Przetwarzanie dokumentów XML i zaawansowane techniki WWW

"Obiektowy język JavaScript oraz format JSON"

(Zajęcia 07 - 18.04.2016 r.)

1) Wprowadzenie do formatu JSON

JSON czyli z ang. *JavaScript Object Notation* (Notacja Obiektowa JavaScriptu), to tekstowy format wymiany danych podobny funkcjonalnie do XML-a, oparty na literałach obiektowych JavaScriptu. Definicyjnie JSON należy on do samego JavaScriptu i w zasadzie jest jego podzbiorem, ale ze względu na tekstowy charakter jest uniwersalny, przenaszalny i niezależny oraz może zostac wykorzystany w każdym języku programowania. Format tekstowy (podobnie zresztą jak XML-a) zapewnia czytelność nie tylko dla parserów, ale również dla ludzi. Implemantacja i i użycie JSONa w większości języków jest ułatwiona przez zastosowanie gotowych bibliotek parujących.

JSON posiada sześć rodzajów wartości:

1. *obiekty,*
2. *tablice,*
3. *łańcuchy,*
4. *liczby,*
5. *wartości logiczne (TRUE FALSE),*
6. *wartość NULL.*

Białe znaki takie jak tabulacje, spacje, przejścia do nowej linii mogą być umieszczne w przed lub po każdej wartości w JSON dla zachowania przejrzystości i łatwiejszego czytania i analizy plików. Natomiast podobnie jak w XML-u znaki te mogą zostać pominięte dla zmniejszenia kosztów przesyłania danych.

Typowy obiekt w formacie JSON jest nieuporządkowanym zbiorem par "**klucz : wartość**", gdzie klucz jest dowolnym łańcuchem tekstowym, natomiast wartość musi być jednym z dozwolonych typów wymienionych powyżej. Obiekty JSON można zagnieżdzać w dowolnym stopniu jednak z reguły dla zachowania przejrzystości stosowana jest zasada "im bardziej płaska struktura tym lepiej". Jednak nie w każdym przypadku i dla wszystkich zastosowań jest to możliwe do utrzymania. Format zapisu obiektu w formacie JSON jest anstępujący:

```
{
      "klucz-JSON" : "wartość-JSON
}
```

Natomiast tablica w formacie JSON jest uporządkowanym ciągiem wartości które znów mogą byc dowolnego typu włącznie z obiektami i tablicami. Format zapisu tablic w JSON jest następujący:

```
[     wartosc1, wartosc2, …, wartoscN  ]
```

Łańcuchy w JSON są ogranicozne znakami podwójnego cudzysłowu " łańcuch", z zachownaiem znaku specjalnego ukośnika "\" dzięku któremu możemy zapisywań inne znaki specjalne. Natomiast liczby w JSON zapisywane są w formacie JavaScvriptu, z zastrzeżeniem że w liczbach całkowitych nie można na początku podawać zera ponieważ wtedy może ona zostać potraktowana jako zapis liczby w formacie ósemkowym.

Pierwszy przykład zapisu danych w formacie JSON:

```
{
```

```
            "first": "Jan",
            "last": "Kowalski".
            "born": 1980,
            "address": "ul. Reymonta 4"
      }
```

W formacie XML dane te wyglądały by następująco:

```
<?xml version='1.0' encoding='utf-8'?>
<person>
      <first>Jan</first>
      <last>Kowalski</last>
      <born>1980</born>
      <address>ul. Reymonta 4</address>
</person>
```

Do analizy struktury danych zapisanych w formacie JSON można użyć przeglądarek online np:
http://json.parser.online.fr/

Inny przykład teraz z tablicą obiektów:

```
{ "samochod": [
      {
            "Marka": "VW",
            "Model": "Golf",
            "Rocznik": 1999
      },
      {
            "Marka": "BMW",
            "Model": "S6",
            "Rocznik": 2007
      },
      {
            "Marka": "Audi",
            "Model": "A4",
            "Rocznik": 2009
      }
]
```

Te same dane w formacie XML:

```
<?xml version="1.0" encoding="utf-8"?>
<KOMIS>
  <Samochod>
      <Marka>VW</Marka>
      <Model>Golf</Model>
      <Rok>1999</Rok>
  </Samochod>
  <Samochod>
      <Marka>BMW</Marka>
      <Model>S6</Model>
      <Rok>2007</Rok>
  </Samochod>
  <Samochod>
```

```
      <Marka>Audi</Marka>
      <Model>A3</Model>
      <Rok>2009</Rok>
  </Samochod>
</KOMIS>
```

Kolejny przykład:

```
{
 "id": 10,
 "tytul": "Podstawy fizyki wysokich energii",
 "autors": [ "D. Perkins", "J.A. Zakrzewski" ],
 "roczniki": [ 1967, 1984, 1985, 1989, 2005 ],
 "wypozyczony": true
}
```

ZADANIE:
Proszę przygotować plik JS zawierający dowolne dane (np. te same które użyli Państwo do przygotowania pierwszego swojego XMLa). Dane docelowo można zapisać w pliku o rozszerzeniu .js

