

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ



УТВЕРЖДАЮ
Декан ФИиВТ

УТВЕРЖДАЮ /А.А. Кречетов/
(Ф.И.О. декана (директора института))

14.02.2024 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Б.1.2.1 Frontend-программирование

(код и наименование дисциплины по учебному плану)

Направление подготовки
(специальность)

09.03.04 Программная инженерия

Квалификация выпускника

Бакалавр

(бакалавр/магистр/специалист)

Направленность

Разработка программных систем

Курс

2

Семестр

3

Распределение учебного времени

Трудоемкость по учебному плану	108 / 3	часов/зачетных единиц
Лекции	36	часов
Лабораторные работы	36	часов
Практические занятия	-	часов
Иная контактная работа	-	часов
Всего контактной работы (без учета экз.)	72	часов
Контактная работа по экзамену	-	часов
Курсовой проект (работа)	-	семестр
Самостоятельная работа обучающихся (без учета экз.)	36	часов
Самостоятельная работа по подготовке к экзамену	-	часов
Экзамен	-	семестр
Зачет	-	семестр
БРК, ДЗ	3	семестр

(год)

Программа составлена в соответствии с требованиями ФГОС ВО направления подготовки (специальности) 09.03.04 Программная инженерия

Программу составили:

заведующий кафедрой с ученой степенью кандидата наук	ИиСП	СОГЛАСОВАНО	А.В. Бородин
(должность)	(кафедра)		(И.О. Фамилия)

РАССМОТРЕНА и ОДОБРЕНА на заседании кафедры, за которой закреплена дисциплина
Кафедра информатики и системного программирования

05.02.2024	протокол №	7
(дата)		

Заведующий кафедрой	СОГЛАСОВАНО	А.В. Бородин
		(И.О. Фамилия)

Рабочая программа СОГЛАСОВАНА с факультетом (институтом), выпускающей(ими)
кафедрой(ами).

СООТВЕТСТВУЕТ действующей ОП.

Заведующий кафедрой	СОГЛАСОВАНО	А.В. Бородин
		(И.О. Фамилия)

Председатель методической комиссии факультета (института), в который входит
выпускающая кафедра

	СОГЛАСОВАНО	А.А. Кречетов
		(И.О. Фамилия)

Эксперт(ы): Егошин Алексей Борисович, ген. директор ООО "Цитрус"

Рабочая программа проверена и зарегистрирована в УМЦ 12.03.2024 г.

Специалист учебно-методического центра СОГЛАСОВАНО /Т.А. Смирнова/

Раздел 1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью освоения дисциплины является достижение планируемых результатов обучения, соответствующих установленным в ОПОП индикаторам достижения компетенций:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Результаты обучения
1. ПК-2 Владение навыками использования различных технологий разработки программного обеспечения	ПК-2.1 Знает современные технологии разработки ПО (структурное, объектно-ориентированное)	знания: Знает современные технологии разработки ПО (структурное, объектно-ориентированное). умения: навыки:
	ПК-2.2 Умеет использовать современные технологии разработки ПО	знания: умения: Умеет использовать современные технологии разработки ПО. навыки:
	ПК-2.3 Имеет навыки использования современных технологий разработки ПО	знания: умения: навыки: Имеет навыки использования современных технологий разработки ПО.

Раздел 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина относится к части, формируемой участниками образовательных отношений ОПОП.

Дисциплина является обязательной

Изучаемая дисциплина является основой для продолжения формирования указанных компетенций в следующих дисциплинах: Технология разработки программного обеспечения (ПК-2), Проектирование игр (ПК-2), Системы виртуальной и дополненной реальности (ПК-2); практиках: Производственная практика. Проектно-технологическая практика (ПК-2); государственной итоговой аттестации в форме: Подготовка к процедуре защиты и защита выпускной квалификационной работы (ПК-2)

Раздел 3. ОПИСАНИЕ ОБРАЗОВАТЕЛЬНЫХ ТЕХНОЛОГИЙ

Для формирования заявленных компетенций используются методологические технологии, реализующие деятельностный, личностно-ориентированный, практико-ориентированный подходы.

