Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 31.10.2022 11:12:20
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«Московский политехнический университет»

УТВЕРЖДЕНО
Декан факультета
Информационных технологий
_____ / Демидов Д.Г. /
«27» апреля 2022 г.

Рабочая программа дисциплины
«ВЕБ-РАЗРАБОТКА»

Направление подготовки:
09.03.01 Информатика и вычислительная техника

Образовательная программа (профиль):
«Интеграция и программирование в САПР»

Год начала обучения:
2022

Уровень образования:
бакалавриат

Квалификация (степень) выпускника:
Бакалавр

Форма обучения:
очная

Москва, 2022

Разработчик(и):

к.т.н., доцент / 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 для очной формы обучения 5
3.2 Тематический план изучения дисциплины для очной формы обучения 5
3.3 Содержание дисциплины 6
3.4 Тематика лабораторных занятий 6
4 Учебно-методическое и информационное обеспечение 7
4.1 Нормативные документы и ГОСТы 7
4.2 Основная литература 7
4.3 Дополнительная литература 7
4.4 Электронные образовательные ресурсы 7
4.5 Лицензионное и свободно распространяемое программное обеспечение 7
4.6 Современные профессиональные базы данных и информационные справочные системы 7
5 Материально-техническое обеспечение 8
6 Методические рекомендации 8
6.1 Методические рекомендации для преподавателя по организации обучения 8
6.2 Методические указания для обучающихся по освоению дисциплины 8
7 Фонд оценочных средств 9
7.1 Методы контроля и оценивания результатов обучения 9
7.2 Шкала и критерии оценивания результатов обучения 10
7.3 Оценочные средства 11

1 Цели, задачи и планируемые результаты обучения по дисциплине

Цель дисциплины: изучить основные элементы и понятия веб-технологий, освоить инструменты разработки и технологии необходимые для создания веб-страниц различной сложности в соответствии с техническим заданием.

Задачи дисциплины - освоение на практике:

1. формирования структуры веб-страниц с помощью HTML
2. оформления веб-страниц, используя CSS
3. добавление интерактивных элементов с использованием JavaScript
4. инструментов разработки (IDE, GIT)

Обучение по дисциплине «Веб-разработка» направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенций	**Индикаторы достижения компетенции**
ПК-1. Способен разрабатывать требования и проектировать программное обеспечение	ИПК-1.1 Знать: Возможности существующей программно-технической архитектуры Возможности современных и перспективных средств разработки программных продуктов, технических средств Методологии разработки программного обеспечения и технологии программирования Методологии и технологии проектирования и использования баз данных Методы и средства проектирования программного обеспечения Методы и средства проектирования программных интерфейсов Принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения Типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения ИПК-1.2 Уметь: Выбирать средства реализации требований к программному обеспечению Вырабатывать варианты реализации программного обеспечения Использовать существующие типовые решения и шаблоны проектирования программного обеспечения Применять методы и средства

	проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов ИПК-1.3 Владеть: Инструментами и технологиями разработки требований и проектирования программного обеспечения инструментами и технологиями разработки программного кода
ПК-2. Способен осуществлять концептуальное, функциональное и логическое проектирование систем среднего и крупного масштаба и сложности.	ИПК-2.1 Знать: Теорию тестирования Методы оценки качества программных систем Методы тестирования ИПК-2.2 Уметь: Исполнять ручные тесты
ПК-3. Способен управлять проектами в области информационных технологий на основе полученных планов проектов в условиях, когда проект не выходит за пределы утвержденных параметров.	ИПК-3.2 Уметь: Контролировать и управлять проектом в области ИТ на основе различных методологий ИПК-3.3 Владеть: Навыками сбора информации, разработки документации проекта с использованием методик и специализированного программного обеспечения.
ПК-4. Способен разрабатывать документы информационно-маркетингового назначения, разрабатывать технические документы, адресованные специалисту по информационным технологиям.	ИПК-4.2 Уметь: Преобразовывать документ в различные выходные форматы (pdf, html, формат электронной справки); применять средства подготовки слайд-шоу Составлять текст для веб-сайтов ИПК-4.3 Владеть: навыками разработки и согласования рекламного материала, статей, включающих мультимедийный и интерактивные элементы с использованием специального программного обеспечения; навыками изучения целевой аудитории, задач, потребностей в информации, уровня подготовки; навыками сбора экспертной информации по предметной области, определения и описания информационной или математической модели;

2 Место дисциплины в структуре образовательной программы

Дисциплина относится к части, формируемой участниками образовательных отношений блока Б1.2, модуля «Разработка в области информационных технологий» и междисциплинарно связана с поддерживающими дисциплинами: основы программирования, и последующими дисциплинами: веб-разработка.

