1. **Представление о макете веб-страницы.**

Когда ваш браузер визуализирует Web-страницу, он строит коллекцию объектов Document Object Model (DOM), которая представляет подборку элементов HTML, из которых состоит страница.

Объектная модель документа(Document Object Model, DOM) — это программный интерфейс, не зависящий от платформы и языка, позволяющий программам и скриптам получить доступ к содержимому HTML, XHTML и XML-документов, а также изменять содержимое, структуру и оформление таких документов. Модель DOM не накладывает ограничений на структуру документа. Любой документ известной структуры с помощью DOM может быть представлен в виде дерева узлов, каждый узел которого представляет собой элемент, атрибут, текстовый, графический или любой другой объект. Узлы связаны между собой отношениями родительский-дочерний. Изначально различные браузеры имели собственные модели документов (DOM), не совместимые с остальными. Для того чтобы обеспечить взаимную и об-ратную совместимость, специалисты консорциума W3C классифицировали эту модель по уровням, для каждого из которых была создана своя спецификация. Все эти спецификации объединены в общую группу, носящую название W3C DOM.

1. **Рекомендации для составления веб-страницы.**

Создавать сайт по госту?

Присутствие header article aside footer?

1. **Инструменты для редактирования веб-страницы.**

**Brackets**(Windows, OS X, Linux; brackets.io)

**Atom**(Windows, OS X, Linux; atom.io)

**jEdit**(Windows, OS X, Linux; jedit.org).

**EditPlus**(Windows, editplus.com)

**Visual Studio Code** (OS X, skedit.com)

**Sublime Text**(OS X, Windows, Linux; sublimetext.com)

Dreamweaver(OS X и Windows, adobe.com/ru/products/dreamweaver.html)

1. **Основы HTML5: историческая справка, стандарты, структура html-документа, назначение**

**HTML (Hypertext Markup Language) —** это набор символов разметки или кода, размещенных в файле, предназначенном для отображения веб-браузером. Веб-браузер обрабатывает код в HTML- файле и отображает документ веб-страницы и связанные с ним файлы. Консорциум W3C устанавливает стандарты дляHTML

**HTML5 —** актуальная версия языка HTML, существует уже несколько лет. В версии HTML5 были добавлены новые элементы. Например, элемент headerможет включать в себя содержимое, которое обычно встречается в верхней части страницы, — логотип и общие для всего сайта навигационные ссылки. Новый элемент nav заключает в себе набор ссылок, использующихся для навигации по сайту, а элемент footer размещает в себе все, что обычно помещается в нижней части страницы,

Кроме того, в HTML5 добавлены новые элементы, позволяющие внедрять на страницу видео- и аудиоконтент, и новые элементы формы, добавляющие такие сложные компоненты, как ползунковые регуляторы, всплывающие панели выбора даты, а также встроенную браузерную поддержку проверки допустимости данных, введенных в форму (которая гарантирует правильное заполнение ваших форм посетителями).

1. **Кодировки (ASCII, Windows-1252, UTF-8, ISO-8859-1)**

Кодировка символов международное представление букв, цифр и символов в веб-странице или любом другом файле, который хранится на компьютере и может быть передан через Интернет.

**ASCII (American Standard Code for Information Interchange, американская стандартная кодировочная таблица для печатных символов и некоторых специальных кодов)—** одна из самых старых компьютерных кодировок, в которой каждому символу соответствует строго определенное число. Это 7-битная кодировка, содержащая 128 символов

**Windows-1251 —** набор символов и кодировка, являющаяся стандартной 8-битной кодировкой для всех русских версий Microsoft Windows. Данная кодировка пользуется довольно большой популярностью в восточно-европейских странах. Windows-1251 отличается от других 8-битных кириллических кодировок наличием практически всех символов, использующихся в традиционной русской типографике для обычного текста (отсутствует только знак ударения)

**UTF-8 –** кодировка символов юникод в двоичном виде. Область UTF-8 с кодами от U+0000 до U+0500 — это базовая область символов кирилицы. Если Вы хотите какой-либо из этих знаков отобразить в HTML-документе, Вы можете использовать шестнадцатеричное (Hex) значение ( &#x + код + ;) или десятичное (Dec) значение ( &# + код + ;) из таблицы кодов

**ISO-8859-1 —** кодировка по умолчанию в большинстве современных браузеров.

Первые 128 символов ISO-8859-1 закодированы так же, как и в ASCII (прописные и строчные латинские буквы, цифры от 1 до 9 и некоторые другие символы).

