ZAAWANSOWANE TECHNIKI WWW (WFAIS.IF-C112)

(zajęcia 03.12.2015 r.)

1. Biblioteka jQuery (dodatkowe informacje)

a. **)** Metody poruszania się po drzewie DOM i dodawanie nowych węzłów i usuwanie węzłów.
b.) Wykorzystanie metody "**ajax()**" do asynchronicznej komunikacji klient-serwer.

Informacje do punktu a.) i b) opisane są szczegółowo w wykładzie 8.

Zadanie 1:

Proszę przygotować dokument HTML w którym będzie lista nienumerowana <ul> z kilkoma pozycjami. Następnie prosze przygotować skrypt który będzie reagował na zdarzenie click() na elemencie listy i w wyniku wywołania zwrotnego usuwał daną pozycje listy.

Zadania 2:

Proszę przygotować dokument HTML z fragmentem dowolnego tekstu. W tekście powinny znajdować się słowa z dodatkowymi wyjaśnieniemi (przypisami) które pojawiją się po najechaniu na dane słowo kursorem mysze w formie tzw. okna pop-up. Funkcję tę prosze oprogramować za pomocą funkcjonalności dostępnych w bibliotece jQuery. Materiały pomocne do realizacji tego zadania dostępne są w wykładzie 8.

2. Wprowadzenie do środkowiska Node.js

NodeJS jest zaawansowanym środowiskiem programistycznym do tworzenia skalowalnych i bardzo wydajnych aplikacji webowych. Środowisko NodeJS (w skrócie po prostu node) napisane jest w pełni w języku JavaScript i dostępne na wszystkie platformy systemowe: Windows, Linux, MacOS X. Aplikacje oparte o środowisko node działają w asynchronicznym systemie wejścia/wyjścia sterowane za pomoca zdarzeń i pozwala na uruchomienie kodu JavaScriptowego poza przeglądarką internetową.

System został stworzony przez Ryana Dahla w 2009 roku i od tego czasu stał sie jedną z najlepszych platform tworzenia aplikacji webowych stosowana przez wiele światowych korporacji m.in. LinkedIn, Microsoft, Yahoo, Walmart, PayPal, a w Polsce np Onet.pl i serwisy spółki Agora.

Platforma ta wykorzystuje silnik Google V8 jako interpretatora języka skryptowego JavaScript. Zastosowanie bibliotek obsługujących zapytania http oraz porty, umożliwia platformie node zachowanie się jak serwer www podobnie jak np. Apache. System posiada bardzo dobrze rozbudowany system wtyczek dostępnych w repozytorium on-line (npn) które można instalować w zależności od potrzeb.

Instalacja:

0) Wejść na stronę: http://nodejs.org

1) Pobrać najnowsza wersję platformy node.js, a następnie rozpakować do katalogu domowego (> tar -zxvf nazwa_archiwum.tar.gz) i zmienić nazwę powstałej kartoteki na "nodejs".

2) W utworzonej kartotece znajduje się kartoteka “bin” w której zlokalizowany jest główny plik wykonywalny o nazwie “node”, stanowiący interpreter kodu JS.

3) Domyślnie używamy powłoki terminala “bash”. Jeśli nie jest to domyślna powłoka należy się do niej przełączyć wpisując w terminalu polecenie:

```
> bash
```

4) Definiujemy tworzymy następujące linki symboliczne w głownej kartotece nodejs:

```
> ln -s ./bin/node node
> ln -s ./bin/npm npm
```

5) W pliku **.bashrc** (jeśli nie istnieje tworzymy taki w sowjej głównej kartotece domowej) dodajemy następującą zmienną środowiskową:

```
export PATH=$PATH:/hom/username/nodejs/bin
```

gdzie username - ocznacza nazwę użytkownika
nodejs - oznacza nazwę kartoteki instalcji nodjs

6) Następnie w głównej kartotece domowej wykonujemy polecenie:

```
> source .bashrc
```

7) Jeśli w kartotece mamy np. plik.js, zawierający skrypt w języku JS, to aby go wykonać należy uruchomić go następującym poleceniem:

```
> node plik.js
```

8) W celu sprawdzenia czy poprawnie “zainstalowaliśmy” środowisko node można wykonać polecenie:

```
> node -v
```

Pierwszy skrypt JS dla środowiska node:

1) Tworzymy plik “witaj.js”
2) W pliku umieszczamy jednolinijkowe polecnie skryptowe:

```
console.log("witaj swiecie!");
```

Metoda console.log()pozwala na wyświetlanie komunikatów do terminala jak np. w C++ cout.

3) Uruchamiamy wykonując polecenie:

```
> node witaj.js
```

4) Wynik działania skryptu powinien być widoczny jako komunikat w terminalu linux.

Inny przykład:

```
// tak dodajemy komentarz
var a = 7;
var b = 8;
console.log( a + b );
console.log( a - b );
console.log( a * b );
console.log( a / b );
```

W języku JavaScript zmienne definiujemy za pomocą słowa kluczowego “var”. Należy podkreślić, że JS

jest językiem zmiannych bez deklaracji typu, co powoduje że typ zmiennej jest definiowany w momencie przypisania jej wartości.

Deklaracja przez obiekty:

```
var obiekt1 = {
           a: 7
           }
var obiekt2 = {
           b: 8
           }
console.log( obiekt1.a + obiekt2.b );
var obiekt3 = {
           a: 7,
           b: 8
           }
console.log( obiekt3.a + obiekt3.b );
```

Tworzenie serwera HTTP w środowiska node:

Do stworzenia serwera http w node należy użyć wbudowanego modułu “http”. Moduły dołączamy dyrektywą require w następujacy sposób:

```
var nazwa_obiektu = require(‘nazwa_modulu’);
```

zwykle dla zachowania przerzystości “nazwa_obiektu” jest tożsama z nazwą dołączanego modułu. W przypadku modułu http dołączenie ma postać:

```
var http = require('http');
```

Następnie używamy kodu do uruchomienia serwera http znajdującego się na stronie głownej projektu node.js:

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(1337, '127.0.0.1');
console.log('Server running at http://127.0.0.1:1337/');
```

Ta prosta funkcja uruchamia serwer http słuchający na porcie 1337 lokalnego hosta. Wynik działania można oberzeć w przeglądarce wpisując adres: http://127.0.0.1:1337.

W przypadku naszego kodu z obiektu http, uruchamiamy metodę “createServer()”. W metodzie tej jako argument podajemy funkcję która jest pewnego rodzaju funkcją odpowiedzi (zwrotną) tzw: “callback”. Zostanie ona wykonana za każdym razem kiedy przyjdzie zapytanie do serwera pod zdefiniowany adres. W funkcji tej mamy dwie zmienne:

req - odpowiedzialną za przechowanie obiektu żądania http.

res - odpowiedzialną za przechowanie obiektu odpowiedzi http.

W naszym kodzie po zgłoszeniu żadania do serwera tworzona jest dopowiedź, najpierw przez utworzenie nagłówka a następnie wygenerowanie ostatecznej odpowiedzi metodą “end”.

Ostatnia linijka zawiera metodę “listen()”, która informuje serwer na jakim porcie ma słuchać i pod jakim adresem.

Zadanie:
Proszę zmodyfikować program tak aby w ostatecznej odpowiedzi renderowana była prosta strona internetowa.