3 Структура и содержание дисциплины

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

3.1 Виды учебной работы и трудоемкость для очной формы обучения

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

3.2 Тематический план изучения дисциплины для очной формы обучения

№ п/п	Разделы/темы дисциплины	Трудоемкость, час					
		Всего	Аудиторная работа				Самостоятельная работа
			Лекции	Семинарские/практические занятия	Лабораторные занятия	Практическая подготовка	
1	Разметка HTML	8			4		4
2	Основы CSS	8			4		4
3	Цвета, единицы измерения, переходы	8			4		4
4	Блочная модель	8			4		4
5	Изображения	8			4		4
6	Flexbox и Grid	8			4		4
7	Формы и ссылки	8			4		4

8	Отзывчивость и доступность	8			4		4
9	Основы JS	8			4		4
10	Объекты, массивы и их методы	8			4		4
11	DOM и события	8			4		4
12	Валидация данных в форме	8			4		4
13	Рубежный контроль	12			6		6
Итого		**108**			**54**		**54**

3.3 Содержание дисциплины

Разметка HTML

Тема раскрывает принцип разметки информации, основные теги, правила разметки, атрибуты.

Основы CSS

Тема раскрывает основные принципы применения стилей к элементам страницы.

Цвета, единицы измерения, переходы

Тема показывает способы управления цветом элементов интерфейса, их размерами и как добавить интерактивности с помощью переходов между значениями свойств.

Блочная модель

Тема раскрывает принципы расчета размеров элементов, разницу между разными видами элементов, как позиционируются элементы и способы управления этим.

Изображения

Тема раскрывает форматы изображений, особенности работы с ними и способы оптимизации.

Flexbox и Grid

Тема раскрывает возможности модуля раскладки flexbox и grid.

Формы

Тема раскрывает правила создания форм и их возможности, а также правила оформления ссылок и их состояния.

Отзывчивость и доступность

Тема раскрывает идею отзывчивых страниц и способ их достижения; необходимость учета различных сценариев использования страниц и базовых приемов для достижения доступности.

Основы javascript

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

Объекты, массивы и их методы

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

DOM и события

Тема дает основы о том, как работать с деревом элементов в браузере, как их создавать, добавлять, управлять их свойствами, а также событиями и работе с ними.

Валидация данных в форме

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

3.4 Тематика лабораторных занятий

Разметка HTML
Основы CSS
Цвета, единицы измерения, переходы
Блочная модель
Изображения
Flexbox и Grid
Формы и ссылки
Отзывчивость и доступность
Основы JS
Объекты, массивы и их методы
DOM и события
Валидация данных в форме

4 Учебно-методическое и информационное обеспечение

4.1 Нормативные документы и ГОСТы

1. Приказ Минтруда России от 18.01.2017 N 44н "Об утверждении профессионального стандарта "Разработчик Web и мультимедийных приложений" (Зарегистрировано в Минюсте России 31.01.2017 N 45481)

2. п. 1 ч. 1 ст. 43 и ст. 58 Федерального закона от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»

4.2 Основная литература

Диков, А. В. Клиентские технологии веб-программирования: JavaScript и DOM : учебное пособие / А. В. Диков. — Санкт-Петербург : Лань, 2020. — 124 с. — ISBN 978-5-8114-4074-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/126934 (дата обращения: 24.09.2022). — Режим доступа: для авториз. пользователей.

4.3 Дополнительная литература

Технологии веб-сервисов : учебно-методическое пособие / А. М. Дергачев, Ю. Д. Кореньков, И. П. Логинов, А. Г. Сафронов. — Санкт-Петербург : НИУ ИТМО, 2021. — 100 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/283676 (дата обращения: 24.09.2022). — Режим доступа: для авториз. пользователей.

