Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 18.09.2025 17:32:11
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение
высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
Факультет информационных технологий

УТВЕРЖДАЮ
Декан факультета
«Информационные технологии»
_____ / Д.Г. Демидов /
«__» _____ 2025 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Основы веб-разработки на стороне клиента»

Направление подготовки/специальность
09.03.01 Информатика и вычислительная техника

Профиль/специализация
«Веб-технологии»

Квалификация
бакалавр

Формы обучения
Очная, заочная

Москва, 2025 г.

Разработчик(и):

к.т.н.  /В.Ю. Верещагин/

Согласовано:
Заведующий кафедрой «Инфокогнитивные технологии»,



к.т.н., доцент / Е.А. Пухова /

Содержание

1 Цели, задачи и планируемые результаты обучения по дисциплине 4
2 Место дисциплины в структуре образовательной программы 4
3 Структура и содержание дисциплины 5
3.1 Виды учебной работы и трудоемкость 5
3.2 Тематический план изучения дисциплины 5
3.3 Содержание дисциплины 6
3.4 Тематика лабораторных занятий 8
4 Учебно-методическое и информационное обеспечение 8
4.1 Нормативные документы и ГОСТы 8
4.2 Основная литература 9
4.3 Дополнительная литература 9
4.4 Электронные образовательные ресурсы 9
4.5 Лицензионное и свободно распространяемое программное обеспечение 9
4.6 Современные профессиональные базы данных и информационные справочные системы 9
5 Материально-техническое обеспечение 9
6 Методические рекомендации 10
6.1 Методические рекомендации для преподавателя по организации обучения 10
6.2 Методические указания для обучающихся по освоению дисциплины 10
7 Фонд оценочных средств 10
7.1 Методы контроля и оценивания результатов обучения 10
7.2 Шкала и критерии оценивания результатов обучения 11
7.3 Оценочные средства 12

1 Цели, задачи и планируемые результаты обучения по дисциплине

Цель дисциплины заключается в приобретении навыков использования языка javascript на стороне клиента для реализации интерактивности интерфейса и получения данных с сервера.

Задачи дисциплины: узнать основы языка, изучить темы интерактивности, асинхронности, реактивности; научиться использовать язык программирования javascript и интерфейсы, предоставляемые браузерами для решения задач на стороне клиента, таких как валидация данных формы, отправка запроса на сервер, обработка получаемого ответа и отображение полученных данных для клиента.

Обучение по дисциплине «Основы веб-разработки на стороне клиента» направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенций	Индикаторы достижения компетенции
ПК-4. Способен проводить работы по интеграции программных модулей и компонентов и проверку работоспособности выпусков программных продуктов	ИПК-4.1. Знает: методы описания алгоритмов, основные абстрактные типы данных и их реализации ИПК-4.2. Умеет: разрабатывать алгоритмы для конкретных задач, учитывая различия в представлении информационных объектов современными браузерами ИПК-4.3. Владеет: навыками работы с веб-технологиями и программирования
ПК-5. Способен осуществлять концептуальное, функциональное и логическое проектирование систем среднего и крупного масштаба и сложности	ИПК-5.1. Знает: базовые технологии разработки веб-приложения на стороне клиента ИПК-5.2. Умеет: декомпозировать функции на подфункции; алгоритмизировать деятельность ИПК-5.3. Владеет: навыками работы со средствами Internet и Web-технологий для решения задач профессиональной деятельности

2 Место дисциплины в структуре образовательной программы

Дисциплина относится к части, формируемой участниками образовательных отношений блока Б1.2 и междисциплинарно связана с поддерживающими дисциплинами: основы программирования, основы сетевых технологий, сети и системы передачи информации и последующими дисциплинами: веб-разработка на стороне клиента; веб-разработка.

3 Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 3 зачетные единицы (108 часа).

