Jakub Bartodziej

JavaScript i jQuery

http://students.mimuw.edu.pl/~jb291470/js/

Podstawowe cechy

- Język programowania stosowany do opisywania dynamicznych elementów stron WWW.
- Interpretowany
- Obiektowy (prototypy)
- Skryptowy
- Funkcje są „obywatelami pierwszej klasy”.
- Dynamiczny
- Słabo typowany

Hello World

1. Otwórz jakąś przeglądarkę
2. Wpisz w pasku adresu:

```
javascript:alert(”Hello, World!”);
```

Hello World w pliku

- Sposób 1:

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

- W hello.js:

```
alert(”Hello, World!”);
```

- Sposób 2:

```
<script type="text/javascript">
  alert(”Hello, World!”);
</script>
```

Zmienne

- Słabe typowanie
- Deklarujemy następująco:

```
var foo;
```

- Można zainicjalizować:

```
var foo = ”bar”;
```

- Przypisania:

```
foo = 8;
```

Prymitywy

- Boolean

```
var foo = true;
var bar = false;
```

- Numeric

```
var dwa = 2;
var trzyipol = 3.5;
```

- String

```
var s = ”String”;
```

Tablice

- Tworzenie:

```
var fib = new Array(0, 1, 1, 2, 3);
var fib = [0, 1, 1, 2, 3];
```

- Można mieszać typy:

```
fib[1]  = ”jeden”;
```

Operatory

- Takie same, jak w Javie.
- `===, !== - identyczność, bądź jej brak`
- `typeof x` – string opisujący typ `x`
- `o instanceof c` - sprawdza, czy obiekt o został stworzony przez konstruktor c
- `delete x – usuwa obiekt`
- `new cl –` nowy obiekt typu cl. cl musi być konstruktorem

Funkcje

- W pliku hello.js:

```
function helloWorld() {
  alert("Hello, World!");
}
```

- W pasku adresu:

```
javascript:helloWorld();
```

Ulubiony przykład

```
function fib(n) {
  if (n == 0) {
    return 0;
  } else if (n == 1) {
    return 1;
  } else {
    return fib(n-1) + fib(n-2);
  }
}
```

Lambda abstrakcja

```
var twice = function(n) {
   return 2 * n;
}
```

Obiekty

- Są tworzone za pomocą konstruktorów:

```
function Osoba(imie) {
   this.imie = imie;
}

var Jan = new Osoba(”Jan”);
```

- Ich pola są domyślnie inicjalizowane przez wartości określone w prototypach

```
Osoba.prototype.nazwisko = ”Kowalski”;

Jan = new Osoba(”Jan”);

//teraz Jan.nazwisko == ”Kowalski”
```

- Można to wykorzystać do dziedziczenia.
- Inna notacja pozwalająca tworzyć obiekty – literały obiektów:

```
Jan = { 'imię': 'Jan', 'nazwisko': 'Kowalski' }
```

Obsługa zdarzeń (Event handling)

- Specjalne atrybuty zaczynające się na on:

```
var helloWorld = function(evt) {
   alert("Hello, World!");
}

document.onclick = helloWorld;
```

- Bardzo użyteczny jest window.onload, który jest wywoływany po załadowaniu dokumentu.

DHTML

- Dynamic HTML
- Drzewo DOM (Dynamic Object Model) – umożliwia dowolną modyfikację całego dokumentu HTML.

Znajdowanie elementów

- Służą do tego metody obiektu `document`:

```
document.getElementById(”id”);
document.getElementsByTagname(”p”);
```

Dodawanie elementów

- Do tworzenia elementów służy metoda obiektu document:

```
var div = document.createElement(”div”);
div.id = ”someDiv”;
div.class = ”someClass”;
```

- Można teraz dodać nowy element do dokumentu:

```
document.documentElement.appendChild(div);
```

Modyfikacja elementów

- Można zmieniać atrybuty HTML za pomocą atrybutów javascriptowych obiektów:

```
document.getElementById(”someDiv”).class = ”otherClass”;
```

- Atrybut `style` modyfikuje styl css

```
document.getElementById('someDiv').style.color = 'red';
```

Usuwanie elementów

- Metoda `removeChild obiektu dokument:`

```
div = document.getElementById(”someDiv”);
document.removeChild(div);
```

jQuery

- Biblioteka JavaScript, która wszystko ułatwia.
- Przenośność między przeglądarkami.
- Łatwa nawigacja po drzewie DOM.
- Łatwa obsługa eventów.
- Łatwa dynamiczna zmiana stylów CSS.
- Łatwy AJAX
- Efekty i animacje.
- Narzędzia ułatwiające pracę.
- Dużo pluginów.

Pierwszy przykład

1. Dodaj między `<body></body>`:

```
<a href="http://jquery.com/">jQuery</a>
```

2. Dodaj między `<script></script>`:

```
$(document).ready(function() {
  $(”a”).click(function(event) {
    alert(”Hello, jQuery!”);
    event.preventDefault();
  });
});
```

jQuery i DOM

- Selektory umożliwiają łatwe wybieranie elementów z drzewa DOM:

```
$(".someClass").css("border", "red");
$("#someId").html(”Modified with jQuery”);
```

- Różne rodzaje selektorów:

`$(”#someId > p”)` - wszystkie elementy typu p, które są dziećmi elementu o Id someId

`$("p[@class]")` – wszystkie elementy typu p z atrybutem class

`$("p.foo[a]")` – wszystkie elementy typu p, klasy foo, które zawierają element typu a.

jQuery i eventy

```
$(”p.shy”).click(function() {
  $(this).hide(”slow”);
});
```

- Wszystkie paragrafy klasy `shy` schowają się po kliknięciu.

jQuery i AJAX

- `http://students.mimuw.edu.pl/~jb291470/js/post.html`