4.4 Электронные образовательные ресурсы

https://online.mospolytech.ru/course/view.php?id=10704

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. Компетенции ПК-1, ПК-2, ПК-3, ПК-4 и их индикаторы заложены в темах 1-12.

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 такое HTML?
2. Что такое тег/атрибут/значение?
3. Для чего используется тег *имя тега*?
4. Зачем нужен DOCTYPE?
5. Как отличить кнопку от ссылки?
6. Что такое CSS?
7. Что такое каскад?
8. Как рассчитывается специфичность?
9. Как специфичность влияет на каскад?
10. Что подразумевает наследование и как его использовать на благо?
11. Из чего состоят правила?
12. Какие бывают селекторы?
13. Для чего нужны normalize.css или reseter.css?
14. Какие существуют 3 способа подключения стилей?
15. Как создаются Custom Properties?
16. Классификация шрифтов.
17. На что влияет свойство font-weight?
18. Почему в для свойства font-family лучше указывать несколько шрифтов?
19. Что меняет свойство color?
20. Что такое CSS?
21. Что такое каскад?
22. Как рассчитывается специфичность?
23. Как специфичность влияет на каскад?
24. Что подразумевает наследование и как его использовать на благо?
25. Из чего состоят правила?
26. Какие бывают селекторы?
27. Для чего нужны normalize.css или reseter.css?
28. Какие существуют 3 способа подключения стилей?
29. Как создаются Custom Properties?
30. Классификация шрифтов.
31. На что влияет свойство font-weight?
32. Почему в для свойства font-family лучше указывать несколько шрифтов?
33. Что меняет свойство color?
34. Как работает та или иная единица измерения?
35. Какие варианты задания цветов существуют и чем они отличаются?
36. Что такое флекс-контейнер?
37. Что такое флекс-элемент?

38. Какие свойства применяются на флекс-контейнере?
39. Какие свойства применяются на флекс-элементе?
40. Какие оси есть у флексов и какие свойства меняют их направления?
41. Как выстраиваются флекс-элементы в контейнере?
42. Какие свойства и как они выравнивают флекс-элементы?
43. Что делают свойства flex-grow, flex-shrink и flex-basis?
44. Как работает та или иная единица измерения?
45. Какие варианты задания цветов существуют и чем они отличаются?
46. Что такое флекс-контейнер?
47. Что такое флекс-элемент?
48. Какие свойства применяются на флекс-контейнере?
49. Какие свойства применяются на флекс-элементе?
50. Какие оси есть у флексов и какие свойства меняют их направления?
51. Как выстраиваются флекс-элементы в контейнере?
52. Какие свойства и как они выравнивают флекс-элементы?
53. Что делают свойства flex-grow, flex-shrink и flex-basis?
54. Как работает та или иная единица измерения?
55. Какие варианты задания цветов существуют и чем они отличаются?
56. Что такое флекс-контейнер?
57. Что такое флекс-элемент?
58. Какие свойства применяются на флекс-контейнере?
59. Какие свойства применяются на флекс-элементе?
60. Какие оси есть у флексов и какие свойства меняют их направления?
61. Как выстраиваются флекс-элементы в контейнере?
62. Какие свойства и как они выравнивают флекс-элементы?
63. Что делают свойства flex-grow, flex-shrink и flex-basis?
64. Как работает та или иная единица измерения?
65. Какие варианты задания цветов существуют и чем они отличаются?
66. Что такое флекс-контейнер?
67. Что такое флекс-элемент?
68. Какие свойства применяются на флекс-контейнере?
69. Какие свойства применяются на флекс-элементе?
70. Какие оси есть у флексов и какие свойства меняют их направления?
71. Как выстраиваются флекс-элементы в контейнере?
72. Какие свойства и как они выравнивают флекс-элементы?
73. Что делают свойства flex-grow, flex-shrink и flex-basis?
74. Как работает та или иная единица измерения?
75. Какие варианты задания цветов существуют и чем они отличаются?
76. Что такое флекс-контейнер?
77. Что такое флекс-элемент?
78. Какие свойства применяются на флекс-контейнере?
79. Какие свойства применяются на флекс-элементе?
80. Какие оси есть у флексов и какие свойства меняют их направления?
81. Как выстраиваются флекс-элементы в контейнере?
82. Какие свойства и как они выравнивают флекс-элементы?
83. Что делают свойства flex-grow, flex-shrink и flex-basis?