3.1 Виды учебной работы и трудоемкость (по формам обучения)

3.1.1 Очная форма обучения

№ п/п	Вид учебной работы	Количество часов	Семестры	
			2	
1	**Аудиторные занятия**	**54**	54	
	В том числе:			
1.1	Лекции	18	18	
1.2	Семинарские/практические занятия			
1.3	Лабораторные занятия	36	36	
2	**Самостоятельная работа**	**54**	54	
3	**Промежуточная аттестация**			
	Экзамен		экзамен	
	Итого:	**108**	108	

3.2 Тематический план изучения дисциплины (по формам обучения)

3.2.1 Очная форма обучения

№ п/п	Разделы/темы дисциплины	Трудоемкость, час					
		Всего	Аудиторная работа				Самостоятельная работа
			Лекции	Семинарские/ практические занятия	Лабораторные занятия	Практическая подготовка	
1	Основы javascript	4	2				2
2	Методологии в программировании	6	2				4
3	Запросы и асинхронность	6	2				4
4	Реактивность и DOM	6	2				4
5	JS переменные, операторы, функции	8	2		2		4
6	Работа с массивами и объектами	8	2		2		4
7	DOM	8	2		2		4
8	Валидация данных в форме	8	2		2		4
9	Функции конструкторы	8	2		2		4

10	Классы в javascript	8			4		4
11	Promise API	8			4		4
12	Fetch API	8			4		4
13	Работа с реактивностью	12			8		4
14	Рубежный контроль	10			6		4
Итого		**108**	**18**		**36**		**54**

3.3 Виды учебной работы и трудоемкость (по формам обучения)

3.3.1 Заочная форма обучения

№ п/п	Вид учебной работы	Количество часов	Семестры	
			3	
1	**Аудиторные занятия**	**12**	12	
	В том числе:			
1.1	Лекции	4	4	
1.2	Семинарские/практические занятия			
1.3	Лабораторные занятия	8	8	
2	**Самостоятельная работа**	**96**	96	
3	**Промежуточная аттестация**			
	Экзамен		экзамен	
	Итого:	**108**	108	

3.4 Тематический план изучения дисциплины (по формам обучения)

3.4.1 Заочная форма обучения

№ п/п	Разделы/темы дисциплины	Трудоемкость, час					
		Всего	Аудиторная работа				Самостоятельная работа
			Лекции	Семинарские/ практические занятия	Лабораторные занятия	Практическая подготовка	
1	Основы javascript	8	2				6
2	Методологии в программировании	8	2				6
3	Запросы и асинхронность	6					6
4	Реактивность и DOM	6					6
5	JS переменные, операторы, функции	8			2		6
6	Работа с массивами и объектами	8			2		6

7	DOM	8			2		6
8	Валидация данных в форме	8			2		6
9	Функции конструкторы	6					6
10	Классы в javascript	6					6
11	Promise API	6					6
12	Fetch API	6					6
13	Работа с реактивностью	6					6
14	Рубежный контроль	18					18
Итого		**108**	**4**		**8**		**96**

3.5 Содержание дисциплины

Основы javascript

Тема раскрывает основы синтаксиса языка и ментальную модель, с которой проще понимать javascript.

Методологии в программировании

Тема показывает какие методологии есть в программировании и что для их реализации есть в javascript.

Запросы и асинхронность

Тема раскрывает работу клиент-серверной архитектуры, что такое функции обратного вызова, объект Promise и Fetch, работа event loop.

Реактивность и DOM

Тема показывает в чем заключается идея реактивности и варианты реализации этой идеи в разных библиотеках и фреймворках.

JS переменные, операторы, функции

Тема дает навыки написания простого кода на javascript в виде отдельных функций.

Работа с массивами и объектами

Тема раскрывает возможности работы массивами и объектами, возможности методов и как их применять для решения задач.

DOM

Тема дает основы о том, как работать с деревом элементов в браузере, как их создавать, добавлять, управлять их свойствами.

Валидация данных в форме

Используя знания работы с деревом элементов, реализуем валидацию данных, которые пользователь вводит в форму.

Функции конструкторы

Тема раскрывает еще одну возможность функций, которые позволяют создавать новые объекты со своими методами, также разбирается работа this.

Классы в javascript

Тема показывает, как создаются класс в javascript, как одни классы могут наследоваться от других и как с помощью этого создаются абстракции.

