

Skok w przyszłość - XHTML

Na początek warto byłoby powiedzieć czym jest XHTML. Otóż wg definicji, XHTML jest aplikacją XML... no i teraz mnie wyzywasz:-) XML jest językiem znaczników, który sam siebie opisuje. W skrócie oznacza to, że tworzysz własne znaczniki a ich wygląd opisujesz w CSS - tak jak tworzyliśmy "nowe" tagi z DIVów. XHTML jest zgodny ze specyfikacją XML i "symuluje" HTML. Tzn., że ma zestaw znaczników prawie identyczny z HTMLem [o czym mogłeś się przekonać], jednak w przyszłości możliwe będzie przekonwertowanie poprawnie napisanych dokumentów XHTML na XML - wydłużając ich żywotność.

Dlaczego należy przestrzegać standardów?

A dlaczego należy przestrzegać przepisów ruchu drogowego?:-) Jeszcze kilka lat temu wcale mnie nie dziwiło, że strona nie jest zgodna ze standardami - "pff, przecież nikt ich nie przestrzega!". Jednak sytuacja się zmieniła diametralnie: nowe przeglądarki, nowe technologie i rzesze nowych twórców, którzy wymyślają to coraz zabawniejsze sposoby na łamanie standardów... Póki na rynku była jedna przeglądarka - rzeczywiście przestrzeganie standardów nie było aż takie konieczne [choć i tak należało ich przestrzegać]. Jednak nie sprawdzało się wtedy poprawności ze standardami, tylko wygląd w tej przeglądarce. Teraz są już co najmniej 3 ważne przeglądarki: MSIE6.0, FF1.0, Opera8.0. A do tego doliczyć można jeszcze Mozillę i Netscape'a. Starczy więc nauczyć się standardów, które wszystkie te przeglądarki przestrzegają [może oprócz MSIE], a nie zestawu nowych tagów, hacków i oryginalnych atrybutów obsługiwanych tylko przez "wybranców". Po za tym - jak napisałem wcześniej - poprawny dokument XHTML można przekonwertować na XML, dzięki czemu wydłuża się życie takiej strony. Często można się spotkać z zarzutem, że standardy nie pozwalają na tworzenia efektownych stron... o większej bzdury nie słyszałem - pozwalają naprawdę rozwinąć skrzydła twórcom i sprawić by ich praca była jeszcze przyjemniejsza - gdyż nie trzeba co chwilę sprawdzać w każdej przeglądarce jak wygląda strona...

Z pewnością, jeśli chcesz się na poważnie zająć tworzeniem stron [nie mówię "zawodowo"], to przestrzegaj standardów, bo strona stworzona zgodnie ze standardami to strona lepsza, mniejsza objętościowo i nowocześniejsza.

Różnice między HTML a XHTML

Na dobrą sprawę prócz kilku prostych rzeczy XHTML jest identyczny z HTML. Zaczniemy od nagłówka:

```
W XHTML: <!DOCTYPE html PUBLIC "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd" > <html
xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl" >
W HTML: <!DOCTYPE HTML PUBLIC "http://www.w3c.org/TR/html4/loose.dtd" > lub:
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3c.org/TR/html4/strict.dtd"> lub: <!DOCTYPE HTML PUBLIC  
"-//W3C//DTD HTML 4.01 Frameset//EN"  
"http://www.w3c.org/TR/html4/frameset.dtd">
```

Każda strona, czy to napisana w HTML, czy to w XHTML powinna zawierać deklarację typu dokumentu. Jak widzimy w XHTML pojawiły się także dodatkowe atrybuty w znaczniku <html>. HTML ma kilka różnych deklaracji, ponieważ jest kilka "odmian" tego samego języka. Najbardziej restrykcyjnym jest 'strict', 'transitional' jest przejściowy, a 'frameset' pozwala na stosowanie ramek.

Lista nowych zasad:

1. Tagi i atrybuty pisane małą literą.

W HTML często stosowało się zapis '<ZNACZNIK atrybut="wartość">'. Było to nawet wygodne i często zalecane, gdyż łatwo było spostrzec co jest znacznikiem, a co jego atrybutem. Czasem stosowało się zapis '<ZNACZNIK OnMouseOver="">', który także powinien zostać zmieniony - wszystko co jest umieszczone między < a > musi być z małej litery. **Wyjątek:** deklaracja 'DOCTYPE' zawsze jest pisana wielką literą - wyjątek potwierdza regułę.

