемой работы.
ПК-7.3 разрабатывает программный код, проводит его отладку и тестирование, своевременно принимает меры по выявлению и устранению сбоев и отказов в работе программного обеспечения, ликвидации их последствий и восстановлению работоспособности	Владеет способностью составлять и контролировать план выполняемой работы, планировать необходимые для выполнения работы ресурсы, оценивать результаты собственной работы.

Код и наименование индикатора достижения компетенции	Наименование показателя оценивания (результата обучения по дисциплине)
ПК-8.1 определяет основные стандарты, нормы и правила разработки технической документации программных продуктов и программных комплексов	Знает техническую и отчетную документацию.
ПК-8.2 составляет техническую документацию на различных этапах жизненного цикла информационных систем, программных продуктов	Умеет формировать техническую и отчетную документацию и разрабатывать технические документы.
ПК-8.3 разрабатывает тестовые документы, включая план тестирования, стратегии тестирования и управления процессом тестирования	Владеет навыками формирования технической отчетной документации.

2. Трудоёмкость дисциплины и видов учебных занятий по дисциплине

Общая трудоёмкость дисциплины составляет 3 зачётных единиц (108 академических часов).

(1 зачетная единица соответствует 36 академическим часам)

Видами учебных занятий и работы обучающегося по дисциплине могут являться:

Обозначение	Виды учебных занятий и работы обучающегося
Лек	Лекции
Лаб	Лабораторные работы
Пр	Практические занятия
ОК	Онлайн курс
СР	Самостоятельная работа обучающегося в период теоретического обучения
Контроль	Самостоятельная работа обучающегося и контактная работа обучающегося с преподавателем в период промежуточной аттестации

I. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА

Лекционные занятия (0 час.)

Не предусмотрено учебным планом.

II. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ КУРСА И САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Лабораторные работы (32 часов)

Лабораторная работа 1. Основы HTML.(2 час.)

Лабораторная работа 2. Использование DHTML для разработки сайта.(2 час.)

Лабораторная работа 3. Основы CSS.(2 час.)

Лабораторная работа 4. Использование языка JavaScript для программирования приложений на стороне клиента.(4час.)

Лабораторная работа 5. Создание сайта с применением JavaScript и модели DOM.(4 час.)

Лабораторная работа 6. Создание сайта с использованием библиотеки JQUERY. (4 час.)

Лабораторная работа 7. Программное рисование, использование CANVAS HTML5. (4 час.)

Лабораторная работа 8. Программное рисование и библиотека Mo tools (4 час.)

Лабораторная работа 9. Основы PHP.(4 час.)

Лабораторная работа 10. Разработка клиент-серверного приложения.(4 час.)

III. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ И ОНЛАЙН КУРСА ПРИ НАЛИЧИИ

№ п/п	Дата/сроки выполнения	Вид самостоятельной работы	Примерные нормы времени на выполнение	Форма контроля
1	Неделя 1,2	Лабораторная работа «Основы HTML», самостоятельные задания	4 часа	лаб. работа №1
2	Неделя 3,4	Лабораторная работа «Использование DHTML для разработки сайта», самостоятельные задания	4 часа	лаб. работа №2
3	Неделя 5,6	Лабораторная работа «Основы CSS», самостоятельные задания	4 часа	лаб. работа №3
4	Неделя 7,8	Подготовка к тесту по разделу «Инструменты создания сайтов»	4 часа	лаб. работа №4
5	Неделя 9,10	Лабораторная работа «Использование языка JavaScript для программирования приложений на стороне клиента», самостоятельные задания	4 часа	лаб. работа №5
6	Неделя 11,12	Лабораторная работа «Создание сайта с применением JavaScript и модели DOM», самостоятельные задания	4 часа	лаб. работа №6
7	Неделя 13,14	Программирование приложений на стороне клиента», самостоятельные задания	4 часа	лаб. работа №7
8	Неделя 15,16	Лабораторная работа «Создание сайта с использованием библиотеки JQUERY», самостоятельные задания	4 часа	лаб. работа №8

9	Неделя 17	Подготовка к опросу по разделу «Программирование на стороне клиента»	4 часа	Защита отчетов
10	18 неделя	Подготовка к зачету	2 часа	Зачет

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине включает в себя характеристику заданий для самостоятельной работы; требования к представлению и оформлению результатов самостоятельной работы; критерии оценки выполнения самостоятельной работы.

Характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению

Самостоятельная работа студентов состоит из подготовки практических работ, работы над рекомендованной литературой. При подготовке практических работ необходимо сначала прочитать основные понятия по теме. При выполнении задания нужно сначала понять, что требуется в задаче, какой теоретический материал нужно использовать, наметить план решения задачи. Рекомендуется использовать методические указания и материалы по курсу, электронные пособия, имеющиеся на сервере Школы естественных наук, библиотеке ДВФУ и в сети Интернет. При подготовке к экзамену нужно освоить теорию: разобрать определения всех понятий и методов, рассмотреть примеры и самостоятельно решить несколько типовых задач из каждой темы. При решении задач всегда необходимо комментировать свои действия и не забывать о содержательной интерпретации.

Требования к представлению и оформлению результатов самостоятельной работы

Результатом самостоятельной работы являются отчеты по практическим работам на выбранную по желанию студента тематику. В процессе подготовки отчетов к практическим работам у студентов развиваются навыки составления письменной документации и систематизации имеющихся знаний.

При составлении отчетов рекомендуется придерживаться следующей структуры:

1. Постановка задачи;
2. Математическая постановка задачи;
3. Описание метода решения;
4. Описание алгоритма метода;
5. Спецификация используемых функций и типов данных;
6. Описание тестов, на которых программа проходит проверку;
7. Результаты и демонстрация эксперимента.

Критерии оценки выполнения самостоятельной работы

Отчет по лабораторной работе должен полностью удовлетворять условию задачи. В случае некачественно выполненных отчетов (не соответствующих заявленным требованиям) результирующий балл за работу может быть снижен. Студент должен продемонстрировать отчетливое и свободное владение концептуально-понятийным аппаратом, научным языком

и терминологией. Наличие письменного доклада, презентации и отчета о проделанной работе, является условием получения зачета.

IV. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА

№ п/п	Контролируемые разделы / темы дисциплины	Коды и этапы формирования компетенций	Оценочные средства		
			текущий контроль	промежуточная аттестация	
1	Инструменты создания сайтов	ПК-7 ПК-8	знает	Тест (ПР-1)	Вопросы 1-24
			умеет	Лабораторная работа (ПР-6)	Задание, тип 1
			владеет	Контрольная работа (ПР-2)	
2	Программирование на стороне клиента	ПК-7 ПК-8	знает	Собеседование (УО-1)	Вопросы 1-14
			умеет	Лабораторная работа (ПР-6)	Задание, тип 2
			владеет	Контрольная работа (ПР-2)	
3	Программирование на стороне сервера	ПК-7 ПК-8	знает	Собеседование (УО-1)	Вопросы 15-30
			умеет	Лабораторная работа (ПР-6)	Задание, тип 3
			владеет	Расчетно-графическая работа (ПР-12)	

V. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература

(электронные и печатные издания)

1. Боженюк А. В. Интеллектуальные интернет-технологии : учебник для вузов / А. В. Боженюк, Э. М. Котов, А. А. Целых. - Ростов-на-Дону : Феникс, 2010. — 382 с.
<http://lib.dvfu.ru:8080/lib/item?id=chamo:419137&theme=FEFU>
2. Бурлакова Н. Н. Создание гипертекстовой среды : учебное пособие для вузов . ч. 1 . Статическое кодирование / Н. Н. Бурлакова. - Владивосток: Изд-во Дальневосточного федерального университета, 2012. — 85 с.
<http://lib.dvfu.ru:8080/lib/item?id=chamo:681561&theme=FEFU>
3. Дронов В. А. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов. — СПб.: БХВ-Петербург, 2011. — 414 с.: ил.
<http://znanium.com/bookread.php?book=351455>
4. Одиночкина, С.В. Web-программирование PHP [Электронный ресурс] : учебно-методическое пособие. — Электрон. дан. — Спб. : НИУ ИТМО

