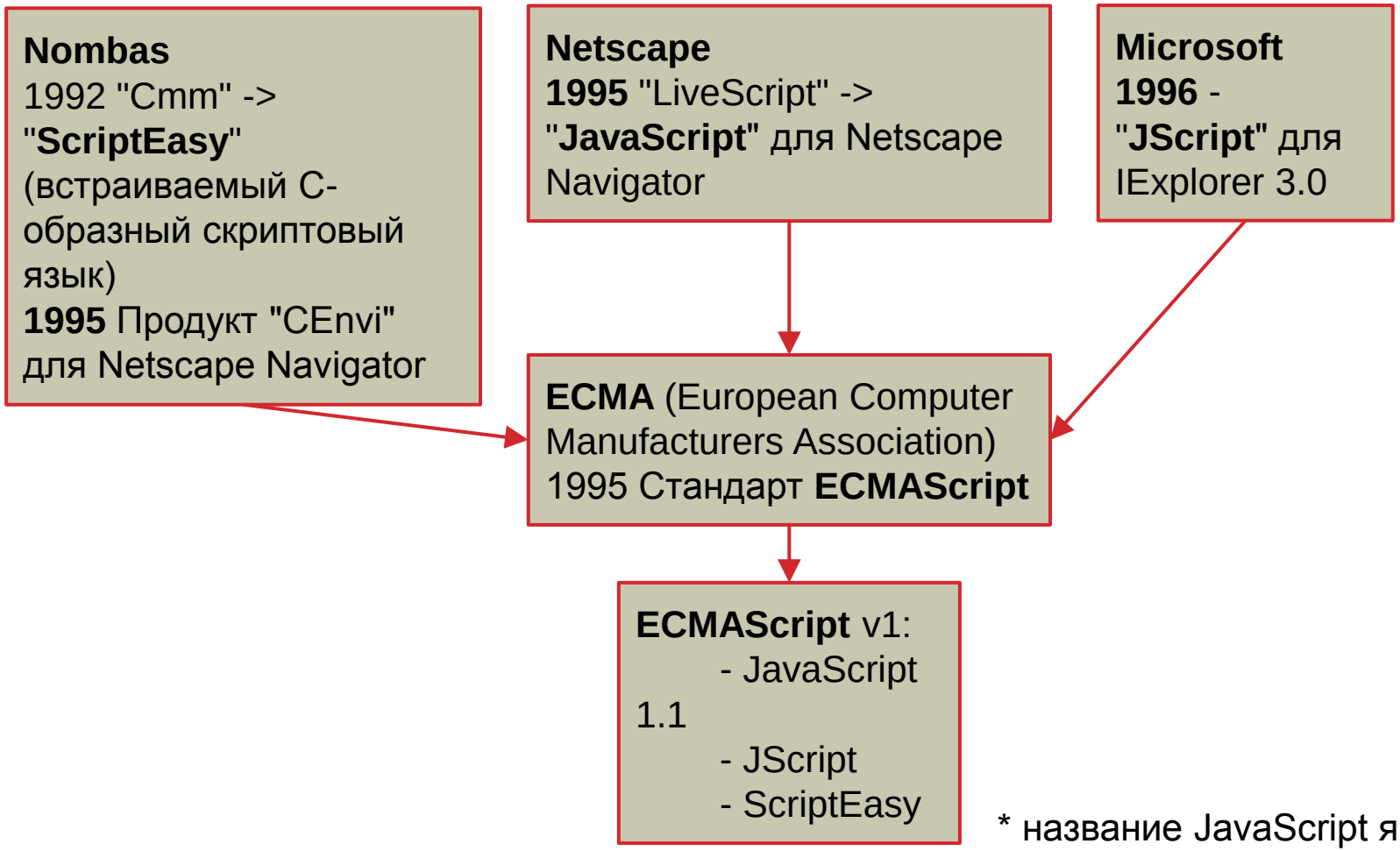
ВВЕДЕНИЕ В JAVASCRIPT

KEYWORDS: JAVASCRIPT, DOM, HTML

СОДЕРЖАНИЕ

- **История**
- **Подключение**
- **Язык**
  - Типы данных
  - Циклы и ветвление
  - Особенности языка
  - Исключения
- **Контекст исполнения**
  - Объекты window, document
  - Взамодействие с документом

ПРОИСХОЖДЕНИЕ



* название JavaScript является товарным знаком компании Sun / Oracle

ВКЛЮЧЕНИЕ СКРИПТОВ

```
<script src="ex1.js"></script>


<script
        type="text/javascript"
        src="ex1.js"
        charset="utf-8"></script>


<script type="text/javascript">
        document.write('some text');
</script>
```

- **Блокирует рендеринг**
- **Упорядочное исполнение**

ДРУГИЕ СПОСОБЫ ВКЛЮЧЕНИЯ

Асинхронная загрузка, исполнение после загрузки DOM

```
<script src="//other-domain.com/1.js" defer></script>
<script src="2.js" defer></script>
```

Асинхронная загрузка, исполнение по готовности

```
<script src="//other-domain.com/1.js" async></script>
<script src="2.js" async></script>
```

Асинхронная загрузка кодом

```
[ '//other-domain.com/1.js', '2.js' ]
        .forEach(function(src) {
                var script = document.createElement('script');
                script.src = src;
                document.head.appendChild(script);
        });
```