Коды от 160 до 255 содержат символы, используемые в западно-европейских странах и часто используемые спецсимволы.

1. **Служебные и структурные теги.**

**Служебные**

**Тег html**

Требуется в начале веб-страницы и с добавленным слешем в конце: </html>. Элемент html сообщает браузеру, что документ является программным кодом на языке HTML. Все содержимое страницы,

включая остальные элементы, находится между открывающим и закрывающим тегами элемента html

**Атрибуты тега html**

**Атрибут dir**

Определяет, в каком направлении браузер должен выводить текст в элементе, к котором применяется этот атрибут. По умолчанию выводит слева направо (значение ltr), для арабского и иврита применяется значение rtl

**Атрибут lang**

Определяет язык, который употребляется главным образом в документе. Использованный с другим тегом, атрибут lang назначает язык содержимого тега. Применяется, чтобы лучше отображать текст. (значения en, en-US, fr, de, it, ul (голландский), el, es, pt, ar, he, ru, zh, hi и т.д. согласно ISO-639)

**Тег head**

Служебный тег, который содержит в себе всю техническую информацию о веб-странице, а также задает ее заголовок. Эта информация не видна обычному пользователю, но помогает браузерам и поисковым системам работать с данными страницы

В нем могут содержаться такие элементы, как title, base, meta, link, style

**Атрибуты тега head**

**Атрибут profile**

Указывает адрес URL профиля метаданных. Обычно браузер распознаёт значение этого атрибута и выполняет некоторые соглашения, связанные с указанным профилем. Загрузки самого документа по указанному адресу в реальности не происходит, более того, его может вообще не быть

**Тег title**

Используется для определения заголовка документа и отображается в заголовке окна браузера

**Тег link**

Используется для подключения к документу стилей, скриптов, изображений или значков из внешних файлов

**Атрибуты тега link**

Атрибут **href**

от англ. "hypertext reference" ‒ «гипертекстовая ссылка»

Указывает местоположение внешнего ресурса или файла.

Атрибут **rel**

от англ. “relation" ‒ «отношение»

Определяет, чем для нашего документа является подключаемый файл

**Тег meta**

Содержит дополнительные данные о документе (метаданные). Эти данные используют браузеры для обработки страницы, а поисковые системы — для ее индексации.

**Атрибуты meta**

**Атрибут charset**

Определяет кодировку документа

**<meta charset="utf-8">**

**Атрибут http-equiv**

Управляет действиями браузеров и используются для формирования информации, выдаваемой обычными http-заголовками

**Значения атрибута http-equiv**

**Content-Type** тип кодировки документа

**expires** устанавливает дату и время, после которой информация в документе будет считаться устаревшей

**pragma** способ кэширования документа

**refresh** указывает время в секундах до перезагрузки страницы или время до перенаправления на другую страницу

**Значения атрибута http-equiv**

**<meta http-equiv="Content-Type" content="text/html; charset=utf-8">**

**<meta http-equiv="expires" content="Sun, 01 Jan 2013 07:01:00 GMT">**

**Обновление страницы через 10 секунд**

**<meta http-equiv="refresh" content="10">**

**Значения атрибута name**

**application-name** указывает название веб-приложения, используемого на странице  
author используется для указания имени автора веб-страницы

**description** информация, которую вы види-те на странице поисковика, когда ваша страница попадает в результаты запроса

**generator** указывает один из пакетов программного обеспечения, используемого для создания документа

**keywords** содержит список ключевых слов, разделенных запятыми, соответствующих содержимому страницы

**robots** показывает, должны ли поисковые системы включать данную страницу в результаты поиска

**viewport** позволяет разработчикам управлять размером исходной области просмотра на различных устройствах

**Тег base**

Тег <base> используется, если требуется выполнить локальное тестирование. Браузер ищет элемент <base>, определяет полный адрес документа и корректно загружает его.

Например, если адрес документа указан как <base href="http://www.megasite.ru/hzchd/">, то при добавлении рисунков достаточно использовать относительный адрес <img src="images/labuda.gif">. При

этом полный путь к изображению будет http://www.megasite.ru/hzchd/images/labuda.gif, что позволяет браузеру всегда находить графический файл, независимо от того, где находится текущая веб-страница.