Основными стратегическими технологиями являются: лекционные занятия, практические и лабораторные занятия, процедуры самообучения

На достижение конкретных целей обучения направлены применяемые тактические технологии: задания, информационные, классическая лекция, проблемная лекция

Раздел 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

3 семестр

Виды и темы занятий	Количество часов	Формируемые компетенции
Введение. Ознакомление с материалом.	34	ПК-2
Лекция. Лекция №1. Процесс разработки и инструменты.	4	
Лекция. Лекция №2. Веб-страницы (HTML).	4	
Лекция. Лекция №3. Веб-страницы (HTML+CSS).	4	
Лабораторная работа. Лабораторная работа №1. Верстка с использованием HTML.	6	
Лабораторная работа. Лабораторная работа №2. Верстка с использованием HTML и CSS.	6	
Задания для самостоятельной работы, в том числе выполнение Проработка лекций. Подготовка к блиц-опросам. Подготовка к выполнению лабораторных работ. Подготовка к защите лабораторных работ.	10	
Внешний интерфейс	34	ПК-2
Лекция. Лекция №4. Разработка внешнего интерфейса.	4	
Лекция. Лекция №5. Блочная верстка. Позиционирование элементов. Адаптивная и отзывчивая верстка. Как сделать страницу отзывчивой. Таблицы и формы.	4	
Лекция. Лекция №6. CSS-препроцессоры. SASS.	4	
Лабораторная работа. Лабораторная работа №3. Верстка по эскизам и прототипам.	6	
Лабораторная работа. Лабораторная работа №4. Использование препроцессинга.	6	
Задания для самостоятельной работы, в том числе выполнение Проработка лекций. Подготовка к блиц-опросам. Подготовка к выполнению лабораторных работ. Подготовка к защите лабораторных работ.	10	
Языки программирования	40	ПК-2
Лекция. Лекция 7. Процесс программирования. JavaScript. Основы JS. JavaScript. Операторы. JavaScript. Циклы. JavaScript. Функции. JavaScript. Массивы. JavaScript. Объекты. JavaScript. DOM. Объектная модель документа. JavaScript. Взаимодействие со страницей. JavaScript. События. jQuery.	4	
Лекция. Лекция 8. Frontend-фреймворки. Zurb Foundation. Адаптивная сетка. Zurb Foundation. Элементы страницы. Zurb Foundation. Плагины.	4	
Лекция. Лекция 9. Оптимизация страницы. Инструменты проверки. Оптимизация страницы. Уменьшаем объем данных. Оптимизация страницы. Уменьшаем число обращений к серверу. Оптимизация страницы. Отложенная загрузка.	4	
Лабораторная работа. Лабораторная работа №5. Верстка динамических страниц со сценариями по индивидуальному заданию.	12	

Задания для самостоятельной работы, в том числе выполнение Проработка лекций. Подготовка к блиц-опросам. Подготовка к выполнению лабораторных работ. Подготовка к защите лабораторных работ. Подготовка к БРК.	16	
Иная контактная работа:	0	

Раздел 5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Изучение дисциплины рекомендуется начать с ознакомления с рабочей программой, ее структурой и содержанием разделов. Учебный материал структурирован, изучение дисциплины осуществляется в тематической последовательности.

Занятия лекционного типа дают систематизированные знания по дисциплине, концентрируют внимание на наиболее сложных и важных вопросах. Во время лекционных занятий рекомендуется вести конспектирование учебного материала; обращать внимание на формулировки и категории, раскрывающие суть проблемы, явления или процесса; зафиксировать выводы и практические рекомендации.

Подготовка к **занятиям семинарского типа** включает ознакомление с планом лабораторного занятия; работу с конспектом лекций, выполнение домашнего задания, работу с учебной и учебно-методической литературой, научными изданиями и электронными образовательными ресурсами, рекомендованными рабочей программой дисциплины.

Содержание **самостоятельной работы** определяется рабочей программой дисциплины, оценочными и методическими материалами, заданиями и указаниями преподавателя. Самостоятельная работа может осуществляться в аудиторной и внеаудиторной формах. Эффективным средством осуществления самостоятельной работы является электронная информационно-образовательная среда университета, которая обеспечивает доступ к образовательной программе, рабочей программе дисциплины, к электронным библиотечным системам, профессиональным базам данных и информационным справочным системам.