ТИПЫ ДАННЫХ

Простейшие

```
var a = "3.1";      // String
var b = 4;          // Number (int, float): NaN, Infinity
var c = false;
var d = null;
var e;              // undefined
document.write("a + b = " + (a + b));                          // 3.14
document.write("parseFloat(a) + b = ", (parseFloat(a) + b));   // 7.1
document.write("a - 0 + b = " + (a - 0 + b));                  // 7.1
document.write("parseInt(a) + b = " + (parseInt(a) + b));      // 7
```

Массивы

```
var array = [ 1 ]; // [], new Array()
array[1] = 2;
array.push(3);
array[3] = 4;
for (var i = 0; i < array.length; i++)
        document.write(array[i] + "; ");
for (key in array)
        document.write(array[key]+ "; ");
var c = array.map(function(x) { return x + 2; }) // [3,4,5,6]
var d = array.reduce(function(accum, next[, idx, arr]) {...})
```

ТИПЫ ДАННЫХ: ДИНАМИЧЕСКАЯ ТИПИЗАЦИЯ

```
var a = 1;
var b = "4";
var d = a - b;
console.log(d);
console.log(1.0 == '1');  // true
console.log(1.0 === '1'); // false
```

ТИПЫ ДАННЫХ

Объекты

```
var obj = { field1 : "one" }; // {}, new Object()
obj.field2 = "two";
obj["field3"] = "three";
for (field in obj)
    document.write( field + "=" + obj[field] + "; ");
```

Функции

```
var foo = function(a) { return a + 3; }
function bar(a) { return a + 3; }
```

Классы

```
var Class1 = function(para) { var q = para; this.p = 2; }
var obj1 = new Class1();
```

ЦИКЛЫ И ВЕТВЛЕНИЯ

Циклы

```
while (A) { … }

do { … } while (A)

for (var a = 0; a < x; a++) { … }

for (key in obj) { … }
```

Ветвления

```
if (A) { … } else { … }

switch (A) {
        case 'a1': …; break;
        default: …;
}
```

ОСОБЕННОСТИ ЯЗЫКА

- **Интерпретируемый**
- **Прототипно-ориентированный**
  - Особенности реализации ООП
- **Функции - объекты I класса**
- **Сборка мусора**

ПРОТОТИПНО-ОРИЕНТИРОВАННОСТЬ

```
function log(val) {
    console.log(val);
}

function bar() {
    //"class" definition
    var MyClass = function() {};
    // object creation
    var myObj = new MyClass();
    log(myObj.foo == undefined); // no method "foo"

    MyClass.prototype.foo = function() { log("some action"); };
    myObj.foo();    // it works!
}
```

ФУНКЦИЯ – ОБЪЕКТ 1 КЛАССА

```
function foo(a, b) {
    return function(x) { return a(x) + b; };
}

function square(v) { return v * v; };

var fn = foo(square, 1);
            /* function v*v+b */
console.log(fn(8));
            /* 8*8 + 1 = 65 */
```

СБОРКА МУСОРА

```
var data = [];
if (confirm("Trash?")) {
	for (var i = 0; i < 10000000; i++)
		data.push("ABCDEFG" + i);
}
if (confirm("Delete?")) {
	data = null;
	if (typeof GC != 'undefined')
		GC.Collect();
}
```

МЕХАНИЗМ ИСКЛЮЧЕНИЙ В JAVASCRIPT

Синтаксические ошибки приводят к ошибке времени компиляции и завершают разбор скрипта (переход далее)

```
var a = { f : 'sdf' ];
```

Семантические ошибки приводят к генерации исключений

```
try {
        var q = null;
        a.text; // err.name = ReferenceError
        q.text; // err.name = TypeError
        throw new BadValueError("text");
} catch (err) {
        alert("name: " + err.name +
                "\nmessage: " + err.message +
                "\nstack: " + err.stack);
} finally {
        boo();
}
```

КОНТЕКСТ

- **window**
  - Объект, описывающий вкладку
    - `.navigator, .location, .console, .document, .toolbar`
  - Объект по умолчанию
    - Все глабальные переменные – поля объекта window

- [window.]**document**
  - Объект DOM

УПРАВЛЕНИЕ ЭЛЕМЕНТАМИ CRUD

Create

```
var a = document.createElement('hr')
document.body.appendChild(a)
parentElement.appendChild(a)
```

Read

```
var someE = document.getElementById("ID")
```

(ещё getElementsBy[Name, ClassName, TagName])

Update

```
someE.innerHTML = 'qwerty';
someE.value = 'qwerty';
parentElement.replaceChild(newNode, oldNode);
```

Delete

```
document.body.removeChild(child);
```

ВЗАИМОДЕЙСТИЕ С ДОКУМЕНТОМ

Назначение обработчиков в HTML

```
<body onload="foo()"> …
<a onclick="bar(this.id)" id="id1">XXX</a>
```

Динамическое назначение обработчиков

```
document
      .getElementById('id1')
      .onclick = function(e) {}
```

АСИНХРОННОСТЬ ИСПОЛНЕНИЯ

JavaScript – однопоточный язык

Асинхронность достижима за счёт

- асинхронных операций (`XMLHttpRequest`)
- Функций
  - `setInterval(func, timeout),`
  - `setTimeout(func, timeout)`

ДОМАШНЕЕ ЗАДАНИЕ

Реализовать любую интерактивную игру

- Судоку
- Крестики-нолики
- Тетрис
- Space Invaders
- Arcanoid
- Volleyball
- …