УВАЖАЕМЫЕ СТУДЕНТЫ! Законспектируйте в своей рабочей тетради по дисциплине приведенную лекцию (объемом 4-5 страницы).

Результаты работы, фотоотчет, предоставить преподавателю на e-mail: igor-gricenko-95@mail.ru **в течении ТРЕХ дней.**

При возникновении вопросов по приведенному материалу обращаться по следующему номеру телефона: (072)132-63-42

ВНИМАНИЕ!!! При отправке работы, не забывайте указывать ФИО студента, наименование дисциплины, дата проведения занятия (по расписанию).

Возможности языка JavaScript.

Основные особенности JavaScript

JavaScript — это относительно простой объектно-ориентированный язык, предназначенный для создания небольших клиентских и серверных приложений для Internet. Программы, написанные на языке JavaScript, включаются в состав HTML-документов и распространяются вместе с ними. Программы просмотра (браузеры – от англ. browser) типа Netscape Navigator и Microsoft Internet Explorer распознают встроенные в текст документа программы-вставки (script-коды) и выполняют их. Таким образом, JavaScript — интерпретируемый язык программирования. Примерами программ на JavaScript могут служить программы, проверяющие введенные пользователем данные или выполняющие какие-то действия при открытии или закрытии документа. Такие программы могут реагировать на действия пользователя — нажатие кнопок "мыши", ввод данных в экранной форме или перемещение "мыши" по странице. Более того, JavaScript-программы могут управлять самим браузером и атрибутами документа.

Язык JavaScript, будучи схожим по синтаксису с языком Java, за исключением объектной модели, в то же время не обладает такими свойствами, как статические типы данных и строгой типизацией. В JavaScript, в отличие от Java,

понятие классов не является основой синтаксических конструкций языка. Такой основой является небольшой набор предопределенных типов данных, поддерживаемых исполняемой системой: числовые, булевские и строковые; функции, которые могут быть как самостоятельными, так и методами объектов (метод в терминологии JavaScript — не что иное, как функция/подпрограмма); объектная модель с большим набором предопределенных объектов со своими свойствами и методами, а также правилами задания в программе пользователя новых объектов.

Для создания программ на JavaScript не требуется никаких дополнительных средств— необходим лишь браузер, поддерживающий язык JavaScript соответствующей версии и текстовый редактор, позволяющий создавать HTML-документы. Так как программа на JavaScript встраивается непосредственно в текст HTML-документа, вы можете немедленно увидеть результаты своей работы во время просмотра документа браузером и при необходимости внести изменения.

1. *Возможности языка JavaScript*

С его помощью можно динамически управлять отображением и содержимым HTML-документов. Можно записывать в отображаемый на экран документ произвольный HTML-код в процессе синтаксического анализа загруженного браузером документа. С помощью объекта Document можно генерировать документы "с нуля" в зависимости от предыдущих действий пользователя или каких-либо иных факторов.

JavaScript позволяет контролировать работу браузера. Например, объект Window поддерживает методы, позволяющие выводить на экран всплывающие диалоговые окна, создавать, открывать и закрывать новые окна браузера, задавать режимы прокрутки и размеры окон и т.д.

JavaScript позволяет взаимодействовать с содержимым документов. Объект Document и содержащиеся в нем объекты позволяют программам читать части HTML-документа и иногда взаимодействовать с ними. Сам текст прочитать невозможно, но можно, например, получить список гипертекстовых ссылок, имеющихся в данном документе. На текущий момент широкие возможности взаимодействия с содержимым документов обеспечивает объект Form и объекты, которые он может содержать: Button, Checkbox, Hidden, Password, Radio, Reset, Select, Submit, Text и Textarea.

JavaScript обеспечивает взаимодействие с пользователем. Важной особенностью этого языка является реализованная в нем возможность определять обработчики событий — произвольные порции кода, которые выполняются при наступлении конкретных событий (обычно действий пользователя). JavaScript позволяет использовать в качестве обработчиков событий любые новые предварительно заданные функции. Например, можно написать программу, которая выведет в строке состояния специальное сообщение, если пользователь установит указатель "мыши" на гипертекстовую ссылку, или выведет на экран диалоговое окно с запросом на подтверждение выполнения некоторого действия, или осуществит проверку введенного пользователем значения и в случае ошибки ввода выдаст соответствующую диагностику и заставит ввести правильное значение.

JavaScript дает возможность выполнять произвольные математические вычисления. Кроме того, этот язык имеет развитые средства работы со значениями даты и времени. JavaScript был создан в качестве альтернативы CGI-программам и языку сценариев Perl, а также в качестве дополнения м в ряде случаев альтернативы языку Java.