Изучение дисциплины включает выполнение лабораторных работ. Периодичность проведения, формы текущего контроля успеваемости, система оценивания хода освоения дисциплин представлены в рабочей программе.

Формой промежуточной аттестации по дисциплине является балльно-рейтинговый контроль.

Раздел 6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Учебно-методическое обеспечение

№№ п/п	Список используемой литературы	Количество экземпляров печатных изданий, имеющихся в библиотеке, или электронный адрес издания (ресурса) в сети Интернет
УЧЕБНЫЕ, УЧЕБНО-МЕТОДИЧЕСКИЕ И НАУЧНЫЕ ИЗДАНИЯ		
1.	Заяц, А. М. Проектирование и разработка WEB-приложений. Введение в frontend и backend разработку на JavaScript и node.js [Электронный ресурс] / Заяц А. М., Васильев Н. П. 3-е изд., стер. Санкт-Петербург: Лань, 2021. - 120 с. ISBN 978-5-8114-7042-6.	https://e.lanbook.com/book/154380
2.	Сычев, А. В. Теория и практика разработки современных клиентских веб-приложений [Электронный ресурс] / Сычев А. В. 2-е изд. Москва: ИНТУИТ, 2016. - 483 с.	https://e.lanbook.com/book/100528
3.	Савельев, А. О. HTML5. Основы клиентской разработки [Электронный ресурс] / Савельев А. О., Алексеев А. А. 2-е изд. Москва: ИНТУИТ, 2016. - 271 с.	https://e.lanbook.com/book/100595
4.	Введение в HTML5 [Электронный ресурс]. 2-е изд. Москва: ИНТУИТ, 2016. - 133 с.	https://e.lanbook.com/book/100680
5.	Введение в стандарты Web [Электронный ресурс]. 2-е изд. Москва: ИНТУИТ, 2016. - 800 с.	https://e.lanbook.com/book/100712
6.	Сычев, А. В. Web-технологии [Электронный ресурс] / Сычев А. В. 2-е изд. Москва: ИНТУИТ, 2016. - 408 с.	https://e.lanbook.com/book/100725
7.	Васильев, Н. П. Введение в гибридные технологии разработки мобильных приложений [Электронный ресурс] / Васильев Н. П., Заяц А. М. 3-е изд., стер. Санкт-Петербург: Лань, 2022. - 160 с. ISBN 978-5-507-44502-8.	https://e.lanbook.com/book/230387
ЭЛЕКТРОННЫЕ ОБРАЗОВАТЕЛЬНЫЕ РЕСУРСЫ		
1.	Научная электронная библиотека eLIBRARY.RU	http://elibrary.ru
2.	Научная электронная библиотека «Киберленинка»	http://cyberleninka.ru

6.2. Материально-техническая база и программное обеспечение

№№ п/п	Аудитории для проведения учебных занятий, самостоятельной работы и проведения государственной итоговой аттестации	Перечень основного оборудования	Программное обеспечение
1.	427 (III)	Мобильный телефон Samsung Galaxy A7 (2), Мобильный телефон Samsung Galaxy S9+ (2), Ноутбук Apple MacBook Pro13 with Retina display and Touch Bar Mid2017 (1), Планшет Apple iPad 2018 (1), Проектор мультимедийный Hitachi CP- RX94 (1), Смартфон APPLE iPhone 8 Plus 64 Gb, MQ8L2RU/A, серый (1), Смартфон APPLE iPhone	Microsoft Windows Enterprise, Справочная правовая система "Консультант Плюс", Microsoft Office Standard, Агент Dr.Web, Комплект ГАРАНТ-Мастер, Microsoft Access, Microsoft Visio Professional, Microsoft