- (Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики), 2012. — 79 с.
http://e.lanbook.com/books/element.php?pl1_id=43562
5. Зудилова, Т.В. Web-программирование HTML [Электронный ресурс] : учебно-методическое пособие / Т.В. Зудилова, М.Л. Бурков. — Электрон. дан. — Спб. : НИУ ИТМО (Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики), 2012. — 70 с. — Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=40724

Дополнительная литература
(печатные и электронные издания)

1. Дунаев В.Я. Основы Web-дизайна. Учебник. БХВ-Петербург – 2013. - 479 с. <http://lib.dvfu.ru:8080/lib/item?id=chamo:702309&theme=FEFU>
2. Дунаев В.Я. HTML, скрипты и стили. Учебник. – СПб: БХВ-Петербург, 2012. - 485 с. <http://lib.dvfu.ru:8080/lib/item?id=chamo:692681&theme=FEFU>
3. Евсеев, Д.А. Web-дизайн в примерах и задачах [Текст]: учеб.пособие / Д.А. Евсеев, В.Р. Трофимов; Под. ред. В.В. Трофимова. – М.: КНОРУС, 2010. – 272 с. <http://lib.dvfu.ru:8080/lib/item?id=chamo:382740&theme=FEFU>
4. Прохоренок Н. HTML, JavaScript, PHP и MySQL джентльменский набор Web-Мастера. – СПб: БХВ-Петербург, 2008. - 622 с. <https://lib.dvfu.ru:8443/lib/item?id=chamo:380700&theme=FEFU>
5. Гоше Х.Д. HTML5. Для профессионалов / Хуан Диего Гоше. - Санкт-Петербург: Питер, 2015. – 559 с. <http://lib.dvfu.ru:8080/lib/item?id=chamo:780190&theme=FEFU>
6. Гуриков С.Р. Интернет-технологии: Учебное пособие / С.Р. Гуриков. - М.: Форум: НИЦ ИНФРА-М, 2015. - 184 с. <http://znanium.com/go.php?id=488074>

Перечень ресурсов информационно-телекоммуникационной сети
«Интернет»

1. Справочник Notepad++ - свободно распространяемый текстовый редактор с открытым исходным кодом – Режим доступа: <https://notepad-plus-plus.org/>;
2. Справочник по HTML. – Режим доступа: <https://htmlbook.ru/html>;
3. CSS справочник. – Режим доступа: <https://css.manual.ru>;
4. Справочник Javascript. – Режим доступа: <https://Javascript.ru>.

VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

1. Описание последовательности действий студента («сценарий изучения дисциплины»).

Для успешного изучения дисциплины «WEB-программирование» студенту необходимо:

1. При подготовке к лекции изучить конспект предыдущей лекции, а также при необходимости теоретический материал, представленный в системе BlackBoard. Далее ознакомиться с материалом следующей лекции (также через систему BlackBoard), вникнуть в суть изучаемой проблемы, подготовить вопросы.
2. На лекционном занятии тщательно конспектировать теоретический материал, участвовать в обсуждении, задавать вопросы.
3. При подготовке к лабораторным занятиям на основе материалов, представленных в системе BlackBoard, сначала понять задание лабораторной работы, найти теоретический материал, необходимый для работы, изучить алгоритм реализации задания, сформулировать вопросы преподавателю.
4. На лабораторном занятии сначала задать вопросы преподавателю по методике выполнения работы, затем выполнить задание «по образцу». Отправить преподавателю через систему BlackBoard выполненное задание на проверку, ответить на вопросы преподавателя. Прежде, чем приступить к выполнению самостоятельных заданий, обдумать алгоритм их реализации, задать вопросы преподавателю по сути заданий, спланировать их выполнение.
5. Самостоятельные задания лабораторной работы можно выполнять как на аудиторном занятии, так и самостоятельно во внеаудиторное время. При этом результат их реализации необходимо проверять через систему BlackBoard и только после этого отправлять преподавателю на проверку.
6. В течение недели выбрать время для работы со специальной литературой в библиотеке и для занятий на компьютере.
7. Самостоятельную работу организовывать в соответствии с графиком выполнения самостоятельной работы, приведенном в приложении 1.

2. Рекомендации по использованию материалов учебно-методического комплекса. Рекомендуется использовать методические указания и материалы по курсу «WEB-программирование», размещенные в системе BlackBoard, идентификатор курса FU50709-02.03.01-IT-01.

3. Рекомендации по работе с литературой. Теоретический материал курса становится более понятным, когда дополнительно к прослушиванию лекций, использованию учебно-методического комплекса, представленного в системе BlackBoard, изучаются и книги из списка основной и дополнительной литературы. Литературу по курсу можно изучать в библиотеке, брать книгу на дом или читать ее на компьютере (если это электронный ресурс). Полезно использовать несколько учебников, однако желательно придерживаться рекомендации преподавателя по выбору книг по каждому разделу. Не рекомендуется «заучивать» материал, желательно добиться понимания изучаемой темы дисциплины, а затем использовать изученный материал для реализации программ и/или сайтов. Кроме того, очень полезно выявить тенденции развития той или иной технологии

разработки, выделить для себя направления дальнейшего изучения материала, для достижения более продвинутого уровня изучения дисциплины.

4. Рекомендации по подготовке к экзамену и зачету. Успешная подготовка к экзамену и зачету включает, с одной стороны, добросовестную работу в течение семестра, выполнение всех заданий преподавателя, а с другой – правильная организация процесса непосредственной подготовки. При подготовке к экзамену необходимо освоить теорию: разобрать определения всех понятий, повторить синтаксис и семантику изучаемых языков программирования, технологию разработки web-сайтов. Затем рассмотреть примеры и самостоятельно реализовать задания из каждой темы. При этом, если задания формулируются студентом самостоятельно, – достигается более продвинутый уровень изучения дисциплины.

Характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению

Расчетно-графическая работа

В расчетно-графической работе требуется создать web-приложение, используя стандартные инструменты разработки web-сайтов на стороне сервера и клиента. Приложение должно быть предназначено для организации работы с пользователем и решения некоторых задач выбранной предметной области. Результат работы – работающий на локальном сервере web-сайт и документация по его разработке.

Основные этапы разработки web-приложения

- 1. Разработка технического задания.** Составление документации, детально описывающей структурную и поведенческую модели проекта. Простым языком: описываем коротко, на техническом языке, как будет выглядеть и как будет работать сайт или программа.
- 2. Создание дизайна и навигации.** Детальная прорисовка и определение позиционирования, типовых структурных элементов проекта. Например: создание эскиза страницы сайта, разработка логотипа, выбор мест расположения элементов навигации, оформление шрифтов и заголовков и многое другое.
- 3. Вёрстка.** Описание типовых структурных элементов проекта на языке разметки, в соответствии с заданной спецификацией. Проще говоря: то, что разработано на 2-м этапе, необходимо привести к рабочему виду, используя разметки HTML и CSS.
- 4. Программирование.** Описание структурной и поведенческой моделей проекта на языках программирования.
- 5. Тестирование и отладка.** Выявление синтаксических и логических ошибок проекта и последующее их устранение.

