

KURS XHTML I CSS – PODSTAWY

Copyright © 2011 INSIGO Grzegorz Róg. Wszelkie prawa zastrzeżone.

INSIGO Grzegorz Róg
www.eduweb.pl

ISBN: 978-83-62083-16-9

Kurs składa się z dwóch elementów: książki oraz płyty. Dołączona płyta stanowi część składową książki i nie może być przedmiotem odrębnej transakcji.

AUTOR

Grzegorz Róg



Adobe

CERTIFIED EXPERT

Grzegorz Róg od wielu lat projektuje serwisy internetowe, dużą wagę przykładając do ich funkcjonalności a także skutecznych technik promocji. Swoje umiejętności praktyczne zdobył projektując grafikę oraz strony internetowe dla wiodących polskich i zagranicznych agencji reklamowych. Prowadzi serwis eduweb.pl, jest trenerem oraz certyfikowanym ekspertem Adobe.

Informacje Ogólne

HTML (the Hypertext Markup Language) oraz CSS (Cascading Style Sheets) są podstawowymi technologiami budowy stron WWW. Dzięki językowi HTML planujemy tzw. strukturę strony, osadzając w niej zawartość i definiując, w jakich kontenerach ma znaleźć się zawartość. Z kolei z pomocą CSS nadajemy wygląd witrynie, odwołując się do odpowiednich znaczników z kodu HTML. Dzięki temu możemy zadbać o układ elementów na stronie oraz ich wygląd - przykładowo kolor i wielkość tekstu.

Aby stworzyć pełną definicję współczesnej strony WWW, do kodu HTML i CSS należy jeszcze dodać skrypty - zarówno te, działające po stronie serwera (PHP czy ASP), jak i po stronie przeglądarki (JavaScript, ActionScript w plikach Flash), a także multimedia - zdjęcia, animacje, video i dźwięki.

Czym jest HTML

HTML trudno zakwalifikować do języków programowania. Jest to raczej język do oznaczania składni (markup), który pozwala klasyfikować zawartość dokumentu z rozszerzeniem .html lub .htm zgodnie z zasadami strukturalnymi. Pojedynczy element HTML określamy jako Tag lub znacznik. Dowolny znacznik może zawierać skojarzone z nim atrybuty, np.

```
<nazwaznacznika atrybut="treść atrybutu"> Zawartość znacznika </nazwaznacznika>
```

W składni występuje kilka charakterystycznych elementów, są to:

- Znacznik otwierający - konstrukcja znaczników opiera się na ostrych nawiasach
- Atrybut, którego wartość ujęta jest w cudzysłowie po znaku równości
- Treść, która składa się na zawartość znacznika
- Znacznik domykający - kończy daną konstrukcję

Przykładowo:

```
<p class="zielonyAkapit"> Treść akapitu </p>
```

Znacznik określa akapit tekstu i dodatkowo posiada atrybut klasy `class`.

W kodzie HTML występują też znaczniki, które domykane są w sposób uproszczony i nie zawierają treści, np. znacznik `` lub znacznik `
`. Aby jednocześnie otworzyć i domknąć znacznik, korzystamy z zapisu `
`, czyli dodajemy ukośnik przed zakończeniem znacznika.

Czym jest XHTML

XHTML to nieco zmodyfikowana składniowo i unowocześniona wersja HTML. Jest to pewna ewolucja języka i bazuje na znacznikach, do których formułowania zostały wprowadzone pewne obostrzenia. XHTML bazuje na XML, jednak w przeciwieństwie do swojego przodka oferuje zamknięty zestaw znaczników na których możemy pracować.

