**Дополнительная профессиональная программа**

**профессиональной переподготовки**

**«Разработка веб-приложений»**

**Дисциплина «Программирование на языке JavaScript»**

**Рабочая тетрадь**

|  |
| --- |
| JavaScript был создан программистом Brendan Eich из Netscape и пред-ставлен в декабре 1995 года под названием LiveScript. Довольно быстро он был переименован в JavaScript, хотя официальным названием JavaScript является ECMAScript. ECMAScript разрабатывается и поддер-живается Международной организацией ECMA (Европейская ассоциация производителей компьютеров).  Что такое JavaScript?  1) JavaScript – язык сценариев, или скриптов. Скрипт представляет собой программный код – набор инструкций, который не требует предваритель-ной обработки (например, компиляции) перед запуском. Код JavaScript интерпретируется движком браузера во время загрузки веб-страницы. Ин-терпретатор браузера выполняет построчный анализ, обработку и выпол-нение исходной программы или запроса.  2) JavaScript – объектно-ориентированный язык с прототипным наследо-ванием. Он поддерживает несколько встроенных объектов, а также позво-ляет создавать или удалять свои собственные (пользовательские) объекты. Объекты могут наследовать свойства непосредственно друг от друга, об-разуя цепочку объект-прототип. |

**Практическое занятие № 1**

**ТЕМА: Введение в JavaScript**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Теоретический материал: первая программа на языке JavaScript** | | | | |
| Web-документ, отображаемый браузером, – это результат исполнения программ, созданных на разных языках. Для описания структуры используется язык разметки (HTML), для описания внешнего вида – язык стилей (CSS). Для описания поведения документа, его реакции на действия пользователя используется язык сценариев (JavaScript).  **Как подключить JavaScript-код к HTML-странице**  JavaScript-код исполняется непосредственно в браузере. В любой современный браузер встроен интерпретатор JavaScript. Следовательно, выполнение программы зависит от того, когда этот интерпретатор получает управление. Опишем несколько способов размещения кода JavaScript на странице:   1. Непосредственно в теговом контейнере <BODY>...</BODY> - теле HTML-страницы:   <body>  …  <script > команды скрипта</script>  …  </body>   1. В теговом контейнере <HEAD>...</HEAD> - если код скрипта представляет собой функцию (то есть набор команд), которая вызывается в ответ на какое-либо событие (например, при нажатии на кнопку, которая размещена на HTML-странице).   <head>  …  <script type=”text/javascript”> Здесь находятся команды сценария </script>  …  </head>   1. Во внешнем файле. По аналогии с тем, как стили подключаются к станице с помощью элемента link, сценарии подключаются с помощью элемента script, только файл имеет расширение не .css, а .js. В наших примерах мы будем использовать преимущественно этот способ подключения скрипта.   <head>  …  <script type=”text/javascript” src=”my.js” > </script>  …  </head>   1. Обработчик события указывается прямо в теге, без заключения в теги <script > </script>   <input type="button" value="Нажать" onClick="window.alert('Hажмите еще раз')">    Существует несколько способов определения момента запуска сценария (программного кода, написанного на языке JavaScript). Вот некоторые из них:  1. При загрузке документа;  2. Сразу после загрузки документа;  3. В ответ на действия пользователя.  Чаще всего мы будем пользоваться третьим способом.  **Синтаксис языка**   * JavaScript –**зависит от** **регистра**. Имена JavaScript и Javascript - разные имена!! Все ключевые слова используют только нижний регистр. * Операторы разделяются точкой с запятой, которую можно опустить, если оператор заканчивается символом новой строки (Enter). * Комментарии:   // однострочный комментарий,  /\* \*/ ..многострочный комментарий  **Первая программа на языке JavaScript: вывод строки «Hello, world!»**   1. **Использование метода alert(‘сообщение’)**   Метод *alert* используется для вывода простейшего диалогового окна, содержащего текст сообщения и единственную кнопку "Ok". Программа выводит сообщение и ожидает нажатия кнопки. После нажатия на кнопку, программа начинает выполняться дальше.   1. **Использование метода document.write()**   JavaScript это объектно-ориентированный язык. Основной единицей в объектно-ориентированном языке является объект, который объединяет в себе данные (свойства) и средства обработки этих данных (методы). Если говорить образно, то объекты – это «существительные», свойства объекта – это «прилагательные», а методы объекта – это «глаголы». Значения свойств объектов можно изменять.  Про JavaScript говорят, что в нем все объект. А именно: объектами являются окно, в котором открывается документ, сам документ, все элементы документа и даже свойства этих элементов. Есть также специальные встроенные объекты. Для упорядочивания огромного количества объектов создатели браузеров придумали объектную модель документа. Эта модель является структурой организации объектов на странице.  Объект document соответствует всему HTML-документу.  Изучим один метод этого объекта, позволяющий динамически формировать документ.  Метод document.write(‘строка html-кода’) - выводит строку в окно документа.  Метод document.writeln (“строка html-кода”) - выводит строку в окно документа, в конце выводится символ пробела.  Метод, применяемый к объекту, пишется после имени объекта через точку.  Содержимое строки должно быть в кавычках или это может быть объединение (сумма) нескольких строк или строковых переменных.  Строка может содержать элементы разметки страницы (теги и их содержимое).  Метод исполняется в процессе загрузки документа. | | | | |