Promise API

Тема раскрывает назначение объекта Promise и общие принципы работы с асинхронностью.

Fetch API

Тема показывает возможности Fetch API, как с его помощью отправляется запрос на сервер.

Работа с реактивностью

Тема раскрывает концепцию реактивности и возможности ее реализации в виде простого модуля.

3.6 Тематика лабораторных занятий

1. **JS переменные, операторы, функции** - JavaScript: типы данных, операторы, переменные, условное и циклическое выполнение, функции
2. **Работа с массивами и объектами -** JavaScript: массивы, объекты
3. **DOM** - JavaScript: DOM, события и их обработка
4. **Валидация данных в форме** - Валидация данных в формах
5. **Функции конструкторы** - new, this, anime.js
6. **Классы в javascript** -
7. **Promise API** - Promise
8. **Fetch API** - Fetch
9. **Работа с реактивностью** - Работа с реактивностью

4 Учебно-методическое и информационное обеспечение

4.1 Нормативные документы и ГОСТы

1. Федеральный закон от 29 декабря 2012 года № 273-ФЗ «Об образовании в Российской Федерации» (с изменениями и дополнениями);
2. Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника, утвержденный Приказом Министерства образования и науки РФ от 19 сентября 2017 г. N 929 "Об утверждении федерального… Редакция с изменениями N 1456 от 26.11.2020
3. Приказ Министерства образования и науки РФ от 05 апреля 2017 г. № 301 «Об утверждении Порядка организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры;
4. Порядок проведения государственной итоговой аттестации по образовательным программам высшего образования – программам бакалавриата, программам специалитета и программам магистратуры, утвержденный приказом Минобрнауки России от 29 июня 2015 г. № 636;
5. Положение о практической подготовке обучающихся, утвержденное приказом Министерства науки и высшего образования Российской Федерации и Министерства просвещения Российской Федерации от 5 августа 2020 г. № 885/390;
6. Устав и локальные нормативные акты Московского политеха

Области профессиональной деятельности и сферы профессиональной деятельности, в которых выпускники, освоившие программу бакалавриата (далее - выпускники), могут осуществлять профессиональную деятельность:

06 Связь, информационные и коммуникационные технологии (в сфере проектирования, разработки, внедрения и эксплуатации средств вычислительной техники и информационных систем, управления их жизненным циклом)

Выпускники могут осуществлять профессиональную деятельность в других областях и (или) сферах профессиональной деятельности при условии соответствия уровня их образования и полученных компетенций требованиям к квалификации работника, предъявляемым соответствующими профессиональными стандартами.

4.2 Основная литература

Верещагин, В.Ю. Создание веб-страниц на стороне клиента: учебно- методическое пособие для студентов бакалавриата, обучающихся по направлению подготовки 09.03.01 «Информатика и вычислительная техника», профиль «Веб-технологии» / В.Ю. Верещагин, М.В. Даньшина. – Москва: Московский Политех, 2023. – 1 CD-R. – Загл. с титул. экрана. – Текст: электронный. ISBN 978-5-2760-2786-9. — URL: https://online.mospolytech.ru/mod/data/view.php?d=127&rid=5281&filter=1

4.3 Дополнительная литература

Полуэктова, Н. Р. Разработка веб-приложений : учебное пособие для вузов / Н. Р. Полуэктова. — Москва : Издательство Юрайт, 2023. — 204 с. — (Высшее образование). — ISBN 978-5-534-13715-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/519714

4.4 Электронные образовательные ресурсы

1. Курс «Основы веб-разработки на стороне клиента» https://online.mospolytech.ru/course/view.php?id=1797

4.5 Лицензионное и свободно распространяемое программное обеспечение

1. Visual Studio Code
2. Браузеры Chrome, Edge, Firefox
3. OpenVPN с правами для запуска у студентов
4. FileZilla
5. PuTTy
6. Git
7. Node.js 18
8. Python 3.10
9. Wireshark

4.6 Современные профессиональные базы данных и информационные справочные системы

1. https://doka.guide/
2. https://developer.mozilla.org/ru/
3. https://roadmap.sh/frontend
4. https://learn.javascript.ru/

5 Материально-техническое обеспечение

Для проведения лабораторных работ и самостоятельной работы студентов подходят аудитории, оснащенные компьютерами с программным обеспечением в соответствии со списком в пункте 4.5 и подключенные к интернету.