2. Zamykanie atrybutów i tagów

Każdy atrybut powinien mieć wartość, czyli:

w HTML było dopuszczalne: <INPUT type="checkbox" name="ok" value="tak" selected> w XHTML już należy zrobić to tak: <input type="checkbox" name="ok" value="tak" selected="selected" />

Jak łatwo zauważyć kod XHTML jest w całości pisany małą literą, atrybut 'selected' zyskał wartość, a dodatkowo na końcu pojawiło się '/>'. Pisząc strony w XHTMLu należy pamiętać, że każda wartość atrybutu **musi** zostać ujęta w cudzysłów. Także każdy element musi zostać zamknięty:

w HTML dopuszczalne było:
 lub to jest opcja w XHTML należy to już zrobić tak:
 to jest opcja

Jeśli znacznik jest pusty [np. br] należy nadać mu nową strukturę: '<pusty_element />'.

3. Zamykanie w odwrotnej kolejności do otwierania

w HTML można było: `<i><u>To jest tekst</u></i>` w XHTML należy już zachować kolejność: `<i><u>To jest tekst</u></i>`

Tu chyba wszystko jasne - najpierw wsiadasz do samochodu, potem go odpalasz. Jak kończysz jazdę, najpierw zatrzymujesz się gasisz silnik, a potem wysiadasz - nie odwrotnie:-)

Inny sposób dołączania JS:

w HTML: `<script language="JavaScript"> // kod skryptu </script>` w XHTML: `<script type="text/javascript"> <![CDATA[// kod skryptu]]> </script>`

Jak widać w przykładzie zmieniono atrybut `language` na `type`. Zamianie uległa także wartość. Następną nowinką jest ujęcie kodu JS w klamrę tworzoną przez `'<![CDATA[...]>'`.

"Słownik" XHTMLowo HTMLowy:-)

- HTML: `'<basefont face="arial" color="red">'`
- zastosowanie: ustawienie domyślnej czcionki i koloru dla całej strony
- XHTML [CSS]: `body { font-family: arial; color: red; }`
- HTML: `'<center></center>'`
- zastosowanie: Wyśrodkowanie elementów
- XHTML [CSS] - wyśrodkowanie zawartości pojemnika: `p { text-align: center; }`
- XHTML [CSS] - wyśrodkowanie pojemnika: `p { margin-left: auto; margin-right: auto; }`
- HTML: `''`
- zastosowanie: ustawienie czcionki i koloru dla fragmentu tekstu
- XHTML [CSS]: `span { font-family: arial; color: red; }`
- XHTML: Wycofano ze specyfikacji 'font'. Należy używać 'span'

Skok w przyszłość - XHTML

Wpisany przez Patryk yarpo Jar
piątek, 18 września 2009 02:52

Prócz wyżej wymienionych elementów wycofano sporą część atrybutów. Wszystkie usunięte atrybuty należy zastąpić [najczęściej identycznie nazwanymi] atrybutami CSS:

- HTML: 'alink, link, vlink'
- zastosowanie: zmiana wyglądu hiperłącza
- miejsce opisu: definicja body ['<body ...>']

- XHTML [CSS]: pseudoklasy - a:active, a:link, a:visited

- HTML: 'background, bgcolor'
- zastosowanie: określanie tła [wczytywanie grafiki oraz zmiana koloru tła]
- miejsce opisu: definicja body oraz definicje innych elementów

- XHTML [CSS]: background-color: kolor; background-image: url(plik.gif);

- HTML: 'height'
- zastosowanie: określanie wysokości tabel
- miejsce opisu: definicja tabeli

- XHTML [CSS]: height: 200px;

- HTML: 'hspace, vspace'
- zastosowanie: określanie odległości poziomych [hspace] i pionowych [vspace]

- XHTML [CSS]: padding:5px; margin:5px;

Przedstawiona tu lista nie jest pełna. Wypisałem tu najczęściej używane właściwości i elementy oraz sposoby ich zastępowania.