		X 64 Gb, MQAD2RU/A, серебристый (1), Шлем виртуальной реальности HTC Vive (2), Комплект учебной мебели (1)	Project Professional, Microsoft Visual Studio Enterprise, Комплект ПО для решения основных пользовательских задач
2.	521 (I)	Комплект учебной мебели (1)	Microsoft Windows Enterprise, Справочная правовая система "Консультант Плюс", Microsoft Office Standard, Агент Dr.Web, Комплект ГАРАНТ-Мастер, Microsoft Access, Microsoft Visio Professional, Microsoft Project Professional, Microsoft Visual Studio Enterprise, Комплект ПО для решения основных пользовательских задач
3.	522 (I)	Анализатор спектра NS-30A (1), Антенна M102 в компл. с кабелем ВЧ TNCm-SMAm (1), Блок питания лаборат. НУ 3003 D-3 (1), Внешний HDD WD 2TB 3.0 , 3.5"USB (1), Внешний накопитель 1 Seagate Original USB 3.0 4 Tb (1), Внешний накопитель флешка USB TRANSCEND Jetflash 780 64 Gb (1), Гигабитный управляемый коммутатор на 16 портов (1), Измеритель CN -801 HP (1), Кондиционер AEG ACS-09HR (1), Многофункциональный измерительный прибор (1), Монитор 20 "Beng FP 202W (2), Монитор LCD Samsung 17" SM 713N (1), МФУ Canon i-SENSYS MF 4018 (1), МФУ 1 Лазерный Canon i-Sensys MF226 (1), Набор ВЧ переходников (1), Ноутбук Dell Latitude E6520 Intel Core I5 Processor 2520M 15,6" (2), Ноутбук TOSHIBA Satellite L655-1H2-RU (1), Паяльная станция AOYUE 968 (1), Переключатель ZX80-DR230 (1), Персональный компьютер 3 Atlant A2X4/4G(3)/512Mb/монитор Pyama 2209/3Y (1), ПК RAMEC GALE LCD LG 23"/Intel i5 4590/MSI B85M-E45/2x4DDR3/GT740	Microsoft Windows Enterprise, Справочная правовая система "Консультант Плюс", Microsoft Office Standard, Агент Dr.Web, Комплект ГАРАНТ-Мастер, Microsoft Access, Microsoft Visio Professional, Microsoft Project Professional, Microsoft Visual Studio Enterprise, Комплект ПО для решения основных пользовательских задач

	2Gb/500Gb/клав,мышь (28), Преобразователь SP-200-24-AC-DC в кожухе 199x99x50мм (1), Приемопередающая программно-конфигурируемая радиоплатформа G32 (1), Принтер Canon LBP 2900 лазерный с кабелем (1), Проектор мультимедийный Hitachi CP-EX250 (1), Проектор мультимедийный Hitachi CP-EX251N (1), Сист. блок Pen D 945 3.4 DDR 2 1024*2/FDD 3.5/250 Gb/DVD-RW/кл+мышь+коврик (1), Системный блок CPU Intel Core i7-6700/ASRod Z-170/32 Gb/GTX 1070/200 Gb/Wi-Fi +клав, мышь (1), Станок сверлильный 350 Вт (1), Универсальная приёмопередающая платформа для проектирования СВЧ-систем компл.mgx92 (1), Усилитель LZY-22 (1), Усилитель ZHL-3A-S (1), Комплект учебной мебели (1)	
--	---	--

Раздел 7. ФОРМЫ КОНТРОЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ/ ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Критерии оценивания индикаторов достижения компетенций направлены на:

- усвоение теоретического материала (объем знаний, глубина усвоения), предусмотренного рабочей программой;
- умение излагать материал (четкость, грамотность изложения материала, точность и полнота воспроизведения учебного материала);
- умение применять теоретические знания при решении практических заданий.

Шкала оценивания представлена ниже.

Уровень сформированности элементов компетенции	Критерии оценивания	Шкала оценивания
Пороговый уровень	Обучающийся имеет знания основного материала, проявляет умение логично его излагать, но может допускать неточности в изложении материала, недостаточно правильные формулировки, испытывает затруднения в выполнении практических заданий.	удовлетворительно
Продвинутый уровень	Обучающийся твердо знает программный материал, излагает его грамотно и по существу, не допускает существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения	хорошо
Высокий уровень	Обучающийся глубоко и прочно усвоил программный материал, грамотно и логически стройно его излагает, дает исчерпывающие ответы на поставленные вопросы. В	отлично