2) Podstawowe informacje o języku JavaScript

Aby użyć w dokumencie hipertekstowym skryptu napisanego w języku Javascript należy kod tego skryptu osadzić wewnątrz dokumentu HTML lub dołączyć go w postaci zewnętrznej biblioteki:

a.) Osadzanie skryptu wewnątrz dokumentu hipertekstowego:

```
<!DOCTYPE html>
<html>
<head>
      <title>Testowanie JS</title>
<script type="text/javascript">
<!--
// tu bedzie pierwszy scrypt napisany w JS
alert("Hello World");
document.write("<h3>To jest tekst wygenerowany w js przed zdarzeniem
onLoad();</h3><hr/>");
//-->
</script>
<noscript>
<center style="color:red;">!!!! Twoja przeglądarka nie obsługuje lub ma
wyłączony JS !!!!</center>
</noscript>
</head>
<body>
<h2>Test JavaScriptu</h2>
Akcja wywołana przed zdarzeniem OnLoad(); w momencie ładowania strony
<hr/>
</body>
</html>
```

b.) Dołączenie skryptu w postaci zewnętrznej biblioteki w pliku “external.js”:

```
<!DOCTYPE html>
<html>
<head>
      <title>Testowanie JS</title>
<script type="text/javascript" src="external.js"></script>
<noscript>
<center style="color:red;">!!!! Twoja przeglądarka nie obsługuje lub ma
wyłączony JS !!!!</center>
</noscript>
</head>
<body>
</body>
</html>
```

2) Zmienne i operacje artmetyczne:

W języku JS nie ma statycznej kontroli typu zmiennej dlatego zmienne deklarujemu za pomocą słowa kluczowego “var”, a typ określamy w momencie przypisania wartości zmiennej:

```
<script type="text/javascript">
<!--
var x;
x = 5;
alert(x*x);
x = "ddd";
alert(x);
x = null;
alert(x);
var imie = "Jan";
var nazwisko = "Kowalski";
alert(imie+" "+nazwisko);
document.write("<h1>"+imie+" "+nazwisko+"</h1>");
document.write("<hr/>");
document.write(typeof(imie));
document.write("<hr/>");
var a;
document.write(typeof(a));
//-->
</script>
```

- Dyrektywa “`document.write()`” pozwala na pisanie dowolnego tekstu do części body dokumentu hipertekstowego.
- metoda “`alert()`” pozwala na wypisanie informacji na ekran w postaci “wyskakującego okna komunikatu”.
- metoda pozwala “`typeof()`” na sprawdzenie typu zmiennej.

Parsowanie zmiennych pozwala np. na zmiane typu zmiennej z String na Integer, lub z String na Float, służą do tego dwie metody “`parseInt()`” oraz “`parseFloat()`”:

```
<script type="text/javascript">
<!--
var x = "1";
var y = "4";
x = parseInt(x);
y = parseInt(y);
document.write("ParseInt "+(x+y));
document.write("<hr/>");
x = "1.5";
y = "4.2";
x = parseFloat(x);
y = parseFloat(y);
document.write("ParseFloat "+(x+y));
document.write("<hr/>");
//-->
</script>
```

3) Tablice numerowane indeksem naturalnym:

```
var tab1 = new Array(); // tablica bez deklaracji
var tab2 = new Array(5); // 5 elementow tablica bez deklaracji wartosci
elementow
var tab3 = new Array(1,2,3,4,5);

// tablice nie maja ograniczenie na typ a wiec mozna:
var tab4 = new Array(2001, "Marcin", "Zielinski", 30, 5.0);
```

4) Pętle wyliczeniowe:

```
// petla for
for(n=0; n<5; n++){
 document.write(n);
 document.write("<hr/>");
}

//petla for-in - wybieranie z tablicy
var tablica;
tablica = new Array("a","b","c","d","e");
for (n in tablica){
 document.write(n);
 document.write(tablica[n]+"<br/>");
}
```

5) Tablice asocjacyjne:

```
// Tablice asocjacyjne przypisanie wartości do nieistniejącej właściwości
obiektu
// nie skończy się błędem ale zostanie utworzona nowa właściwość obiektu i
przypisana
// zostanie do niej wartość.
var tab5 = new Object();
```

```
tab5["Ala"] = "kot";
tab5["Pi"] = 3.1415;
document.write(tab5["Ala"] + "<br>" + tab5["Pi"]);
document.write("<hr/>");

// Tablice asocjacyjne - kolejny przyklad
var tablica = new Object();
tablica["Ala"] = "kot";
tablica["Pi"] = 3.1415;
for (var klucz in tablica)
    document.write(klucz + ": " + tablica[klucz] + "<br>");
```

6) Funkcje i metody:

```
// definiowanie funkcji
function Wypisz(imie){
 document.write("Witaj "+imie);
}
Wypisz("Jan");

// funkcja która coś zwraca
function Dodaj(a,b){
 return a+b;
}
document.write(Dodaj(5,7));

// konstruktor
function Wspolrzedne(x,y){
      this.x = x;
      this.y = y;
}
// tworzenie obiektu klasy -> operator new
var punkt = new Wspolrzedne(2,5);
//odwolanie do pol obiektu przez .
document.write("X: "+punkt.x+" Y: "+punkt.y);
document.write("<hr/>");

// inny przyklad
function NaEkran(){
   document.write("Wynik: "+this.a);
}
function Potega(a){
      this.a = a*a;
    this.wypisz = NaEkran;
}
var zmp = new Potega(5);
zmp.wypisz();
document.write("<hr/>");
```