**структурные теги**

***Main:*** применяется для создания основного блока страницы. Это контент, внутри которого могут быть такие теги, как: Header, Section, Article и т. д. Содержимое Main должно быть уникальным. Это достигается двумя путями: выносом из блока контента, который повторяется на других страницах, или добавление к тегу Main идентификатора aria-labelledby.

***Header****:*или шапка внутри документа(не путать с <head> ), используется для заглавия статей или разделов. Хедер может содержать, помимо заголовка, дату публикации контента или оглавление статьи.

***Nav****:* используется для включения меню в разные части сайта и содержит данные навигации. Тег nav является лишь оболочкой списков меню, поэтому между <nav> и </nav> должны вставляться <ul> и <ol>. Google может включать в расширенные сниппеты с быстрыми ссылками меню сайта, заданное через тег Nav.

***Section:*** служит для выделения разделов сайта. Типичный пример применения — лендинг или одностраничное портфолио, где элемент section вкупе с Id используется для перехода к разделам «Портфолио», «Контакты» и т. п. Этот элемент HTML5 создан специально для того, чтобы избежать чрезмерного употребления div class или div id со слишком глубокой вложенностью дочерних элементов. Все секции могут применяться как контейнеры для организации контента внутри них. Допустимо даже использование section внутри section, когда необходимо разделение на подразделы в рамках родительской секции или для сайта целиком, если речь об одностраничнике. Если разместить section внутри article, тег будет подчиняться семантике и контентным правилам статейных блоков. Бытует мнение, что подобное использование section может нарушить приоритеты структуры, но спецификация w3 содержит примеры таких вложений, поэтому конструкция считается валидной.

***Article:*** каждая статья сайта. Статьи в данном контексте — это блоки информации. Они могут быть новостями, лонгридами, инструкциями, постами блога, услугами и продуктами, но не обязаны ими являться. Суть идеи состоит в том, что в одном разделе присутствуют разные материалы article, сгруппированные с учетом некой логики внутри родительских тегов. Важный момент — тег может быть неуникальным в рамках сайта, содержать вложенные элементы того же или более высокого уровня — на новостных сайтах часто встречаются примеры article внутри article.

***Aside:*** боковая панель или сайдбар. На самом деле здесь можно разместить любой контент, косвенно связанный с основным содержанием страницы; категории, статьи, облако тегов, последние комментарии, связанные записи и т. д. Тег aside может быть дополнением материала страницы, либо содержать отдельную общую информацию в рамках сайта, например, боковое меню.

***Footer:*** представляет краткую информацию о разделе, авторах или сайте — авторские права, ссылки на служебные страницы и т. п. Футер должен располагаться внизу страницы, раздела, статьи и т. д.

**7. Семантические теги.**

**Семантические теги —** это теги, которые предназначены для того чтобы компьютерные программы (поисковые системы, сборщики информации, речевые браузеры и т.д.), понимали какой тип информации заложен в данных тегах.

**header** устанавливает верхний колонтитул («шапку») для страницы или раздела

**nav** предназначен для создания блока навигации веб-страницы или всего веб-сайта, при этом не обязательно должен находиться внутри

**footer** устанавливает нижний колонтитул («подвал») для страницы или раздела

**section** представляет собой общий раздел документа или приложения. Раздел в этом контексте представляет собой тематическую группу контента, обычно с заголовком

**article** представляет собой полную или автономную композицию в документе, странице, приложении или сайте и, в принципе, может независимо распространяться или повторно использоваться, например в синдикации. Это может быть сообщение на форуме, статья в журнале или газете, запись в блоге, комментарий пользователя, интерактивный виджет или гаджет или любой другой независимый элемент контента.

**main** определяет уникальный основной контент для веб-страницы, в нем не должно быть элементов, располагаемых на всех страницах сайта, например, заголовка, нижнего колонтитула и основной навигационной панели

**aside** предназначен для оформления боковых колонок, рекламных блоков, группирования навигационных элементов и прочего содержимого, которое лишь косвенно касается основного контента страницы.

**details** используется для информации, которую можно скрыть или показать по желанию пользователя

summary определяет видимый заголовок для элемента

**figure** заключает в себе потоковый контент, который может сопровождаться заглавием <figcaption>. Предназначен для контента, который можно удалить со страницы не изменяя смысла остальной информации.