Число рабочих мест в аудитории должно быть достаточным для обеспечения индивидуальной работы студентов.

Рабочее место преподавателя должно быть оснащено компьютером с подключенным к нему проектором или иным аналогичным по функциональному назначению оборудованием.

6 Методические рекомендации

6.1 Методические рекомендации для преподавателя по организации обучения

1. При подготовке к занятиям следует предварительно проработать материал занятия, предусмотрев его подачу точно в отведенное для этого время занятия. Следует подготовить необходимые материалы – теоретические сведения, задачи и др. При проведении занятия следует контролировать подачу материала и решение заданий с учетом учебного времени, отведенного для занятия.

2. При проверке работ и отчетов следует учитывать не только правильность выполнения заданий, но и оптимальность выбранных методов решения, правильность выполнения всех его шагов.

6.2 Методические указания для обучающихся по освоению дисциплины

Изучение дисциплины осуществляется в строгом соответствии с целевой установкой в тесной взаимосвязи учебным планом. Основой теоретической подготовки студентов являются лекции и самостоятельная работа.

В процессе самостоятельной работы студенты закрепляют и углубляют знания, полученные во время аудиторных занятий, готовятся к промежуточной аттестации, а также самостоятельно изучают отдельные темы учебной программы.

На занятиях студентов, в том числе предполагающих практическую деятельность, осуществляется закрепление полученных, в том числе и в процессе самостоятельной работы, знаний. Особое внимание обращается на развитие умений и навыков установления связи положений теории с профессиональной деятельностью будущего специалиста.

Самостоятельная работа осуществляется индивидуально. Контроль самостоятельной работы организуется в двух формах:

- самоконтроль и самооценка студента;
- контроль со стороны преподавателей (текущий и промежуточный).

Текущий контроль осуществляется на аудиторных занятиях.

Критериями оценки результатов самостоятельной работы студента являются:

- уровень освоения студентом учебного материала;
- умения студента использовать теоретические знания при выполнении практических задач;
- сформированность компетенций;
- оформление материала в соответствии с требованиями.

Приветствуется обсуждение самих заданий с другими студентами: можно как давать, так и получать советы по общей стратегии выполнения и изучения материала, давать и получать помощь в отладке. Однако писать код студент должен самостоятельно. Делиться кодом или писать его совместно запрещено.

7 Фонд оценочных средств

7.1 Методы контроля и оценивания результатов обучения

Приведенные ниже правила выставления оценок и опозданий могут быть изменены, если преподаватель сочтет это необходимым. Важно, чтобы студенты регулярно просматривали план курса, выложенный в СДО, на предмет его обновления или изменения.

Достижение компетенций оценивается с помощью лабораторных работ и рубежных контролей. Индикатор ИПК-4.1 заложен в темах 1, 2. ИПК-4.2 заложен в темах 7, 11, 12. ИПК-4.3 заложен в темах 5, 6, 7. Владеет: навыками работы с веб-технологиями и программирования. ИПК-5.1 заложен в темах 1, 2, 3, 4. ИПК-5.2 заложен в темах 8, 9, 10. ИПК-5.3 заложен в темах 11, 12, 13.

В соответствии с планом на дисциплины студентам выдаются задания на лабораторные работы. Помимо требований и описания функционала в работе указан крайний срок сдачи. Для сдачи лабораторной работы студенту необходимо прислать ссылку на репозиторий в GitHub Classroom и на хостинг, где размещен результат работы с реализованным функционалом, описанным в задании. Работа считается сданной если в ней реализовано 80% и более требований и функционала, описанного в задании.

Каждый студент имеет право на 6 дней опоздания, которые могут быть потрачены на любые задания в течение семестра. Опоздания предназначены для решения особых ситуаций, таких как болезнь или чрезвычайные семейные обстоятельства.

