

Тема: Основы языка JavaScript

Вопрос 1. Назначение и основные понятия JavaScript

1995 г. компания Netscape Corporation

Основная идея JavaScript состоит в возможности изменения значений атрибутов HTML-контейнеров и свойств среды отображения в процессе просмотра HTML-страницы пользователем. При этом перезагрузки страницы не происходит

JavaScript стандартизован ECMA (European Computer Manufacturers Association - Ассоциация европейских производителей компьютеров)

Сценариями называются небольшие программы, предназначенные для выполнения определенных операций и функций.

Задачи:

- Создание анимации;
- Аудио сопровождение страницы в соответствии с контекстом;
- Динамическое изменение вида и содержимого страницы;
- Проверка и передача данных, введенных пользователем в поля формы.

Коды сценариев JS неразрывно связаны с HTML – кодом Web – страницы и предназначены для динамического взаимодействия с элементами Web – страницы, созданными с помощью кодов HTML.

Добавления сценариев в код HTML:
<SCRIPT></SCRIPT>.

Сценарии можно добавлять

- основной раздел Web – страницы (<BODY>.....</BODY>)
- в раздел заголовка (<HEAD>.....</HEAD>)
- в отдельный файл с расширением .js

Путь к файлу указывают в атрибуте SRC
дескриптора <SCRIPT>:

```
<SCRIPT SRC='scripts/MyScript.js'  
  LANGUAGE='JavaScript'>  
</SCRIPT>
```

Рамка 1

Документ 1 (разделы, абзацы, фрагменты текста)

Рамка 2

Документ 2 (разделы, абзацы, фрагменты текста)

Таблица

Ячейка1

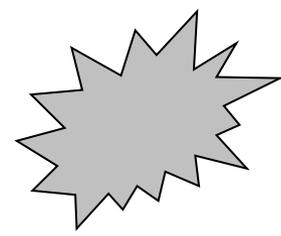
Ячейка2

Форма

Поле ввода

Кнопка 1

Рисунок



Объектам верхнего уровня автоматически присваиваются стандартные имена:

- `window` – окно;
- `document` – документ в окне или рамке;
- `form` – форма.

Для любого элемента Web-страницы можно задать уникальное имя, присвоив его атрибуту ID и/или NAME.

Например, присваиваем элементу основного раздела страницы имя body:

<BODY ID='body' NAME='body'>

**window.body.bgcolor='red' – цвет фона
страницы**

Объект характеризуется: свойством, методом,
событием

Свойства определяют вид и особенности (поведение) объекта (ширина и высота рамки элемента, цвет, текст).

Методы представляют собой встроенные функции изменения свойств объекта.

Например, с объектом "документ" связаны методы `open()`, `write()`, `close()`.

События устанавливают взаимосвязь между действием пользователя над объектом и внешней функцией обработки события. Например, щелчок мышью по кнопке вызывает событие **ONCLICK** и функцию, назначенную этому событию.

Вопрос 2. Размещение кода на HTML-странице

Способы функционального применения
JavaScript:

- гипертекстовая ссылка (схема URL);
- обработчик события (handler);
- вставка (контейнер SCRIPT).

1. URL-схема JavaScript

```
<A HREF="JavaScript:JavaScript_код">
```

```
...</A>
```

```
<IMG SRC="JavaScript:JavaScript_код">
```

"JavaScript_код" обозначает программы-обработчики на *JavaScript*, которые вызываются при выборе гипертекстовой ссылки в первом случае и при загрузке картинки - во втором.

ПРИМЕР

```
<html>
```

```
<head><title>Сообщения в строке  
состояния</title> </head>
```

```
<body>
```

```
<A
```

```
  HREF="JavaScript:alert('Внимание!!!');">
```

```
  Внимание!!!</A>
```

```
</body>
```

```
</html>
```

2. Обработчики событий (handler)

указываются в атрибутах контейнеров, с которыми эти события связаны. Например, при нажатии на кнопку происходит событие click:

```
<FORM>
```

```
<INPUT TYPE=button VALUE="Кнопка"  
onClick="window.alert('intuit');"></FORM>
```

ПРИМЕР

```
<html>
```

```
<head><title>Сообщения в строке  
состояния</title> </head>
```

```
<body>
```

```
<FORM>
```

```
<INPUT TYPE=button VALUE="Кнопка"  
onClick="window.alert('intuit');">
```

```
</FORM>
```

```
</body>
```

```
</html>
```

3. Вставка (контейнер SCRIPT - принудительный вызов интерпретатора)

Код можно разместить либо внутри контейнера HEAD, либо внутри BODY, либо в отдельном файле

Функции контейнера SCRIPT:

1. Размещение кода в HTML – документе;
2. Условная генерация HTML – разметки на стороне браузера.

Вопрос 3 . События и функции обработки событий

Действия пользователей, в ответ на которые происходит изменение страницы, называются **событиями**.