84. Как работает та или иная единица измерения?
85. Какие варианты задания цветов существуют и чем они отличаются?
86. Что такое флекс-контейнер?
87. Что такое флекс-элемент?
88. Какие свойства применяются на флекс-контейнере?
89. Какие свойства применяются на флекс-элементе?
90. Какие оси есть у флексов и какие свойства меняют их направления?
91. Как выстраиваются флекс-элементы в контейнере?
92. Какие свойства и как они выравнивают флекс-элементы?
93. Что делают свойства flex-grow, flex-shrink и flex-basis?
94. Как работает та или иная единица измерения?
95. Какие варианты задания цветов существуют и чем они отличаются?
96. Что такое флекс-контейнер?
97. Что такое флекс-элемент?
98. Какие свойства применяются на флекс-контейнере?
99. Какие свойства применяются на флекс-элементе?
100. Какие оси есть у флексов и какие свойства меняют их направления?
101. Как выстраиваются флекс-элементы в контейнере?
102. Какие свойства и как они выравнивают флекс-элементы?
103. Что делают свойства flex-grow, flex-shrink и flex-basis?
104. Как работает та или иная единица измерения?
105. Какие варианты задания цветов существуют и чем они отличаются?
106. Что такое флекс-контейнер?
107. Что такое флекс-элемент?
108. Какие свойства применяются на флекс-контейнере?
109. Какие свойства применяются на флекс-элементе?
110. Какие оси есть у флексов и какие свойства меняют их направления?
111. Как выстраиваются флекс-элементы в контейнере?
112. Какие свойства и как они выравнивают флекс-элементы?
113. Что делают свойства flex-grow, flex-shrink и flex-basis?
114. Что значит “JS это слабо типизированный язык”?
115. Что значит “JS язык с динамической типизацией”?
116. Какие типы значений есть в JS?
117. Чем отличаются строки "", ” и “?
118. Что такое операнд и оператор?
119. Какие операторы есть в JS?
120. Как проверяются условия в JS?
121. Какие есть циклы в JS?
122. Как создаются функции?
123. Как хранятся данные в массиве?
124. Какие методы массивов вы знаете?
125. Как обратиться к элементу массива?
126. Как можно перебирать элементы массива?
127. Что такое объект?
128. Как получить значение из свойства объекта?
129. Для чего используется объект Math?

130. Приведите примеры свойств и методов в Math.
131. Что такое DOM?
132. Какие есть способы получения элементов из DOM?
133. Что такое событие?
134. Как добавить обработчик события?
135. Примеры свойств в объекте event.
136. Примеры методов у элементов DOM.
137. Назовите фазы жизненного цикла события.
138. На какой фазе изначально срабатывают обработчики событий?
139. Всплывают ли события blur и focus?
140. Для чего нужен preventDefault()?
141. Какие атрибуты полей форм можно использовать для добавления ограничений?
142. Почему требуется проверять данные на стороне клиента?
143. Что позволяет задать метод setCustomValidity()?
144. В какой момент показывается сообщение заданное в setCustomValidity()?
145. Что такое метод объекта?

Пример задания рубежного контроля

Главные требования

1. Работа выполнена самостоятельно.
2. Работа создана во время проведения занятия.
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.
5. Работа размещена на хостинге (сервер Мосполитеха).

Полезные ресурсы

Цвета:

- Готовые цветовые палитры: https://colorhunt.co/, https://coolors.co/
- Подобрать самостоятельно https://color.adobe.com/create/color-wheel

Шрифты: https://fonts.google.com/ (Продвинутый способ: https://google-webfonts-helper.herokuapp.com/fonts)

Примеры внешнего вида

- https://dessign.net/web-developer-portfolio-examples/

Задание

Документ index.html содержит:

- Шапка страницы:
    - Элемент ведущий на главную страницу (абсолютный путь до вашего домена): инициалы шрифтом из группы Display или произвольный логотип.
    - Ссылка на ваш профиль на github.
    - Навигационное меню (nav) с якорными ссылками на разделы страницы.
- Основная часть:
    - Крупный блок на весь первый экран (называется Hero Image или Jumbotron). Ваше имя, одно предложение о вас, кнопка с призывом связаться с вами, на фоне крупное изображение (можно взять тут https://unsplash.com/).