	<p>ответе тесно увязывается теория с практикой, при этом обучающийся не затрудняется с ответом при видоизменении задания, свободно справляется с задачами, вопросами и другими видами применения знаний, показывает знакомство с монографической литературой, периодическими изданиями, правильно обосновывает принятые решения, свободно владеет разносторонними навыками, приемами выполнения практических работ</p>	
--	--	--

7.1. Текущий контроль успеваемости

Текущий контроль успеваемости обеспечивает оценивание хода освоения дисциплины (модуля) и производится с применением технологии рейтингового контроля в соответствии с технологической картой дисциплины. Порядок составления технологической карты и алгоритм проведения процедуры оценивания видов деятельности обучающихся, направленных на освоение знаний, умений, навыков и/или опыта деятельности, по накопительной системе в баллах устанавливается положением о системе РИТМ в ФГБОУ ВО «ПГТУ»

7.2. Промежуточная аттестация обучающихся

Промежуточная аттестация обучающихся направлена на оценивание результатов обучения по дисциплине (модулю) и проводится с использованием фондов оценочных средств.

Примеры типовых контрольных заданий из базы фонда оценочных средств по образовательной программе.

1. Какой тег должен открывать HTML-программу, а какой закрывать её?

Ответ. Тег `<HTML>` должен открывать программу, а тег `</HTML>` — закрывать её.

2. Из каких двух частей состоит HTML-программа? Запишите общий вид HTML-программы.

Ответ. Между двумя основными тегами `<HTML>` и `</HTML>` располагается *головная часть* программы и ее *тело*:

```
<HTML> ..голова ..тело </HTML>
```

3. Какими тегами задаётся элемент, содержащий головную часть программы? Запишите общий вид головной части HTML-программы.

Ответ. Головная часть HTML-программы записывается между открывающим тегом `<HEAD>` и закрывающим `</HEAD>`:

```
<HEAD> ..содержимое головной части </HEAD>
```

4. Какими тегами задаётся элемент, содержащий тело программы? Запишите

общий вид тела HTML-программы.

Ответ. Тело HTML-программы записывается между открывающим тегом `<BODY>` и закрывающим `</BODY>`:

```
<BODY> ..содержимое тела программы</BODY>
```

5. Какие указания записываются в головной части программы?

Ответ. В элементе `<HEAD>...</HEAD>` описываются общие правила интерпретации HTML-кода в целом, и содержится вспомогательная информация о документе и авторе документа (теги `<META>`). Кроме того, головная часть содержит элемент `TITLE` — в нём задаётся название окна показа документа.

6. Для чего нужно указывать кодировку, в которой написана гипертекстовая страница?

Ответ. Указание кодировки браузер использует для выбора кодовой таблицы, задающей соответствие между кодами символов и их образами на экране.

7. Что произойдёт, когда браузер будет интерпретировать HTML-код с неверно указанной кодировкой?

Ответ. Текст на экране будет в нечитаемом виде.

8. Как указать кодировку, в которой написана гипертекстовая страница?

Ответ. Нужно в головную часть (`<HEAD>...</HEAD>`) поместить тег `META` с указанием кодировки в виде:

```
<META http-equiv="Content-Type" content="text/html; charset=windows-1251">
```

В примере задана кодировка windows-1251.

9. В каких случаях нельзя указывать кодировку гипертекстовой страницы?

Ответ. Если сервер, на котором расположен сайт, поддерживают автоматическую перекодировку страниц, то указание кодировки может привести к путанице: сервер сообщит браузеру, что перешлёт ему страницу в той кодировке, которая указана в теге `META`, а сам перекодирует страницу в другую. Пользователь увидит на экране “кракозяки”.

Разработчик должен поинтересоваться у провайдера сервера, на котором он собирается держать свой сайт, будет ли сервер перекодировать страницы перед

передачей браузеру клиента, нужно ли на страницах указывать кодировку.

10. Как в HTML-коде задаётся название окна, в котором будет отображаться документ?

Ответ. Название окна задаётся в головной части при помощи элемента **TITLE**:

```
<TITLE>Название окна</TITLE>
```

11. Почему название окна должно совпадать с заголовком страницы?