Критерии оценки выполнения самостоятельной работы

Самостоятельная работа студентов включает расчетно-графическую работу, подготовку к устным опросам, тестирование и лабораторные работы.

Устные опросы

Устные опросы проводятся преподавателем по завершению изучения каждого раздела. Вопросы и задания приведены в приложении 2. Для подготовки используется основная и дополнительная литература по дисциплине «WEB-программирование», а также информация, размещенная в LMS BlackBoard.

Вопросы, возникающие в процессе подготовки, студент может задать преподавателю либо на консультациях, либо через специальное средство LMS BlackBoard.

Тестирование

Тестирование осуществляется на занятии через систему BlackBoard. Для подготовки тестов используются пробные тесты, размещенные в системе BlackBoard. Эти тесты не оцениваются преподавателем и служат элементом самоконтроля. Оба типа теста формируются на основе одной базы вопросов.

Лабораторные работы

Выполняются в соответствии с методическими указаниями к выполнению лабораторных работ, размещенных в системе BlackBoard. Выполненные работы отправляются преподавателю на проверку через LMS BlackBoard.

Требования к представлению и оформлению результатов самостоятельной работы

Результатом самостоятельной работы студентов являются:

1. Расчетно-графическая работа – web-сайт, реализованный стандартными средствами разработки web-приложений и оформленный в соответствии со стандартами. Созданные сайты обязательно отсылаются преподавателю через систему BlackBoard. Защита расчетно-графического задания осуществляется только после того, как приложение проверено и оценено преподавателем в системе BlackBoard.

2. Тесты самоконтроля, выполненные в системе BlackBoard. Все попытки, выполненные пользователем, сохраняются. К основному тесту студент допускается только после прохождения на зачетном уровне данного вида тестов.

3. Самостоятельные задания к лабораторным работам, реализованные необходимыми средствами разработки web-приложений и оформленные в соответствии со стандартами. Реализованные задания обязательно отсылаются преподавателю через систему BlackBoard. Защита лабораторной работы осуществляется только после того, как задание проверено и оценено преподавателем в системе BlackBoard.

Правила оформления HTML/CSS кода

1. Протокол.

Не указывайте протокол при включении ресурсов на страницу. Опускайте название протокола (**http:**, **https:**) в ссылках на картинки или другие медиа-ресурсы, файлы стилей или скрипты, конечно, если эти файлы доступны по обоим протоколам. Отсутствие протокола делает ссылку относительной, что предотвращает смешивание ресурсов из разных протоколов и незначительно уменьшает размер файлов.

Не рекомендуется:

```
<script src="http://www.google.com/js/gweb/analytics/autotrack.js">
</script>
```

Рекомендуется:

```
<script src="//www.google.com/js/gweb/analytics/autotrack.js"></script>
```

0. Общее форматирование

1. Отступы

Всегда используйте для отступа два пробела. Не используйте табуляцию и не смешивайте табуляцию с пробелами.

Рекомендуется:

```
<ul>
  <li>
  <li>
</ul>
```

```
.example {
  color: blue;
}
```

0. Регистр.

Всегда пишите в нижнем регистре. Весь код должен быть написан в нижнем регистре: это относится к названиям элементов, названиям атрибутов, значениям атрибутов (кроме текста/CDATA), селекторам, свойствам и их значениям (кроме текста).

Не рекомендуется:

```
<A HREF="/">Домой </A>
```

Рекомендуется:

```

```

0. Пробелы в конце строки

Убирайте пробелы в конце строки. Пробелы в конце строк не обязательны и усложняют использование diff.

0. Общие мета правила

1. Кодировка

Используйте UTF-8 (без BOM). Убедитесь, что ваш редактор использует кодировку UTF-8 без метки порядка байтов (BOM). Указывайте кодировку в HTML шаблонах и документах с помощью

```
<meta charset="utf-8">.
```

Опускайте кодировку для css-файлов: для них UTF-8 задана по умолчанию.

