



# Введение в JavaScript

# Рекомендуемая литература

1. **Аллен В.** - JavaScript. Справочник;
2. **Вагнер Р., Вайк А.** - JavaScript. Энциклопедия пользователя;
3. **Дронов В.** - JavaScript в Web-дизайне;
4. **Дунаев В.** - Самоучитель JavaScript;
5. **Монкур М.** - Освой самостоятельно JavaScript за 24 часа;
6. **Мэрдок К.** - JavaScript. Наглядный курс создания динамических Web-страниц;
7. **Николенко Д.** - Практические занятия по JavaScript;

# Недостатки HTML

- Поскольку HTML - это всего лишь язык разметки, он не позволяет пользователю управлять внешним видом документа, а автоматически отображает страницу так, как было определено автором на этапе разработки. Вторым недостатком HTML является то, что он работает только со статическими элементами

# Как придать интерактивность документу

- Для придания документу интерактивности необходимо внедрить в него *сценарий*, написанный на одном из языков *клиентских скриптов* (JavaScript, VBScript и т.д.). Слово "клиентский" обозначает, что сценарий выполняется только на компьютере клиента без какого-либо участия сервера

# Для чего применяются сценарии

- Сценарии позволяют динамически управлять элементами страницы, а также изменять их при взаимодействии с ней пользователя (например, изменить текст в текстовом поле при щелчке по кнопке)
- Сценарий может содержать как одну строку, так и большой листинг. Он выполняется в браузере без каких-либо дополнительных программных средств



# Что можно сделать при помощи сценариев

- Сценарии позволяют выполнять следующие действия:
  1. Определять используемый браузер и настраивать страницу в соответствии с ним;
  2. Динамически создавать документ;
  3. Совместно с CSS создавать динамические страницы (технология DHTML – Dynamic HTML);
  4. Проверять правильность заполнения пользовательских форм до отправки их на сервер;
  5. Управлять размещением и поведением окон браузера;
  6. Создавать интерактивные элементы (такие как графические баннеры, автоматически изменяющиеся надписи и т.д.);
  7. Использовать объекты операционной системы (такие как реестр, файловая система и т.д.);



# Центр администрирования библиотеки

[Управление новостями](#) | [Управление пользователями](#) | [Управление библиотекой](#) | [Службная информация](#)

Управление но

- Управление разделами
- Управление подразделами
- Управление авторами
- Управление издательствами
- Управление литературой

Операции: [Добавить новость](#) Редактировать новость Удалить н

Новостей: 0

Запрошенные данные не обнаружены

# Языки создания сценариев

- Наиболее популярным языком создания сценариев является язык JavaScript. Кроме него могут использоваться такие языки как VBScript (подмножество языка Visual Basic) и Jscript (интерпретатор языка JavaScript фирмы Microsoft)
- Язык JavaScript является наиболее предпочтительным, поскольку поддерживается всем популярными браузерами. Язык же VBScript поддерживается только браузером Microsoft Internet Explorer



# Встраивание сценариев в HTML-документ

- Встроить сценарий в HTML-документ можно четырьмя способами:
  1. Поместить текст сценария внутрь тега **<script></script>**;
  2. Сделать ссылку на файл, содержащий текст сценария;
  3. Использовать операторы языка в качестве атрибутов тегов;
  4. Определить обработчик события в нужном теге

# Использование тега <script>

- Для внедрения сценариев на страницу язык HTML содержит специальный тег-контейнер **<script></script>**, внутри которого размещается текст сценария. Данный тег имеет следующие атрибуты:

1. **language** – задает язык сценариев. В качестве значения используются названия языков

Пример

```
<script language="javascript">
```

*текст скрипта*

```
</script>
```

2. **type** – задает тип MIME (Multipurpose Internet Mail Extension – многоцелевые расширения электронной почты). Представляет собой набор стандартов для передачи информации посредством электронной почты. В качестве значения указывается зарезервированное значение MIME. Для языка JavaScript оно задается как **text/javascript**, для VBScript - **text/vbscript**, а для Jscript - **text/Jscript**

Пример

```
<script type="text/javascript">
```

*текст скрипта*

```
</script>
```

- Документ может содержать несколько тегов <script>, расположенных в любом месте. Все они последовательно обрабатываются интерпретатором языка по мере отображения страницы. В связи с этим ссылка на переменную, определенную в сценарии, размещенном в конце документа, может привести к генерации ошибки интерпретатора при обращении к такой переменной из сценария, расположенном в начале документа. Поэтому рекомендуется размещать сценарии с глобальными функциями и переменными в разделе <head> документа