Когда использованы все дни опоздания за каждый день просрочки начисляется штраф в размере 25% от максимального результата за задание. Задания, присланные позже, чем 4 дня, не будут оцениваться. В связи с зависимостью между работами студентам может потребоваться все равно выполнить предыдущие работы, даже если они не оцениваются.

После сдачи лабораторной работы студент должен ее защитить. Во время защиты лабораторной работы преподаватель проверяет репозиторий, хостинг и выполнение критериев и требований задания, а студент отвечает на вопросы преподавателя по его коду, а также теоретических вопросов, приведенных после текста задания лабораторной работы. Если студент отказывается отвечать на вопросы, или дает полностью неверные ответы, или ответы не по теме, то работа может считаться сданной, но при этом она не оценивается.

Работа должна быть выполнена студентом самостоятельно: в репозитории в системе контроля версий студента содержатся коммиты только за его авторством, по этим коммитам можно проследить как велась работа, студент может объяснить свой код и ход выполнения работы, если эти правила не соблюдаются, то работа не считается сданной и не оценивается.

Рубежные контроли пишутся в аудитории индивидуально по варианту задания, выданному преподавателем в назначенные дни. При отсутствии студента в день написания контрольной работы ему дается еще один шанс ее написать на последнем занятии в семестре, но обязательно очно.

Студенты должны заранее сообщать о том, что у них могут возникнуть трудности со своевременной сдачей задания или проекта. При наличии реальных причин задержки студентам следует как можно скорее связаться с преподавателем и обсудить возможные условия.

7.2 Шкала и критерии оценивания результатов обучения

Лабораторная работа оценивается в процентах степени выполнения следующих критериев и для выставления оценки суммируются проценты за каждый из четырех критериев:

1. Полнота выполнения практического задания (30%): соответствует ли функциональность заданным требованиям и целям, насколько точно и без ошибок код выполняет поставленные задачи, насколько эффективно задание отвечает требованиям целевой аудитории и обеспечивает приятное восприятие.
2. Качество и структура кода (10%): качество, читаемость и организация кода, рациональность выполнения задания, последовательность именования и соблюдение лучших практик.
3. Творчество и инновации (10%): творческий подход студентов к выполнению заданий, насколько студенты вышли за рамки основных требований и реализовали дополнительные возможности или использовали уникальные решения.
4. Ответы на вопросы по коду студента и теории (50%):

Дает краткий ответ, содержащий ошибки или неточности. На наводящие вопросы отвечает неправильно (10% из 50%)

Дает развернутый ответ, содержащий ошибки или неточности. На наводящие вопросы отвечает неверно (20% из 50%)

Дает развернутый ответ, содержащий ошибки или неточности. На наводящие вопросы отвечает правильно (30% из 50%)

Дает правильные и развернутые ответы на вопросы (50% из 50%).

R лабораторные рассчитывается как среднее результатов за все лабораторные работы. За полное и безошибочное выполнение всех лабораторных работ в срок и их защиту можно получить максимум 100 баллов (R лабораторные).

Рубежный контроль оценивается по следующим критериям:

Полнота выполнения практического задания: соответствует ли функциональность заданным требованиям и целям, насколько точно и без ошибок код выполняет поставленные задачи.

Качество и структура кода: качество, читаемость и организация кода, рациональность выполнения задания, последовательность именования и соблюдение лучших практик.

Творчество и инновации: творческий подход студентов к выполнению заданий, насколько студенты вышли за рамки основных требований и реализовали дополнительные возможности или использовали уникальные решения.

Пользовательский опыт: отзывчивость, доступность, насколько эффективно задание отвечает требованиям целевой аудитории и обеспечивает приятное восприятие.

Самостоятельность решения: в репозитории студента есть коммиты только за его авторством, по коммитам в репозитории можно проследить как велась работа, студент может объяснить свой код и ход выполнения работы, если эти правила не соблюдаются, то работа не считается сданной.

Более подробное описание критериев дается в тексте задания рубежного контроля.

За полностью выполненные рубежные контроли также можно получить 100 баллов (R контроль).

Также имеется коэффициент сданных работ K сданные, который равен 1 если все работы сданы и 0 если хотя бы одна работа не сдана.