0. Комментарии

По возможности поясняйте свой код, где это необходимо. Используйте комментарии, чтобы пояснить свой код: что он делает, за что отвечает, и почему используется выбранное решение.

0. Правила оформления HTML

1. Тип документа

Используйте HTML5. HTML5 (HTML синтаксис) рекомендуется для всех html-документов: `<!DOCTYPE html>`.

0. Валидность HTML

По возможности используйте валидный HTML. Используйте валидный HTML код, кроме случаев, когда использование не позволяет достичь размера файла, необходимого для нужного уровня производительности. Используйте такие инструменты как [W3C HTML validator \(англ.\)](#) чтобы проверить валидность кода.

0. Семантика

Используйте HTML так, как это было задумано. Используйте элементы (Иногда неверно называемые “тегами”) по назначению: заголовки для заголовков, **p** для абзацев, **a** для ссылок и т.д. Это облегчает чтение, редактирование и поддержку кода.

Не рекомендуется:

```
<div onclick="goToRecommendations();">All recommendations </div>
```

Рекомендуется:

```
<a href="recommendations/"> All recommendations </a>
```

0. Альтернатива для мультимедиа

Всегда указывайте альтернативное содержимое для мультимедиа. Постарайтесь указать альтернативное содержимое для мультимедиа: например для картинок, видео или анимаций, заданных с помощью **canvas**. Для картинок это осмысленный альтернативный текст (**alt**), а для видео и аудио расшифровка текста и подписи если это возможно.

Не рекомендуется:

```

```

Рекомендуется:

```

```

0. Разделение ответственности

Разделяйте структуру, оформление и поведение. Держите структуру (разметка), оформление (стили) и поведение (скрипты) отдельно и постарайтесь свести взаимодействие между ними к минимуму. Убедитесь, что документы и шаблоны содержат только HTML, и что HTML служит только для задания структуры документа. Весь код, отвечающий за оформление, перенесите в файлы стилей, а код отвечающий за поведение — в скрипты. Старайтесь сократить их пересечения к минимуму, включая в шаблоны минимальное количество файлов стилей и скриптов.

Отделение структуры от представления и поведения помогает облегчить поддержку кода. Изменение шаблонов и HTML-документов всегда занимает больше времени чем изменение файлов стилей или скриптов.

0. Атрибут *type*

Не указывайте атрибут type при подключении стилей и скриптов в документ. Не используйте атрибут **type** при подключении стилей (кроме вариантов когда используется что-то кроме CSS) и скриптов (кроме вариантов когда это не JavaScript). Указывать атрибут **type** в данном случае

не обязательно потому что HTML5 использует [text/css](#) и [text/javascript](#) по умолчанию. Это будет работать даже в старых браузерах.

Не рекомендуется:

```
<link rel="stylesheet" href="//www.google.com/css/maia.css" type="text/css">
```

Рекомендуется:

```
<link rel="stylesheet" href="//www.google.com/css/maia.css">
```

0. Правила форматирования HTML

1. Форматирование

Выделяйте новую строку для каждого блочного, табличного или списочного элемента и ставьте отступы для каждого дочернего элемента. Независимо от стилей заданных для элемента (CSS позволяет изменить поведение элемента с помощью свойства **display**), переносите каждый блочный или табличный элемент на новую строку.

Также ставьте отступы для всех элементов вложенных в блочный или табличный элемент.

0. Правила оформления CSS

6.1 Идентификаторы и названия классов

Используйте шаблонные или имеющие смысл имена классов и идентификаторы. Вместо использования шифров, или описания внешнего вида элемента, попробуйте в имени класса или идентификатора выразить смысл его создания, или дайте ему шаблонное имя...

Рекомендуется выбирать имена, отражающие сущность класса, потому что их проще понять и, скорее всего, не понадобится менять в будущем.