Событие	Описание	Дескрипторы, поддерживаю щие событие
ONCLICK	Щелчок мышью на элементе	Большинство дескрипторов
ONDBCLICK	Двойной щелчок мышью на элементе	Большинство дескрипторов

ONMOUSEOVER

**Пользовате
ль наводит
указатель
мыши на
элемент**

**Большин
ство
дескрип
торов**

ONMOUSEOUT

**Пользовате
ль убирает
указатель
мыши с
элемента**

**Большин
ство
дескрип
торов**

ONFOCUS	Пользователь выбирает элемент с помощью клавиши <Tab>	<A>, <AREA> <INPUT>, <SELECT>, <BUTTON>, <TEXTAREA>
ONBLUR	Элемент теряет выделение при следующем нажатии клавиши <Tab>	<A>, <AREA> <INPUT>, <SELECT>, <BUTTON>, <TEXTAREA>

ONSELECT	Пользователь выделяет текст элемента	<INPUT>, <TEXTAREA>
ONCHANGE	Пользователь изменяет текст элемента	<INPUT>, <SELECT>, <TEXTAREA>
ONSUBMIT	Пользователь щелкнул по кнопке «Подача запроса»	<FORM>

ONRESET	Пользователь щелкнул по кнопке «Сбросить»	<FORM>
ONLOAD	Web – страница загружается в окно обозревателя или в рамку	<BODY>, <FRAMESET>, <FRAME>
UNONLOAD	Web – страница замещается другой страницей в окне обозревателя или в рамке	<BODY>, <FRAMESET>, <FRAME>

Создание пользовательских функций

Функции в сценариях используются для достижения следующих целей:

- для динамического добавления или удаления элементов Web – страницы, изменения их свойств;**
- для выполнения вычислений и возвращения результата по месту вызова функции (return).**

Пользовательские функции создаются с помощью ключевого слова **function**:

<SCRIPT LANGUAGE='JavaScript'>

function ИмяФункции(список аргументов) {

код функции

}

другие функции ...

</SCRIPT>

Вызов функции:

Имя_функции(список аргументов);

ПРИМЕР

Вопрос 4. Программирование свойств окна браузера

Класс объектов Window — это самый старший класс в иерархии объектов JavaScript.

К нему относятся объекты Window и Frame.

Свойства	Методы	События
status defaultStatus location history navigator	open() close() focus()	load unload

Поле статуса

Internet Explorer window: InTuition.ru: Интернет-Университет Информационных Те...

Address bar: http://www.intuit.ru/all_specialities.html

Website header: ИНТЕРНЕТ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Navigation: Специальности | Все курсы | Расписание | Учебники | Н...

Login section: Вход для студентов

Fields: Ваше имя (ник), пароль

Callout: Поле статуса (status bar)

Specialties section: Все специальности

Links: Алгоритмы и сложность, Архитектура вычислительных систем

Footer: http://www.intuit.ru/cgi-bin/go

Программа на JavaScript имеет возможность работать с этим полем как с изменяемым свойством окна.

С ним связаны два разных состояния:
`window.status` – происходят события;
`window.defaultStatus` – нет никаких событий.

Способы управления текстом строки состояния:

1. **defaultStatus=text** – устанавливает текст строки состояния по умолчанию.
2. **status=text** – используется в функциях обработки событий для установки контекстно-зависимых сообщений.

Данные свойства вызываются либо для объекта окна обозревателя по его имени, либо для текущего окна с ключевым словом **self** или **window**.

Пример:

ONFOCUS = “self.status= ’Введите свой адрес электронной почты’”

ONFOCUS = “window.status=’Введите свой адрес электронной почты’”

ONFOCUS = “MyWindow.status=’Введите свой адрес электронной почты’”

В случае использования свойства `status` для обработки событий `ONMOUSEOVER` и `ONMOUSEOUT` функция должна завершаться командой `return true`:

```
ONMOUSEOVER = "self.status= 'Моя  
электронная почта'; return true"
```

ПРИМЕР ПРИМЕР1

.....

```
<body>
```

```
<script>
```

```
self.defaultStatus="Монмарт"
```

```
</script>
```

```
<h1 align=center> Пример управления  
текстом строки состояния</h1> <center>
```

```
<A HREF=1.jpg >
```

```
<img src=1.jpg width=150 hight=200
```

```
  ONMOUSEOVER="self.status='щелкните,  
чтобы посмотреть большую фотографию';  
return true; ></A>
```

.....

Методы управления окнами.

- **alert();**
- **confirm();**
- **prompt().**

**В JavaScript используются три
встроенных диалоговых окна. Заголовки
и набор кнопок в этих окнах постоянны.
Текст сообщения задается аргументом
функции открытия диалогового окна.**

- **alert(“сообщение”)** – открывает диалоговое окно с текстом сообщения и единственной кнопкой **ОК**.

Это диалоговое окно применяется для показа предупреждений или информационных сообщений, не требующих от пользователя принятия каких-либо решений.

```
<A HREF="javascript:window.alert('Внимание')">
```

Повторите запрос!

```
</A>
```

- **confirm**(“сообщение”) – открывает диалоговое окно с текстом сообщения и двумя кнопками – **ОК** и **Cancel**.

В зависимости от выбора кнопки функция **confirm** возвращает значение **TRUE** (ОК) или **FALSE** (Cancel).

- **prompt**(“сообщение”, “текст по умолчанию”) – открывает окно для ввода данных пользователем.

Содержит кнопки **ОК**, **Cancel** и текстовое поле.

В текстовом поле по умолчанию отображается текст, заданный во втором аргументе функции **prompt**. Можно оставить текстовое поле пустым – “”.

Если второй аргумент пропустить, то в текстовом поле отобразится текст **undefined**. После щелчка по кнопке **ОК** функция **prompt** возвращает значение текстового поля, а после щелчка по кнопке **Cancel** – значение **false**.

[ПРИМЕР](#) [ПРИМЕР3](#)