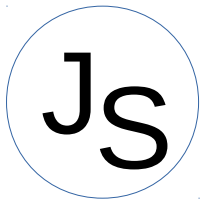


# JavaScript в Web-браузере



- Роль JS в Web, обзор основных возможностей JS в web-браузере
- Объект window как глобальный контекст
- Внедрение JS в HTML
- Выполнение программ JS
- Безопасность JavaScript

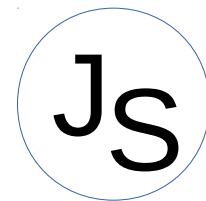


## Клиентский JavaScript

Окружение в браузере:

- Глобальный объект window.
- Иерархия объектов. DOM – её часть.
- Управляемая событиями модель программирования.

# Объект window



- Глобальный объект
- Свойства:
  - document
  - window, self
  - глобальные переменные

```
var answer = 42;  
window.answer = 42;
```

- Методы
  - open, close

# Внедрение JS в HTML



```
<html>
  <head>
    <title>Today's Date</title>
    <script>
      function print_todays_date( ) {
        var d = new Date( );
        document.write(d.toLocaleString( ));
      }
    </script>
  </head>
  <body>
    The date and time are:<br>
    <script>
      print_todays_date( );
    </script>
  </body>
</html>
```

# Внедрение JS в HTML

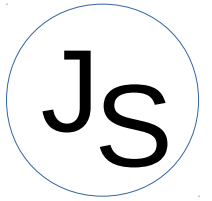


```
<script  
  language="JavaScript1.5"  
  type="text/javascript"  
>  
  . . .    // JS  
</script>
```

```
<script src="url"></script>
```

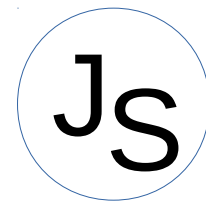
```
<script src="url" async></script>  
<script src="url" defer></script>
```

# Выполнение JS



- Скрипты в порядке следования
  - `document.write` – после `</script>`
  - Для внешних файлов – аналогично
  - Нужно учитывать последовательность разбора HTML
  - Синхронные скрипты должны быть быстрыми
  - При загрузке внешнего файла разбор останавливается
- 
- При перезагрузке страницы `window` возвращается в исходное состояние

# Безопасность JS



## Отсутствие возможностей

- Не поддерживает работу с локальными файлами
- Работа с сетью ограничена
  - Загрузка URL, отправка данных
  - Отправка почты
  - Нет сетевых примитивов
- Компоненты (плагины, ActiveX, Java) имеют больше возможностей
- Приватность (тип браузера, IP, история)

## Ограничение возможностей

`window.close()`

- `history.back()` `.forward()` `.go()`
- `FileUpload.value` – read only
- `submit()` – нельзя `mail:` и `news:`
- `window.open`, `window.resize` – минимальные размеры окна
- `about:` нельзя
- `Event` – все свойства read only



# Same-Origin Policy

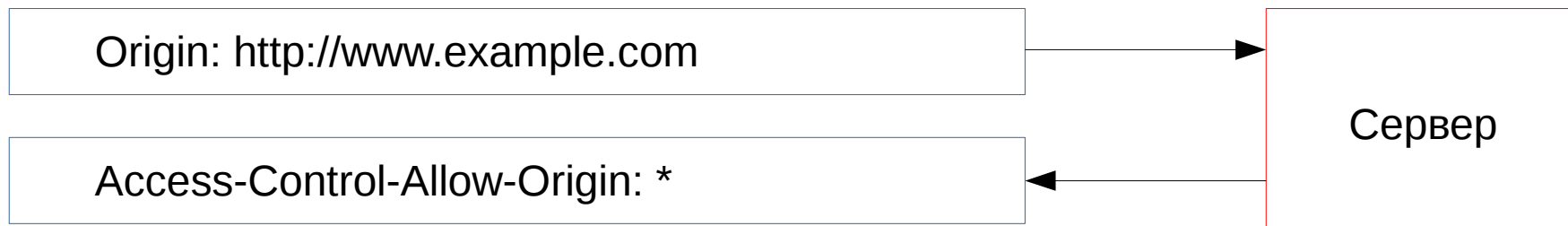


Происхождение = протокол + хост + порт

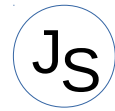
Скрипт имеет доступ к свойствам `document` и `window`, имеющим то же происхождение.  
Применяется к большинству свойств.

Ограничение иногда приходится ослаблять

CORS = Cross-Origin Resource Sharing



# JavaScript в Web-браузере



- Роль JS в Web, обзор основных возможностей JS в web-браузере
- Объект window как глобальный контекст
- Внедрение JS в HTML
- Выполнение программ JS
- Безопасность JavaScript

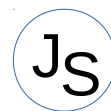


### Клиентский JavaScript

Окружение в браузере:

- Глобальный объект window.
- Иерархия объектов. DOM – её часть.
- Управляемая событиями модель программирования.

## Объект window

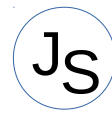


- Глобальный объект
- Свойства:
  - document
  - window, self
  - глобальные переменные

```
var answer = 42;  
window.answer = 42;
```

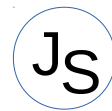
- Методы
  - open, close

# Внедрение JS в HTML



```
<html>
  <head>
    <title>Today's Date</title>
    <script>
      function print_todays_date( ) {
        var d = new Date( );
        document.write(d.toLocaleString( ));
      }
    </script>
  </head>
  <body>
    The date and time are:<br>
    <script>
      print_todays_date( );
    </script>
  </body>
</html>
```

# Внедрение JS в HTML



```
<script  
  language="JavaScript1.5"  
  type="text/javascript"  
>  
  . . .    // JS  
</script>
```

```
<script src="url"></script>
```

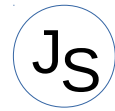
```
<script src="url" async></script>  
<script src="url" defer></script>
```

## Выполнение JS



- Скрипты в порядке следования
  - `document.write` – после `</script>`
  - Для внешних файлов – аналогично
  - Нужно учитывать последовательность разбора HTML
  - Синхронные скрипты должны быть быстрыми
  - При загрузке внешнего файла разбор останавливается
- 
- При перезагрузке страницы `window` возвращается в исходное состояние

## Безопасность JS



### Отсутствие возможностей

- Не поддерживает работу с локальными файлами
- Работа с сетью ограничена
  - Загрузка URL, отправка данных
  - Отправка почты
  - Нет сетевых примитивов
- Компоненты (плагины, ActiveX, Java) имеют больше возможностей
- Приватность (тип браузера, IP, история)



### Ограничение возможностей

`window.close()`

- `history.back()` `.forward()` `.go()`
- `FileUpload.value` – read only
- `submit()` – нельзя `mail:` и `news:`
- `window.open`, `window.resize` – минимальные размеры окна
- `about:` нельзя
- `Event` – все свойства read only

## Same-Origin Policy



Происхождение = протокол + хост + порт

Скрипт имеет доступ к свойствам document и window, имеющим то же происхождение.  
Применяется к большинству свойств.

Ограничение иногда приходится ослаблять

CORS = Cross-Origin Resource Sharing