# Что делать, если в браузере отключена поддержка скриптов

- Некоторые пользователи из соображений безопасности отключают поддержку скриптов в своем браузере. В этом случае сценарий, внедренный на страницу, работать не будет. Специально для таких случаев был введен тег **<noscript></noscript>**. Если браузер поддерживает скрипты, содержимое этого тега игнорируется

```
<script type="text/javascript">
```

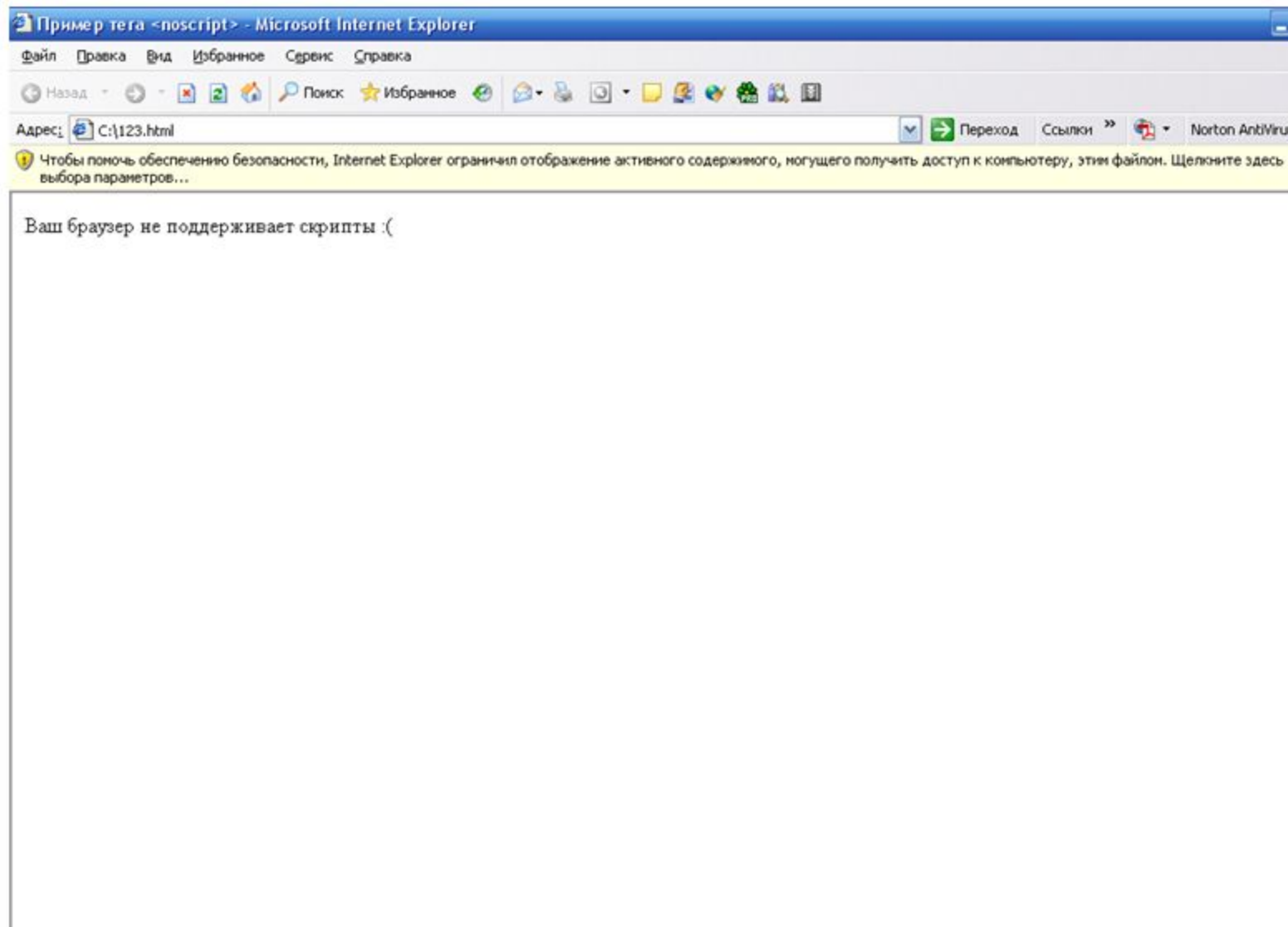
```
    текст скрипта
```

```
</script>
```

```
<noscript>
```

```
Ваш браузер не поддерживает скрипты :(
```

```
</noscript>
```