Ответ. Это удобно по трём причинам:

1. Когда окно с сайтом свёрнуто в кнопку на *Панели Задач*, пользователь видит на этой кнопке заголовок текущей страницы.
2. Название окна заносится в набор избранных сайтов (меню *Избранное*), если пользователь решит запомнить ссылку на текущую страницу.
3. Название окна — дополнительная подсказка пользователю о его текущем положении на сайте.

12. Для чего используются теги **METAS** атрибутами **description** и **keywords**?

Ответ. Эти элементы задают описание документа и ключевые слова для поисковых систем, что способствует продвижению (“раскрутке”) сайта в Интернете.

13. Какую роль играют заголовки документа?

Ответ. Заголовки — важный структурный элемент разметки документа. Они разделяют информацию на отдельные логические части и поэтому существенно улучшают её восприятие.

14. Каким элементом задаются заголовки?

Ответ. В HTML-документе поддерживаются заголовки шести уровней при помощи конструкции:

```
<Hn>текст заголовка</Hn>
```

В качестве **n** следует использовать одну из цифр 1–6, обозначающую уровень заголовка.

15. Сколько уровней заголовков поддерживает HTML?

Ответ. Шесть уровней: **h1** (первый уровень), **h2**, **h3**, **h4**, **h5**, **h6** (последний уровень).

16. Как выделяются заголовки в окне документа?

Ответ. При отображении HTML-документа браузер выделяет заголовки размером шрифта и жирностью.

17. Почему не следует использовать элементы `h1` для выделения обычного текста?

Ответ. Элементы `h1` предназначены для структурной разметки. Применять их для обычных выделений текста неразумно по двум причинам.

Во-первых, нарушение логики разметки будет мешать самому разработчику.

Во-вторых, существуют специальные программы, которые автоматически составляют оглавление гипертекстового документа, используя элементы `h1`. Если заголовочные теги обрамляют обычные тексты или не отражают иерархическую структуру документа, оглавление будет построено неверно.

18. Каким элементом задаётся горизонтальная линия?

Ответ. Горизонтальная линия задаётся тегом `<hr>`, и этот тег не имеет парного закрывающего.

19. Зачем текст разделяется на абзацы?

Ответ. Абзацы помогают структурировать текст, разделяя его на небольшие логические порции.

20. Назовите два приёма, позволяющих зрительно отделить абзацы друг от друга в тексте.

Ответ. Абзацы отделяются друг от друга “красными” или пустыми строками. Красная строка — это первая строка абзаца с текстом, сдвинутым вправо.

21. Какой приём использует браузер для отделения абзацев друг от друга?

Ответ. По умолчанию браузер отделяет абзацы друг от друга пустыми строками.

22. Как на HTML задаётся абзац?

Ответ. Абзац задаётся тегами `<p>` и `</p>`, между которыми помещается текст.

23. Опишите правила построения браузером абзаца на экране.

Ответ. Браузер строит абзац следующим образом:

1. Абзац выводится с новой строки и перед ним формируется отступ (пустая строка).
2. Абзац выравнивается по левому краю.
3. Между словами помещается ровно по одному пробелу независимо

от того, сколько пробелов проставлено в HTML-коде.

4. Перенос текста на новую строку выполняется, если очередное слово не помещается в экранной строке, а не тогда, когда в HTML-коде начинается новая строка.
5. Завершая абзац, браузер выполняет переход на новую строку и формирует после абзаца отступ (пустую строку).

24. Опишите поведение объекта, соответствующего блочному элементу.

Ответ. Блочные элементы выводятся на экран прямоугольниками, последовательно сверху вниз. Ширина прямоугольника совпадает с шириной родительского элемента, а высота определяется по содержимому.

25. Опишите поведение объекта, соответствующего строчному элементу.

Ответ. Строчные элементы содержатся внутри блочных и занимают место в строке, переходя на следующие строки, если в текущей не хватило места.

К строчным относятся элементы, позволяющие выделять текст (например, курсивом или жирностью), а также обычный неразмеченный текст — он автоматически собирается браузером в безымянные строчные элементы.

26. Приведите примеры блочных элементов.

Ответ. Самый главный блочный элемент — **HTML**. Он содержит блочный элемент **BODY**, в котором располагаются все другие элементы страницы.