W XHTML definiujemy podstawową strukturę strony w oparciu o podział jej treści. Niektóre z nich to:

- Elementy podstawowe - wersja XHTML `<doctype>`, tytuł strony `<title>`, znaczniki `<meta>` i `<head>`, które zawierają np. skrypty JavaScript i style CSS lub odwołania do nich.
- Podstawową strukturę dokumentu, czyli jego ciało `<body>`
- Nagłówki, czyli znaczniki od `<h1>` do `<h6>`
- Akapity tekstu, które oznaczamy jako `<p>`, kontenery `<div>` czy ``, oraz nowe elementy składniowe jak `<header>`, `<footer>`, `<article>` czy `<aside>`
- Tabele `<table>`
- Listy nieuporządkowane `` i uporządkowane ``
- Grafiki, czyli np. pliki `.jpg`, `.gif`, `.png` w znacznikach ``
- Hiperłącza, czyli odnośniki do innych dokumentów html - znacznik `<a href>`
- Elementy multimedialne jak `<video>`
- Formularze, oraz przyciski służące do ich obsługi w znaczniku `<form>`

Czym jest CSS

CSS jest językiem, z pomocą którego modyfikujemy warstwę prezentacji strony WWW i określamy jej wygląd. Z pomocą stylów możemy wpływać na wygląd struktury przygotowanej w HTML poprzez odwołanie do tagów. Ze znacznikami HTML możemy komunikować się poprzez tzw. selektory i dzięki temu określać wygląd:

- Samych tagów, np. wszystkich akapitów `<p>` formułując selektor `p`
- Atrybutów klas CSS przypisanych do znaczników, przykładowo do `<p class="akapit">` odwołamy się z pomocą selektora `.akapit`, gdzie kropka na początku wskazuje, że odwołujemy się do klasy. Co do zasady, w ramach jednego dokumentu .html możemy nadawać ten sam atrybut klasy wielu elementom, a następnie wpływać na wygląd ich wszystkich z pomocą jednej reguły CSS
- Atrybutów identyfikatorów przypisanych do znaczników, przykładowo do `<p id="akapit2">`, odwołamy się z pomocą selektora `#akapit2`, gdzie znak `#` reprezentuje identyfikator. Przyjęło się, że ten sam identyfikator występuje tylko raz w kodzie dokumentu .html i jest to unikalny, większy fragment naszej strony.

Wprowadzenie języka HTML5 uprości nieco proces tworzenia takich odwołań, ponieważ będziemy mieli do dyspozycji wiele nowych znaczników określających, jaka treść się w nich znajduje. Dzięki temu będziemy mogli zminimalizować ilość odwołań do klas i identyfikatorów, a zwiększyć ilość odwołań do tagów.

Przykładowo, zamiast tworzyć wiele elementów blokowych `div`, np:

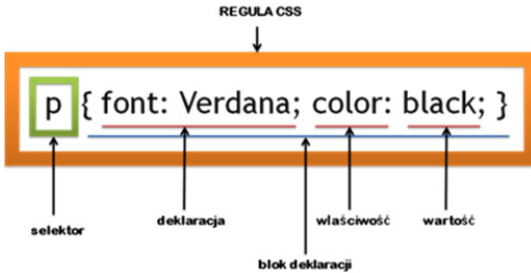
```
<div id="footer">
```

i odwoływać się do identyfikatora w CSS, będziemy mogli odwołać się do nowego znacznika:

```
<footer>
```

Tworząc w XHTML możemy już teraz pomyśleć o tym, jak przygotować się do zmiany struktury na nowe standardy, w momencie gdy przeglądarki będą do nich przystosowane. Możemy np. używać identyfikatorów identycznych, jak nazwy elementów składniowych HTML5 a następnie tylko nieznacznie zmodyfikować strukturę strony, pozostawiając jako nazwę tagu zawartość atrybutu.

Składnia reguły CSS



Aby stworzyć regułę CSS która zmodyfikuje właściwości wszystkich akapitów na naszej stronie, możemy posłużyć się np. taką składnią:

```
p {
  font-family: Verdana;
  color: #000; }
```

W ten sposób zdefiniujemy wygląd wszystkich akapitów w dokumentach, które korzystają z tego stylu CSS tak, że będą one zapisane krojem Verdana w czarnym kolorze.