Итоговый балл рассчитывается по формуле: R сем = (0,5 x R лабораторные + 0,5 x R контроль) * K сданные.

Итоговый балл пересчитывается по шкале ниже и на основании полученной оценки фиксируется результат промежуточной аттестации.

Соответствие баллов в 100 балльной рейтинговой системе оценке по 4-бальной шкале:

0-54 - неудовлетворительно

55-69 - удовлетворительно

70-84 - хорошо

85-100 – отлично

7.3 Оценочные средства

7.3.1 Текущий контроль

Примерный список вопросов

1 Что значит “JS это слабо типизированный язык”?
2 Что значит “JS язык с динамической типизацией”?
3 Какие типы значений есть в JS?
4 Чем отличаются строки "", ” и “?
5 Что такое операнд и оператор?
6 Какие операторы есть в JS?
7 Как проверяются условия в JS?
8 Какие есть циклы в JS?
9 Как создаются функции?
10 Как хранятся данные в массиве?
11 Какие методы массивов вы знаете?

12 Как обратиться к элементу массива?
13 Как можно перебирать элементы массива?
14 Что такое объект?
15 Как получить значение из свойства объекта?
16 Для чего используется объект Math?
17 Приведите примеры свойств и методов в Math.
18 Что такое DOM?
19 Какие есть способы получения элементов из DOM?
20 Что такое событие?
21 Как добавить обработчик события?
22 Примеры свойств в объекте event.
23 Примеры методов у элементов DOM.
24 Назовите фазы жизненного цикла события.
25 На какой фазе изначально срабатывают обработчики событий?
26 Всплывают ли события blur и focus?
27 Для чего нужен preventDefault()?
28 Какие атрибуты полей форм можно использовать для добавления ограничений?
29 Почему требуется проверять данные на стороне клиента?
30 Что позволяет задать метод setCustomValidity()?
31 В какой момент показывается сообщение заданное в setCustomValidity()?
32 Что такое метод объекта?
33 Что такое функция-конструктор?
34 Для чего используется ключевое слово new?
35 Как работает и на что указывает ключевое слово this?
36 Что такое функция обратного вызова (callback)?
37 Что такое асинхронность и почему приходится ее использовать в JS?
38 Общий принцип работы event loop.
39 Promise и его состояния.
40 Методы объекта Promise. Примеры их использования.
41 Как используется async/await?
42 Суть AJAX
43 Что такое и как работает Fetch API
44 На базе чего построен Fetch API
45 Какие есть свойства и методы у объекта Response
46 Какие методы из DOM позволяют создавать и добавлять элементы
47 В чем заключается идея реактивности?
48 Что подразумевается под прогрессивным фреймворком?
49 Чем библиотека отличается от фреймворка?
50 Зачем в принципе использовать фреймворки Vue/Angular?
51 Для чего используется виртуальный DOM в react/vue?

Пример задания рубежного контроля

Главные требования

Работа выполнена самостоятельно.

1. Работа написана во время занятия.
2. В работе не использованы сторонние библиотеки и компоненты.
3. Не менее трех коммитов за авторством студента, равномерно распределенных по времени выполнения задания.
4. Работа размещена на удаленном репозитории на Github, который создан через Github Classroom. (Без этого работа не проверяется)
5. Работа размещена на хостинге (сервер Мосполитеха) или другом.

Варианты - Тематика

- Вариант 1 - Frontend
- Вариант 2 - Backend
- Вариант 3 - Тестировщик
- Вариант 4 - UX Дизайнер
- Вариант 5 - Архитектор

Материалы

Информационное наполнение

- https://roadmap.sh/

Цветовые палитры

- http://a11yrocks.com/colorPalette/
- https://clrs.cc/a11y/
- https://venngage.com/tools/accessible-color-palette-generator