Примеры блочных

элементов: **ADDRESS, BLOCKQUOTE, BODY, DD, DIV, DL, DT, Hn, HR, HTML, LI, OL, P, PRE, TABLE, T**
BODY, TD, TFOOT, TH, THEAD, TR, UL.

27. Приведите примеры строчных элементов.

Ответ. Примеры строчных

элементов: **A, ABBR, ACRONYM, B, BIG, BR, CITE, CODE, DFN, EM, FONT, I, IMG, KBD, Q, SAMP, SMAL**
L, SPAN, STRONG, SUB, SUP, TT, U, VAR.

28. Может ли блочный элемент содержать блочные элементы? А строчные? А смесь из строчных и блочных?

Ответ. Блочный элемент может содержать блочные элементы, строчные, а также смесь из блочных и строчных.

29. Может ли строчный элемент содержать блочные элементы? А строчные? А смесь из строчных и блочных?

Ответ. Строчный элемент может содержать только строчные элементы.

30. Может ли элемент `<div>` содержать блочные элементы?

Ответ. Нет. Это исключение из общего правила.

31. Что происходит с элементами на экране, когда пользователь меняет ширину окна документа?

Ответ. При изменении ширины окна меняется ширина `<html>` и вложенного в него `<body>` таким образом, чтобы занять по ширине всё окно.

Соответственно, меняется ширина всех вложенных в `<body>` элементов таким образом, чтобы они занимали всю ширину `<body>`. И так происходит на каждом иерархическом уровне при вложении элементов друг в друга.

Строчные элементы при этом также стараются занять всю ширину своих блочных предков.

32. Что делает браузер, когда ширина окна уменьшается настолько, что в окне не может целиком уместиться даже одно слово?

Ответ. В этом случае браузер вынужден включать горизонтальную прокрутку, чтобы дать пользователю возможность просмотреть целиком содержимое страницы.

33. Как записывать код лесенкой и для чего это нужно?

Ответ. Код, записанный лесенкой, гораздо легче читать и понимать. Структурная лесенка должна повторять иерархическую структуру блочных элементов: каждый потомок смещается вправо по отношению к телу своего родителя.

34. Почему рекомендуется записывать код строчками не длиннее 80 символов?

Ответ. Если записывать строчки более длинными, то для их просмотра придётся использовать горизонтальную прокрутку окна текстового редактора, а это не удобно.

35. В каком регистре рекомендуется записывать имена тегов, названия атрибутов и их значения? Чем обосновывается эта рекомендация?

Ответ. Имена тегов рекомендуется записывать прописными буквами для выделения в коде тегового каркаса. Названия атрибутов и их значения рекомендуется записывать строчными буквами, чтобы не переборщить с выделениями.

Вопросы к БРК

36. Что такое front-end и чем он отличается от back-end? Какая часть сайта относится к фронтенду?
37. Как происходит процесс разработки страницы? Какие современные инструменты фронтенд-разработки существуют?
38. Из чего состоит веб-страница? Как разместить на странице материал и как оформить внешний вид страницы?
39. Какие основные требования предъявляются к веб-странице? Что такое валидация и зачем она нужна?
40. Чем отличается блочная и табличная верстка?
41. Чем отличается фиксированный макет от резинового? Рассказать про оба вида.
42. Чем отличается адаптивный макет от отзывчивого? Как сделать макет адаптивным/отзывчивым? Рассказать про оба вида.
43. Что такое CSS-препроцессоры? Какие преимущества они дают? Рассказать на примере SASS.
44. Что такое JavaScript? Как начать использовать его на веб-странице? Рассказать о структуре кода, переменных, основных операторах.
45. JavaScript. Функции, массивы, объекты.
46. DOM. Взаимодействие с деревом документа.
47. jQuery. Что это такое и какие преимущества дает?
48. Что такое фронтенд-фреймворк? Какие преимущества он дает? Рассказать на примере Bootstrap/Foundation.
49. Что такое адаптивная сетка и как ей пользоваться?
50. Какие вы знаете способы оптимизации клиентской части веб-страницы? Какие есть инструменты для проверки оптимизации?