**mark** представляет фрагмент текста в документе, помеченный цветом или выделенный подсветкой для справочных целей

**time** используется для обозначения конкретной даты и времени

**address** контактная информация элемента <article> или <body>

**8. Теги формы**

**form** определяет HTML-форму для ввода пользователем данных, которые впоследствии можно принять и обработать на стороне сервера

**input** создает пользовательские элементы формы

**label** устанавливает связь между определенной меткой, в качестве которой обычно выступает текст, и элементом формы

**textarea** определяет многострочное текстовое поле

**datalist** используется для автоматического выбора значения элемента

**select** используется для создания раскрывающегося списка, который появляется при щелчке мышью по элементу формы

**ortgroup** группирует пункты в раскрывающемся списке

**option** создает элемент раскрывающегося списка

**button** cоздает в документе интерактивные кнопки

**fieldset** служит для группировки элементов, размещенных внутри формы

**keygen** используется для шифрования значения элемента формы

**progress** определяет индикатор прогресса выполнения задачи, отображающего, какой процент задачи уже выполнен

**meter** определяет скалярное измерение в пределах известного диапазона или дробное значение

**output** определяет область, в которую выводится результат вычислений или действий пользователя (обычно рассчитывается с помощью скриптов) значение

**Атрибуты тега form**

**action** указывает расположение обработчика данных формы

**accept-charset** устанавливает кодировку, в которой сервер может принимать и обрабатывать данные формы

**method** указывает метод (get или post) отправки данных формы на сервер

**target** указывает место (фрейм или окно браузера) в которое должен быть загружен, указанный ресурс

**Значения атрибута target**

**\_blank** загружает страницу в новую вкладку браузера.

**\_self** загружает страницу в текущую вкладку.

**\_parent** Загружает страницу во фрейм-родитель; если фреймов нет, то это значение работает как **\_self**.

**\_top** отменяет все фреймы и загружает страницу в полном окне браузера; если фреймов нет, то это значение работает как **\_self**

**Атрибуты тега label**

**for** устанавливает связь между определённой меткой и элементом формы

<label for="name"> ... </label>

<input type=“radio" id="name">

**Атрибуты тега input**

**type** определяет вид элемента формы

**аccept** устанавливает фильтр на типы файлов, которые можно отправить через поле загрузки файлов

**alt** альтернативный текст для кнопки с изображением

**autocomplete** устанавливает или отменяет автозаполнение полей формы

**autofocus** автоматически переводит фокус на элемент формы сразу после полной загрузки страницы

**checked** применяется для изначальной установки поля с флажком или радио-кнопкой в активное состояние

**dirname** указывает уникальное имя значения направления вывода текста

**disabled** блокирует элемент формы, то есть делает его неактивным

**autocomplete** устанавливает или отменяет автозаполнение полей формы

**form** указывает одну или несколько форм, к которой принадлежит элемент

**formaction** указывает расположение обработчика данных формы

**formenctype** указывает способ кодирования данных формы, отправляемых на сервер

**formmethod** указывает метод отправки данных формы на сервер

**formnovalidate** отключает проверку данных формы (перед отправкой на сервер) на корректность

**list** с помощью идентификатора присоединяет к элементу формы элемент **<datalist>** с вариантами автодополнения

**max** указывает максимальное значение элемента

**pattern** задаёт шаблон для значения текстового элемента формы

**placeholder** указывает краткую подсказку (слово или короткую фразу), предназначенную для помощи пользователю с вводом данных

**readonly** поле ввода предназначено только для чтения

**required** указывает что данный элемент формы (или группа элементов) обязателен для заполнения перед отправкой формы

**value** присваивает элементу значение

1. **Теги списков, таблиц и изображений.**

**Список**

**<ul>** маркированный список

**<ol>** нумерованный список

**<li>** пункт списка маркированного или нумерованного

**<dl>** список определений

**<dt>** для добавления термина определения

**<dd>** для вставки определения

**Атрибуты <ol>**

**Атрибут start**

Позволяет задать номер списка, с которого будет начинаться нумерация

**Атрибут reversed**

Нумерация в списке становиться по убыванию

**Атрибут type**

Устанавливает вид маркера списка

**Таблица**

**<table>** выделяет элементы таблицы из содержимого документа

**<tr>** определяет строку

**<th>** определяет заголовок строки или столбца (необязательный)

**<td>** для создания ячейки

**<caption>** для добавления названия таблицы перед ней

**<thead>** для определения группы заголовочных строк

**<tfoot>** для вставки нижнего колонтитула таблицы

**<tbody>** для разбиения таблицы на разделы

**<colgroup>** определяет группу столбцов в таблице

**<col>** определяет столбец в группе стобцов

**Атрибут summary (тега table) п**редназначен для краткого описания таблицы или указания ее **п**редназначения. В отличие от тега **<caption>** содержимое summary никак не отображается и в браузере не выводится, однако может использоваться