        - Список технологий и языков программирования с логотипами: что уже изучаете, что планируете в ближайшее время. Укажите опыт работы с технологией/языком программирования (полгода, год и т.д.). Списки стоит разместить в нескольких колонках.
        - Примеры ваших работ (скриншот, краткое описание, ссылка на сервер если есть) в виде карточек или карточки выполненных лабораторных работ по дисциплине “Основы веб-технологий”, номер, процент выполнения. В начале разделе есть ссылка со стрелкой “смотреть еще”.
        - Отзывы на ваши работы: рейтинг звездочками или другими изображениями, дата, отзыв, имя автора отзыва.
        - Контакты: почта, телефон, telegram и т.д. (можно использовать **ненастоящие** данные). Визуально выделите предпочтительный вариант для связи.
- Подвал:
    - ФИО, группа, дата написания РК.
    - Форма с возможностью оставить имя и номер телефона для обратной связи и чекбокс для срочной связи.

Критерии оценки (за невыполнение критерия баллы снимаются полностью или частично)

Общие

- Выполнены главные требования.
- Содержание страницы соответствует заданию.
- Страница выглядит цельно.

Код

- В коде соблюдаются отступы.
- В CSS не используются вложенность селекторов более двух.

HTML

- Правильная структура документа.
- Правильно использованы заголовки разных уровней.
- Использованы семантические теги html5: header, main, footer, nav, section; по необходимости article.
- HTML валиден, нет нарушений вложенности тегов.
- Выбраны правильные форматы изображений.
- Задан правильный язык страницы.
- Ссылки с телефонами, email сделаны корректно.
- Нет лишних оберток div.
- Если нет ссылки в шапке на ваш домен, то работа считается не загруженной на сервер.

CSS

- Корректно подключен и использован шрифт с Google Fonts.
- Использовано не менее двух разных значений для font-weight.
- Использованы пользовательские свойства (custom properties в :root) для цветовой палитры.
- Цвета из палитры применены к тексту, фону, границам.
- Header и footer выделены фоновым цветом.
- Контент страницы имеет ограничение максимальной ширины (1200-1400px) и выровнен по середине (одинаковые поля слева и права).

- Содержимое элементов не переполняет их.
- Стили выдерживаю увеличение или уменьшение количества контента.
- Элементы не наезжают на элементы, следующие за ними.
- Текст и изображения не слипаются друг с другом. Грамотно использованы внутренние и внешние отступы.
- Применено скругление границ.
- Использован **box-sizing: border-box;**.
- Карточки располагаются в ряд, имеют отступы друг от друга, использован flexbox.
- Изменяется оформление элемента при наведении на него курсора.
- К карточкам с проектами применена анимация, которая срабатывает при наведении.
- Использованы переходы.
- Изменены стандартные стили для элементов формы и чекбокса.
- Изменение состояния курсора при наведении на кнопки (cursor: pointer;).
- Классы названы логично, а не являются набором букв и цифр.

GIT и хостинг

1. Клонировать репозиторий, который был создан Github Classroom (**git clone ...**).
2. Установить необходимые пакеты для работы над проектом (**npm ci**).
3. При разработке периодически сохранять результаты в репозиторий git (команды **git add**, **git commit -m "комментарий к версии"**).
4. Перед завершением работы проверьте валидность вашего кода с помощью валидаторов для html и css.
5. Отформатируйте все документы html, css.
6. После проверки на валидаторе не забудьте отправить ваши изменения на github (**git push**).
7. Для загрузки на хостинг соберите проект (**npm run build**).
8. Создайте домен и загрузите в нужную папку с помощью FileZilla файлы из папки build.

7.3.2 Промежуточная аттестация

Оценочные средства для промежуточной аттестации не требуется, так как оценка за промежуточную аттестацию выставляется по балльно-рейтинговой системе, описанной в пункте 7.2.