Шрифты

- https://gwfh.mranftl.com/fonts

Задание

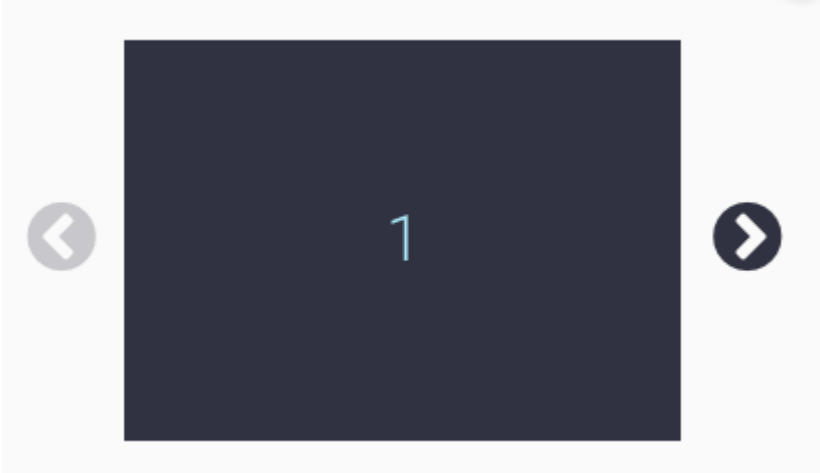
Перед началом работы создайте новую ветку в проекте. В конце работы откройте pull request на github вашей ветки с основной.

Документ index.html содержит:

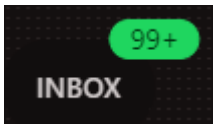
1. Header:
    1. Логотип в растровом формате (с srcset и x дескриптором).
    2. Навигационное меню (nav) с якорными ссылками на разделы страницы.
2. Основную часть:
    1. Hero-раздел с названием направления, кратким описанием, изображением и призывом к обучению. Должно занимать весь первый экран.
    2. Раздел с подзаголовком и набором из трех или более слайдов в карусели с описанием тем для изучения со ссылками:
        1. Слайды переключаются по клику на кнопки влево, вправо.
        2. Текст структурирован, семантически размечен и имеет форматирование.
        3. Добавлено изображение в каждый слайд (логотип технологии/стандарта или схема).
    3. Часть с подзаголовком и редактируемым списком:
        1. Поле для ввода текста, кнопка для добавления элемента в список.
        2. В списке отображается название элемента и кнопка для удаления элемента списка.
        3. У подзаголовка есть счетчик, который показывает количество элементов в списке.
3. Footer:
    1. ФИО, группа, дата РК.



Пример вступительной части (hero)



Пример карусели



Пример счетчика

Критерии оценки

Общие

1. Страница отзывчива (RWD + mobile-first).
2. Страница удовлетворяет критериям доступности.
3. Проект структурирован. Файлы проекта распределены по папкам.
4. Код отформатирован (легко читаем).

HTML

1. HTML валиден.
2. Использованы семантические теги.

CSS/SCSS

1. Элементы стилизованы (цвета, шрифты), логичные внутренние и внешние отступы, аккуратное выравнивание.

2. Использована методология БЭМ при именовании классов. Нет ошибок, понятные имена классов.
3. Верстка не “ломается” и данные читаются корректно при изменении размера (объема) контента.
4. Элементы страницы хорошо читаемы.
5. Подключен из папки вашего проекта и использован шрифт.
6. Изменено оформление у элементов формы.
7. Изменяется оформление элемента при наведении на него курсора.
8. Добавлен переход (transition).
9. Добавлена анимация.
10. Страница отзывчива (RWD). Адекватно отображаются при ширине экрана от 320px до 2560px:
    1. На малых экранах нет горизонтальной прокрутки и контент не прилипает к краям.
    2. На больших экранах страница ограничена по ширине и не расползается.
11. SASS/SCSS.

JavaScript

1. Реализован функционал слайдера.
2. Реализован функционал списка:
    1. Добавление нового элемента списка по нажатию на кнопку с иконкой.
    2. При добавлении нового элемента поле очищается.
    3. Удаление элемента списка с помощью кнопки в этом элементе списка.
3. Реализован функционал счетчика.
4. Отсутствуют ошибки в консоли браузера при работе пользователем со списком.

После занятия

Разместить работу на хостинге (сервер Мосполитеха) или другом, например Netlify.