поисковыми системами или речевыми браузерами

**Атрибуты тега td**

**colspan** объединяет несколько ячеек в строке

**rowspan** объединяет несколько ячеек в столбце

**headers** позволяет связать ячейки таблицы с заголовками

**Теги изображений**

**img** используется для вставки в изображений

**map** создаёт карту-изображение.

Карта изображений — это зоны обычного изображения, при нажатии на которые происходит активация ссылок связанных с конкретной зоной

**area** определяет активную зону карты-изображения

**canvas** определяет область, в которой можно рисовать при помощи скриптов

**Атрибуты тега img**

**src** (от англ. Sorce источник) указывает местоположение файла изображения

**alt** атрибут содержит текст, который браузер выводит вместо изображения, если графика не поддерживается, или показывает в виде подсказки, когда пользователь размещает над изображением мышиный курсор.

**height** высота изображения

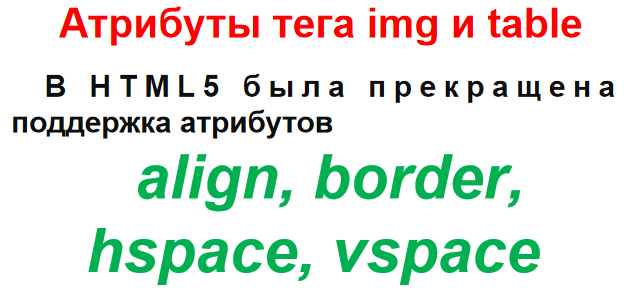
**width** ширина изображения

**srcse**t указывает список из одного или нескольких значений, разделённых запятыми, указывающих набор возможных изображений для отображения в браузере

**sizes** задает размеры изображения для разных макетов страницы

**usemap** определяет изображение в качестве клиентской карты-изображений

**ismap** указывает браузеру, что данное изображение является серверной картой-изображением



**Атрибуты тега map**

**name** присваивает уникальное имя элементу, через которое устанавливается связь с изображением

**<img usemap="#student">**

**<map name="student">**

**Атрибуты тега area**

**alt** альтернативный текст для области изображения.

**coords** координаты активной области

**download** сообщает о том, что эта ссылка должна быть использована для скачивания файла, и, когда пользователь нажимает на ссылку, ему будет предложено сохранить файл как локальный.

**href** задает адрес документа, на который следует перейти

**shape** форма области.

**target** имя фрейма, куда браузер будет загружать документ

**10. Теги для встраиваемого контента.**

**audio** добавляет, воспроизводит и управляет настройками аудиозаписи на веб-странице

**video** вставляет видео файл в HTML-документ. В настоящее время поддерживается три видео формата — MP4, WEBM, OGG

**source** используется для указания расположения медийных ресурсов (файлов) для медиа-элементов

**track** используется в качестве дочернего элемента для <audio> и <video>, определяя текстовые дорожки

**embed** используется для загрузки и отображения объектов (видеофайлов, флеш-роликов, некоторых звуковых файлов и т. д.), которые исходно браузер не понимает

**param** используется для определения параметров для плагинов, встроенных в элементы <object>

**object** используется для вставки в HTML-документ мультимедиа объектов (аудио и видео файлов, Java апплетов, ActiveX, PDF и Flash приложений)

**iframe** применяется для создания «плавающего» фрейма и вставки его в обычный HTML-документ

**Атрибуты тега audio**

**autoplay** звук начинает играть сразу после загрузки страницы.

**сontrols** добавляет панель управления к аудиофайлу.

**loop** повторяет воспроизведение звука с начала после его завершения.

**muted** отключает звук при воспроизведении музыки.

**preload** указывает как аудио должно загружаться при загрузке страницы. Атрибут игнорируется, если присутствует атрибут autoplay.