Niezwykle ważną techniką jest tworzenie zewnętrznych arkuszy stylów, czyli plików z rozszerzeniem `.css` i modyfikowanie w nich wyglądu wielu dokumentów `.html`. W ten sposób minimalizujemy ilość kodu, która potrzebna jest nam do wprowadzenia zmian i zwiększamy szybkość ładowania podstron, ponieważ pliki `.css` po pierwszym załadowaniu znajdują się w pamięci podręcznej przeglądarki.

Aby podlinkować plik `css` do wielu dokumentów `.html`, w ich sekcji `<head>` dodajemy następujący znacznik:

```
<link rel="stylesheet" href="ścieżka-do-pliku.css" type="text/css" media="screen" />
```

Nadchodzi epoka HTML5

HTML5 to kolejny etap ewolucji języka. Tym razem udoskonalony został XHTML, do którego doszło wiele nowych znaczników pomocnych przy definiowaniu treści na stronie. HTML5 nadąża za aktualnymi trendami w projektowaniu i wspomaga tworzenie większych struktur aplikacji w przeglądarce. Dzięki różnicowaniu treści na etapie nazw znaczników HTML5 przeglądarki różnych urządzeń, w tym telefonów czy tabletów są w stanie lepiej skategoryzować i wyświetlać treści.

Warto zaznaczyć, że wiele funkcji HTML5 okrzykniętych jako rewolucyjne, tak na prawdę jest zastugą CSS3, który niesie za sobą jeszcze większy, w naszej ocenie, potencjał.

Powstaje zatem istotne pytanie - czy nie lepiej od razu zacząć naukę od HTML5? Ponieważ język ten jest identyczny składniowo i pozostały w nim praktycznie wszystkie techniki pracy opisane w kursie, warto poznać najpierw XHTML, który jeszcze przez kilka lat będzie głównym standardem pracy, a następnie uzupełniać swoją wiedzę o wspierane przez nowsze wersje przeglądarek znaczniki HTML5 aby sukcesywnie zacząć wprowadzać je do swoich projektów. Ostateczne wdrożenie obsługi HTML5 do silników przeglądarek jest planowane po 2020 roku i wtedy bezpiecznie będzie można korzystać ze wszystkich elementów tego języka.

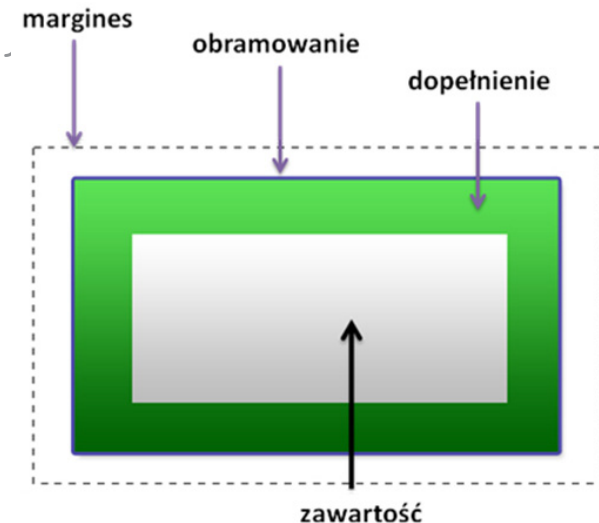
HTML i Flash

Wiele osób konfrontuje ze sobą te technologie, sugerując, że wykorzystanie jednej wyklucza wprowadzenie drugiej. Tymczasem należy rozpatrywać HTML i Flash łącznie, ponieważ flashowe elementy wnoszą zupełnie nową jakość do stron HTML, a sam Flash nie funkcjonuje w sieci bez struktury HTML w której jest osadzony.

HTML5 nie jest tak naprawdę ani wyzwaniem, ani zagrożeniem dla Flasha. Poziom zaawansowania Flasha w wielu aspektach o kilka lat wyprzedza to, co możemy zrobić z pomocą HTML5. Jednak niektóre, proste projekty będziemy mogli teraz wykonać z łatwością wyłącznie w HTML5, bez użycia Flash'a, nie rezygnując z treści takich jak video.

