

Sprawozdanie z ćwiczenia nr 1 z laboratorium Programowanie internetowe

Tomasz Cudziło
PW EE INFORMATYKA

23 października 2011

1 Opis projektu

Celem ćwiczenia było zapoznanie się z językami HTML 4.01 oraz XHTML 1.0, wykorzystanie elementów przez nie oferowanych oraz dostrzeżenie i zaprezentowanie różnic między nimi.

1.1 Wykonanie

Adres projektu:

<http://volt.iem.pw.edu.pl/~cudzilot/pi/cw1/index.html>

Repozytorium projektu

<https://github.com/student-tomasz/pi-cwiczenie-1>

Repozytorium sprawozdania:

<https://github.com/student-tomasz/pi-sprawozdanie-1>

1.2 Narzędzia

Kod został napisany w Vim, bez wykorzystania generatorów¹. Zarządzanie kodem odbywa się za pomocą git i serwisu [GitHub](#). Również aktualizacja źródeł na serwerze wykorzystuje git.

1.3 Biblioteki

Strony posiadają JavaScript korzystający z biblioteki [jQuery](#) w wersji 1.6.4, oraz [google-code-prettify](#) w wersji release z dnia 2011-06-01.

¹Oprócz tagu <embed> w liście elementów HTML, który został skopiowany ze strony filmu.

Wykorzystywane też są style CSS normalizujące wyświetlanie elementów. Pochodzą one z biblioteki [YUI 3](#). Są to pliki `reset.css`, `base.css` i `fonts.css` w katalogu `../css/` w stosunku do katalogu zawierającego projekt z ćwiczenia.

2 Wnioski

2.1 Elementy HTML

Język HTML pozwala na tworzenie rozbudowanych dokumentów, które przy stosowaniu się do standardów W3C oferują identyczne doświadczenie użytkownikowi niezależnie od jego urządzenia i systemu operacyjnego.

Ponieważ dostęp do Internetu jest powszechny i nowoczesne przeglądarki wspierają standardy W3C, jest to wygodna platforma do tworzenia aplikacji.

2.2 Porównanie języków

Język XHTML powstał by znormalizować HTML na tyle by parsery XML mogły swobodnie przetwarzać dokumenty. Nie rozszerza jego funkcjonalności w znaczy sposób, wymusza precyzyjną składnię i rozdziela warstwę prezentacji od danych — tj. usuwa atrybuty i tagi z HTML, które odpowiadają za wygląd a nie zawierają treści.

2.3 Budowa dokumentów oparta o elementy `<div>` i `<table>`

Tworzenie strony, której layout jest stworzony na bazie `<table>` wymaga więcej czasu i kodu niż stworzenie identycznego wyglądu na bazie `<div>`. Strony porównań są identyczne względem ich wyglądu. Strony z tabelami zawierają około 40% linii niż ich odpowiedniki z `<div>`.

Ustawianie elementów strony za pomocą tabel jest zupełnie nie semantyczne, bezpośrednio wymusza ich pozycję w dokumencie HTML. Funkcjonalność ta jest spełniana stylami CSS. W efekcie nadmiarowa ilość znaczników zamykających komórki i wiersze tabel dodatkowo zmniejsza czytelność kodu i powoduje raka płuc. *Tabele służą tylko do prezentowania zestawu danych.*

Technika tworzenia stron w oparciu o tabele była popularna kiedy obowiązujące standardy HTML i CSS nie dawały innych możliwości. To było w 1996 roku.

Layout na `<div>` jest łatwiejszy w tworzeniu i utrzymaniu. Zmiany przeważnie wprowadza się tylko w stylach, kod dokumentu jest przejrzysty i semantycznie poprawny.

3 Uwagi

Wszystkie strony wymagają włączonej obsługi JavaScript. Wszystkie są zgodne ze standardami HTML 4.01 Strict lub XHTML 1.0 Strict, ich style z CSS 2.1.

Strony zostały przetestowane w Firefox 7.0.1, Safari 5.1.1 i Chrome 15.0.

3.1 Elementy HTML

Strona zawierająca demonstracyjne wykorzystanie elementów HTML jest niezgodna ze standardem ze swojej deklaracji DOCTYPE. Zawiera elementy przedstawione na wykładzie, nie wchodzące w skład specyfikacji XHTML 1.0 Strict.

3.2 Porównanie języków

Strony wyglądają identycznie. W Operze 11.52 margines dolny tabeli jest traktowany niż w silnikach Gecko lub Webkit i jest doliczany do odstępu marginesu górnego elementu `<h2>` walidacji. Stąd drobna różnica w odstępie.

3.3 Wyświetlanie źródła strony

Wyświetlanie źródła strony jest oparte na JavaScript. Kod wstawiający źródło na stronę jest umieszczony w pliku `../js/insert-source.js` w stosunku do indexu projektu.

Gdy strona zostanie wczytana i DOM załadowany, wysyłane są żądania po pliki źródłowe wykorzystywane na aktualnie przeglądanej stronie. Najpierw zbierane są ścieżki do plików podlinkowanych na stronie, oraz ścieżka samej strony. Następnie wysyłane jest żądanie po każdy z tych plików, którego odpowiedź interpretowana jest jako zwykły tekst.

Żądania są wysyłane synchronicznie, może się zdarzyć, że przedłuży to znacznie czas ładowania strony. Nowoczesne przeglądarki zachowują załadowane zasoby w pamięci podręcznej, z której powinny być wczytywane odpowiedzi na następne zapytania.