# Что делать, если браузер не поддерживает скрипты

- Если используется старая версия браузера, в которой не предусмотрена поддержка скриптов, текст сценария необходимо поместить между символами комментария HTML следующим образом:

```
<script language=[язык]>
```

```
<!--
```

```
    текст скрипта
```

```
//-->
```

Пример

```
<script language="javascript">
```

```
<!--
```

```
    текст скрипта
```

```
//-->
```



# Создание ссылки на файл сценария

- Тег `<script>` имеет атрибут **src=url**, позволяющий связать встраиваемый сценарий с внешним файлом. В качестве значения задается полный или относительный URL-адрес файла со сценарием. Ссылка задается следующим образом:

```
<script src=[имя_файла]>  
</script>
```

- Для файлов сценария JavaScript используется расширение **\*.js**, а для файлов VBScript - **\*.vbs**

Пример

```
<script type="text/javascript" src="admmenu.js">  
</script>
```

# Язык сценариев JavaScript

- Язык JavaScript был разработан фирмой Netscape для создания интерактивных HTML документов. Представляет собой объектно-ориентированный язык разработки сценариев. JavaScript является интерпретируемым языком. Синтаксис языка похож на синтаксис языка Java, поэтому его часто называют Java-подобным. Однако Java и JavaScript являются совершенно разными языками и ориентированы на выполнение различных задач в сети

# Две части JavaScript

- Сценарии JavaScript могут выполняться как на стороне клиента, так и на стороне сервера. Клиентские сценарии непосредственно встраиваются в HTML-документ и интерпретируются браузером по мере отображения его частей. Серверные сценарии выполняются на стороне сервера и предоставляют больше возможностей для создания динамических веб-страниц (например, использование баз данных, серверных объектов и т.д.)



# Версии JavaScript

- Язык JavaScript начал свое существование с момента выхода в мир браузера Netscape Navigator 2.0. Существует пять версий JavaScript:
  1. JavaScript 1.0 - первая версия языка, поддерживаемая Internet Explorer 3.0 и Netscape Navigator 2.0;
  2. JavaScript 1.1 - поддерживается Netscape Navigator 3.0 и Internet Explorer 4.0 (почти полностью);
  3. JavaScript 1.2 - поддерживается Netscape Navigator 4.0 и Internet Explorer 4.0 (частично);
  4. JavaScript 1.3 - поддерживается Netscape Navigator 4.5;
  5. JavaScript 1.5 - поддерживается Netscape 6.0. Большинство средств этой спецификации поддерживается Internet Explorer версии 5.5 и выше.
- Каждая последующая версия браузера поддерживает версии JavaScript, используемые в предыдущих версиях. Браузеры, которые поддерживают новые версии JavaScript, позволяют выполнять сценарии, созданные в ранних версиях JavaScript

# Ядро языка JavaScript

- И в клиентских и в серверных сценариях используется общий компонент языка, называемый *ядром*. Ядро включает в себя определение стандартных объектов и конструкций (переменные, функции, основные объекты)



# Синтаксис языка

- Сценарий JavaScript представляет собой набор операторов языка, последовательно обрабатываемых интерпретатором. Операторы можно располагать как в отдельной строке, так и в одной. В этом случае операторы отделяются друг от друга символом ; (если операторы располагаются в разных строках, использование этого символа не обязательно)
- Любой оператор можно расположить в разных строках без всякого символа продолжения
- Строковый литерал обязательно должен располагаться на одной строке
- Комментарии в JavaScript оформляются аналогично комментариям в C++ и C#. Символ // задает однострочный комментарий, а /\* \*/ - многострочный комментарий
- Язык JavaScript чувствителен к регистру символов

# [Тренировка]

- Создайте и просмотрите следующий html-документ:

```
<html>
<head>
<title>
Знакомство с JavaScript
</title>
<script type="text/javascript">
<!--
  alert("Я начинаю изучать JavaScript");
//-->
</script>
<noscript>
  Не могу выполнить сценарий
</noscript>
</head>
<body>
</body>
</html>
```