**src** указывает путь к воспроизводимому файлу

**Атрибуты тега video**

**poster** определяет изображение, которое будет показываться во время загрузки видеофайла или пока пользователь не нажмет кнопку воспроизведения видео

1. **Теги для отображения и форматирования текста**

**h1, h2, h3, h4, h5, h6** создание заголовков раздела

**p** создание нового абзаца

**br** для принудительного разрыва строки

**hr** определяет тематический разрыв контента на HTML странице, отображается как горизонт. Линия

**wbr** вызывает переход на новую строку, если текущая строка вышла за пределы окна браузера

**code** для выделения части текста моноширинным шрифтом

**em** для выделения текста курсивом, на который необходимо обратить внимание

**kbd** используется для обозначения текста, который набирается на клавиатуре или для названия клавиш

**pre** определяет блок в который помещают предварительно отформатированный текст

**dfn** выделяет специальный термин впервые появившийся в тексте

**cite** представляет название произведения

**abbr** указывает, что заключенный в нем текст является сокращенной формой длинного слова или фразы

**del** используется для выделения текста, который был удалeн

**s** определяет текст, который больше не является правильным или актуальным

**b** отображает вложенный в него текст полужирным шрифтом

**i** используется для выделения текста курсивом. Является элементом физического форматирования текста

**sup**  используется для создания верхнего индекса

**sub** используется для создания нижнего индекса

**u** создает подчеркивание текста

**small** уменьшает размер вложенного в него шрифта на одну единицу, кроме случаев когда исходный текст уже имеет наименьший допустимый размер

**bdi** гарантирует, что в обратном направлении будет написан только тот текст, который заключен в элементе

**bdо** используется для переопределения текущего направления текста

**ins** определяет текст, который был добавлен в новой версии документа

<ins cite = "http://redactor.com/info.html" datetime = "2018-05-30T10:00:00+03:00">

**ruby** предназначен для добавления небольшой аннотации сверху или снизу от заданного текста

**rt** выводит аннотацию сверху или снизу от текста, заключенного в элемент <ruby>

**rp** выводит альтернативный текст для браузеров, не поддерживающих элемент <ruby>. В остальных браузерах текст в теге <rp> не отображается

1. **Атрибуты HTML5. Специальные символы.**

**13. Глобальные атрибуты**

**14. Атрибуты обработчиков событий**

1. **Атрибуты тегов формы**

**16. Основы CSS3**

**17. Создание стилей. Классификация селекторов.**

**18. Псевдоклассы и псевдоэлементы**

**19. Механизм наследования стилей.**

**20. Единицы измерения значений свойств CSS**

**21. Каскадность стилей. Приоритет стилей**

**22. Конфликты стилей. Устранение конфликтов стилей.**

**23. Использование селекторов потомков**

**24. Форматирование текста на CSS3: абзацев, символов и слов**

**25. Использование веб-шрифтов. Использование службы Google Fonts.**

**26. Понятие блочной модели.**

**27. Форматирование таблиц**

**28. Форматирование веб-форм**

**29. Добавление фоновых изображений. Использование градиентных**

**фонов.**

1. **Построение навигационной системы сайта.**

**31. Форматирование ссылок**

**32. Использование ролловеров. Метод CSS-спрайт**

**33. CSS-переходы.**

**34. CSS-преобразования.**

**35. Функции 3D-преобразований.**

**36. CSS-анимация.**

**37. Форматирование списков перечисления.**

**38. Системы модульной верстки.**

**39. Система модульной верстки Skeleton.**

**40. Структурирование HTML-кода под модульную сетку и ее использо-вание.**

1. **Основы Grid-верстки.**

**42. Медиазапросы. Создание адаптивного дизайна**

**43. Основы Flexbox-верстки.**

**44. Основы Sass**

**45. Основы Sass. Наследование свойств**

**46. Установка Sass**

**47. Примеси Sass**

**48. Переменные Sass**

**49. Операторы Sass**

**50. Основы Sass. Вложенные правила**

**51. Основы XML**

**52. Правила документов XML.**

**53. Элементы и атрибуты XML**

**54. Стандарты XML**

**55. Cхема DTD**

**56. XML Schema**

**57. Сущности XML**

**58. Язык преобразований XSLT**

**59. Сортировка данных в XML**

**60. Язык переадресации XPath**

**61. Язык запросов XQuery**

**62. Введение в XLinks. Применение XPointer**

**63. Основы SVG**

**64. Атрибуты SVG**

**65. Анимация SVG**

**66. Основные элементы SVG**

**67. Работа с текстом SVG**

**68. Обзор и версии языков разметки веб-страниц**