Ниже приведена таблица, в которой проводится сравнение Java и JavaScript:

JavaScript	**Java**
Исходный код программ встраивается непосредственно в HTML-документ либо загружается из независимых файлов.	*Исходный код программ* не распространяется с приложением - апплетом. Апплеты загружаются с сервера из независимых файлов.
Программа выкладывается на сервер в виде исходного кода в текстовой форме и в дальнейшем *интерпретируется* (без предварительной компиляции) браузером после загрузки ее с сервера.	Программа *компилируется* в машинно-независимый байтовый код Java-код, после чего выкладывается на сервер. Браузер (виртуальная Java-машина) исполняет Java-код.
Объектный. Можно программировать как без использования объектного программирования, так и используя предопределенные встроенные классы. Имеется теоретическая возможность расширения этих классов, но реально она никогда не используется.	*Объектно-ориентированный*. Программировать без использования объектного программирования нельзя. Апплеты состоят из классов с возможностью иерархического наследования по традиционной схеме наследования. Использование наследования и полиморфизма – основа программирования в Java.
В отличие от подавляющего большинства языков объектного программирования в JavaScript *структура объектов* не	*Структура объектов* полностью задается на этапе компиляции их классов.

задается однозначно устройством класса, а *является динамической* и может меняться на этапе выполнения программы. Объекты могут динамически получать новые поля и методы или изменять любые параметры старых.	
Свободная типизация: элементарные типы данных переменных не описываются, при присваивании тип левой части всегда определяется по результату присваивания (т.е. по правой части)	*Строгая типизация*: типы данных любых переменных должны быть описаны перед использованием, тип левой части должен совпадать с типом правой (за редкими исключениями, когда работает автоматическое приведение типа результата к типу левой части)
Динамическое связывание кода с объектами: ссылки на объекты проверяются во время выполнения программы.	*Статическое связывание кода с объектами*: ссылки на объекты должны существовать на момент компиляции

2. ***Основные типы данных***

Значения переменных, функций и выражений бывают следующих типов:

1. целые численные:

в десятичной системе единиц: 0, 29, 70, -147 и т.п.;

в 16-ричной: 0x70 или 0x70, 0XFA7D0 и т.п.;

в 8-ричной: 070, 0710 (***Внимание!!!*** Ведущий ноль воспринимается как символ 8-ричного числа) и т.п.

2. вещественные численные:

0.0, -2.9, 0.7E1, 14.7e-2, 1e+308 (максимальное вещественное число), 1.001e-305 (минимальное по модулю вещественное число, отличное от нуля) и т.п.;

3. логические (булевские): true и false;
4. строковые: "Привет, все!", "ОК", 'Слово "Привет!" с кавычками внутри строки', "Другой вариант 'Привет' с кавычками внутри строки" и т.п. (допускаются оба типа кавычек и многократное использование таких пар внутри друг друга). Специальные символы обозначаются комбинацией символа \ и буквы (или последовательности цифр), например: \b — "забой", \n — перевод на новую строку, \" — "кавычка".
5. null — специальное значение для обозначения “пустого множества” значений.

1. ***Переменные. Приведение типов***

Глобальные переменные можно вводить в любом месте текста программы путем простого присваивания значения. Но необходимо, чтобы переменная была определена до того момента, когда вызывается при исполнении:

var myVariable=0.1

var B=false

и т.п. При этом тип переменной приводится к типу присваиваемого значения, причем в последующем этой же переменной можно присвоить значение другого типа:

myVariable="Теперь это текстовая переменная"

При задании переменной использование зарезервированного слова var не обязательно, но желательно, т.к. помогает при использовании отладчика фирмы Microsoft и делает текст программы более структурированным. На деле вместо переменной в текущем объекте window создается новое поле с таким именем. В функциях при задании локальных переменных использование var обязательно (иначе будет создана глобальная переменная).

При наличии численных и строковых значений в одном выражении идет приведение к строковому типу. Значением переменной

a=7+"раз отмерь,"+1+"раз присвой"

будет строка "7 раз отмерь, 1 раз присвой".

Стоит отметить, что существуют также функции parseFloat и parseInt, которые осуществляют преобразование из строкового значения в численное.

Идентификатором переменной может быть последовательность символов из набора букв от "A" до "Z", от "a" до "z", цифр от "0" до "9", а также символ подчеркивания "_". При этом первым символом имени не может быть цифра, а заглавные и строчные буквы отличаются (т. е. имена MyVariable и myvariable относятся к разным переменным).

Кроме глобальных переменных в функции или другом блоке кода можно определить локальные, для них областью видимости будет только функция (без кода), в которой они определены:

var myLocalVariable=0.

2. *SCRIPT-вставки в HTML-документе*

Для встраивания программы на JavaScript в HTML — файл используются теги <script> и </script>. При этом результат работы можно увидеть сразу и при необходимости внести изменения.

```
<html>

<head>

<script language="JavaScript">

document.write("Hello,world!<p>")

</script>

</head>

<body>

It was dynamically created text.

</body>

</html>
```

Будет сформирован текст:

Hello, world!

It was dynamically crested text.

Заметим, что внутри тегов <script> и </script> может содержаться любое число конструкций языка JavaScript.

Некоторые старые браузеры не поддерживают язык JavaScript и поэтому, чтобы скрыть от них вставки JavaScript, в программу добавляют следующий комментарий:

<!-- скрываемый текст -->

Пример:

<html>

<head>

<script language="JavaScript">

<!-- hidden text

document.write("Hello from JavaScript")

//end of hidden text -->

</script>

</head>

<body>

The end

</body>

</html>

Вставки могут быть и внутри <body>.