| Типы данных  Переменные не имеют строгой типизации. Объявляются с помощью оператора var, который можно опускать, за исключением объявления локальных переменных в теле функции. Возможно объявление c одновременной инициализацией, например:  var s = 123 //объявляется целочисленная переменная x, имеющая десятичное значение 123  var d=3.14 //объявляется переменная с плавающей точкой (десятичная дробь)  var str1=’Строковая переменная’  var p=true //объявляется логическая переменная  Тип переменной может изменяться в процессе выполнения программы. Если в выражении содержатся и числовые и строковые переменные, то числовые переменные автоматически приводятся к строковому виду.  Математические операции    Операции сравнения    Логические операции    Операторы присваивания | | | | |
| **Пример 1.1** | | | | |
| ***Задача:*** | | | | |
|  | | | Вывести с помощью метода alert сообщение “Hello, world!” | |
| ***Решение:*** | | | | |
|  | | | Проект состоит из двух файлов: первый файл index.html, в котором в строке 3 подключается второй файл script.js – сценарий на языке JavaScript  Файл index.html:    Файл script.js:    В единственной строке скрипта вызывается метод alert, которому передается текстовое сообщение Hello, world. | |
| ***Ответ:*** | | | | |
|  | | | При открытии файла index.html браузером выводится диалоговое окно, содержащее строку Hello, world!    В зависимости от браузера диалоговое окно может иметь разный внешний вид. | |
| **Пример 1.2** | | | | |
| ***Задача:*** | | | | |
|  | | | Вывести с помощью метода document.write сообщение “Hello, world!” | |
| ***Решение:*** | | | | |
|  | | | Проект состоит из двух файлов: первый файл index.html, в котором в строке 4 подключается второй файл script.js – сценарий на языке JavaScript  Файл index.html:    Файл script.js:    В единственной строке скрипта вызывается метод document.write, которому передается текстовое сообщение Hello, world. | |
| ***Ответ:*** | | | | |
|  | | | При открытии файла index.html браузером выводится следующее: | |
| **Пример 1.3** | | | | |
| ***Задача:*** | | | | |
|  | Вывести на страницу сайта наперед заданное в сценарии значение переменной | | | |
| ***Решение:*** | | | | |
|  | Проект состоит из двух файлов: первый файл index.html, в котором в строке 5 подключается второй файл script.js – сценарий на языке JavaScript  Файл index.html:    Файл script.js:    В строке 1 скрипта объявляется переменная X, которой присваивается значение 25. В строке 2 скрипта выводится текстовая строка с помощью метода document.writeln, добавляющего пробел после сообщения. В третьей строке выводится переменная X (при выполнении скрипта мы увидим значение, которое эта переменная принимает). | | | |
| ***Ответ:*** | | | | |
|  | При открытии файла index.html браузером выводится следующее: | | | |
| **Пример 1.4** | | | | |
| ***Задача:*** | | | | |
|  | Продемонстрировать применение основных математических операций с переменными | | | |
| ***Решение:*** | | | | |
|  | Файл index.html аналогичен примеру 1.3  Файл script.js выглядит следующим образом:    В строках 1 и 2 объявлены переменные X и Y, а также заданы их значения. Дальнейший код демонстрирует применение различных математических операций | | | |
| ***Ответ:*** | | | | |
|  |  | | | |
| **Задание 1.1** | | | | |
| ***Задача:*** | | | | |
|  | | Вывести с помощью метода alert свое имя и фамилию | | |
| ***Решение:*** | | | | |
|  | |  | | |
| ***Ответ:*** | | | | |
|  | |  | | |
| **Задание 1.2** | | | | |
| ***Задача:*** | | | | |
|  | | Вывести с помощью метода document.write свое имя и фамилию | | |
| ***Решение:*** | | | | |
|  | |  | | |
| ***Ответ:*** | | | | |
|  | |  | | |
| **Задание 1.3** | | | | |
| ***Задача:*** | | | | |
|  | | | | В сценарии на языке JavaScript объявить переменную и инициировать ее значением своего возраста. Вывести значение переменной с помощью метода alert. |
| ***Решение:*** | | | | |
|  | | | |  |
| ***Ответ:*** | | | | |
|  | | | |  |
| **Задание 1.4** | | | | |
| ***Задача:*** | | | | |
|  | | | | В сценарии на языке JavaScript объявить три переменные X, Y и Z и инициировать их, соответственно, значениями года, месяца и дня рождения. С помощью метода document.write отобразить на странице index.html следующие величины: 1) X+Y+Z; 2) X-Y+Z; 3) X/Y-Z; 4) Z/X\*Y; 5) (X%Y)+Z |
| ***Решение:*** | | | | |
|  | | | |  |
| ***Ответ:*** | | | | |
|  | | | |  |