6.2. Названия идентификаторов и классов

Для идентификаторов и классов используйте настолько длинные имена, насколько нужно, но настолько короткие, насколько возможно. Попробуйте сформулировать, что именно должен делать данный элемент, при этом будьте кратки насколько возможно. Такое использование классов и идентификаторов вносит свой вклад в облегчение понимания и увеличение эффективности кода.

6.3. Селекторы типа

Избегайте использование имен классов или идентификаторов с селекторами типа (тега) элемента.

6.4. 0 и единицы измерения

Не указывайте единицы измерения для нулевых значений. Не указывайте единицы измерения для нулевых значений если на это нет причины.

6.5. 0 в целой части дроби

Не ставьте "0" в целой части дробных чисел. Не ставьте 0 в целой части в значениях между -1 и 1.

6.6. Кавычки в ссылках

Не используйте кавычки в ссылках. Не используйте кавычки (" ", ') с `url()`.

6.7. Шестнадцатеричные названия цветов

Используйте трехсимвольную шестнадцатеричную запись где это возможно. Трехсимвольная шестнадцатеричная запись для цветов короче и занимает меньше места.

Не рекомендуется:

```
color:#eebbcc;
```

Рекомендуется:

```
color:#ebc;
```

6.8. Разделители в классах и идентификаторах

Разделяйте слова в идентификаторах и именах классов с помощью дефиса или подчеркивания. Не используйте ничего, кроме дефиса или подчеркивания, для соединения слов и сокращений в селекторах, чтобы повысить удобство чтения и легкость понимания кода.

0. Правила форматирования CSS

1. Упорядочивание объявлений

Сортируйте объявления по алфавиту. Задавайте объявления в алфавитном порядке, чтобы получить согласованный код, с которым легко работать. При сортировке игнорируйте браузерные префиксы. При этом, если для одного свойства используются несколько браузерных префиксов, они также должны быть отсортированы (например **-moz** должен быть перед **-webkit**)

0. Отступы в блоках.

Всегда ставьте отступы для содержимого блоков. Всегда ставьте отступы для любого блочного содержимого. Например для правил внутри правил или объявлений, чтобы отобразить иерархию и облегчить понимание кода.

0. После объявлений

Ставьте точку с запятой после каждого объявления. После каждого объявления ставьте точку с запятой для согласованности кода и облегчения добавления новых свойств.

0. После названий свойств

Используйте пробелы после двоеточий в объявлениях. Всегда используйте один пробел после двоеточия (но не до) в объявлениях, для порядка в коде.

0. Отделение селектора и объявления

Отделяйте селекторы и объявления переносом строки. Начинайте каждый селектор или объявление с новой строки.

Не рекомендуется:

```
a:focus, a:active {  
  position: relative; top: 1px;  
}
```

Рекомендуется:

```
h1,
```

```
h2,
```

```
h3 {
```

```
  font-weight: normal;
```


line-height: 1.2;

}

0. Разделение правил

Разделяйте правила пустой строкой. Всегда ставьте пустую строку между правилами.

VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Для дисциплины «Web-программирование» используются следующие оценочные средства:

Аудиторные занятия по дисциплине включают лекции и лабораторные работы. Для проведения лекционных занятий необходима аудитория со следующим оборудованием:

- мультимедийный проектор Optima EX542I – 1 шт;
- аудио усилитель QVC RMX 850 – 1 шт;
- колонки – 1 шт;
- ноутбук;
- ИБП – 1 шт;
- настенный экран;
- микрофон – 1 шт.

Лабораторные работы проводятся в компьютерном классе, в котором должно быть установлено:

- 15 моноблоков Lenovo C360G-i34164G500UDK с установленным стандартным программным обеспечением, доступом к сети Интернет.
- мультимедийный проектор OptimaEX542I – 1 шт;
- настенный экран;
- аудио усилитель QVC RMX 850 – 1 шт;
- колонки – 1 шт;
- ИБП – 1 шт;
- микрофон – 1 шт.
- документ-камера.