Obecnie jednak najlepszym zastosowaniem będzie mieszanie tych technologii na stronach - tworzenie głównej struktury strony w XHTML i CSS a dodatkowych elementów interaktywnych z pomocą Flasha. Technologia firmy Adobe zostanie też zapewne niepokonana w dziedzinie udostępniania zróżnicowanej treści na urządzenia mobilne, jak również tworzenia bogatych aplikacji (RIA) działających po stronie przeglądarki.

W kontekście zmian które mają miejsce w zakresie projektowania na potrzeby sieci, uważam że konieczne będzie przynajmniej rozumienie obu tych technologii, po to by móc zdecydować, którą z nich wykorzystać na potrzeby konkretnego projektu. Często właściwa decyzja może kilkukrotnie przyspieszyć pracę a połączenie HTML5 i Flash na pewno będzie jeszcze przez wiele lat najbardziej pożądanym zabiegiem na profesjonalnych witrynach WWW.



Model pudełkowy składa się z wielu składowych. Pamiętaj, że na rzeczywisty rozmiar elementu składa się nie tylko właściwość `width`. Należy do niej dodać grubość obramowania oraz marginesy.

Marginesy dwóch obiektów, które znajdują się obok siebie, nakładają się i redukują do największej wartości.

Układ Strony

Każdy element modelu pudełkowego ma także swoje miejsce w tzw. układzie strony. Każdy z kontenerów zabiera określoną ilość miejsca i może dwójako wpływać na inne elementy w tym układzie. Kontenery możemy podzielić na dwa rodzaje:

- Blokowe - zajmują one całą dostępną szerokość w dokumencie rozciągając się w poziomie i w ten sposób wypychając obiekty, które znajdują się po prawej do kolejnego wiersza. Są to przykładowo sekcje `<div>` czy znaczniki `<p>`, `<h1>`, ``
- Liniowe - są rozmieszczone wewnątrz kontenera w którym się znajdują w taki sposób, że nie wypychają innych elementów i zajmują tyle miejsca, co ich zawartość. Są to znaczniki takie jak `<a>`, `` czy ``

tip

Z pomocą CSS możemy wymusić na elemencie liniowym, aby zachowywał się tak, jak element blokowy. Wykorzystujemy do tego zapis np:

```
a {
  display: block;
}
```

Oddziałując w ten sposób na linku sprawimy, że cały wiersz będzie można wygodnie kliknąć, co jest w praktyce często spotykane na stronach WWW.

Dostępność jest ważna

Dostępność to niezwykle ważny aspekt pracy z kodem HTML i CSS. Pozwoli ona osobom, które korzystają z Twojej witryny przy użyciu specjalnego oprogramowania np. do czytania z ekranu (w przypadku osób niedowidzących), sprawnie przegłądać

treści. Jednak dostępność ma także inne aspekty i tak naprawdę odnosi się do wszystkich, którzy korzystają z Twojej strony. Dostępność (accessibility) biorą pod uwagę także roboty indeksujące wyszukiwarki w procesie pozycjonowania. Poniżej znajdziesz kilka wskazówek na temat dostępności.

Wykorzystuj atrybuty alt

Atrybut alt w specyfikacji w3c jest obowiązkowym elementem tagu `` i ma służyć za źródło alternatywnej treści w przypadku braku możliwości wyświetlenia obrazka. Podczas walidacji narzędziami w3c tagi `img`, którym nie przypisano atrybutu alt będą zgłaszać błędy. Dlatego w niektórych przypadkach warto stosować tzw. null alt, czyli pusty atrybut alt w przypadku grafik, które mają mniejsze znaczenie na stronie lub stanowią tło. Możemy wówczas zastosować zapis `alt=""`, pozostawiając pusty cudzysłów.

Warto przemyśleć odpowiednio strukturę i nazewnictwo atrybutów alt po to, aby zawrzeć w nich jak najwięcej słów kluczowych, które zaindeksuje robot wyszukiwarki. Musisz jednak pamiętać, że słowa te muszą występować lub wiązać się bezpośrednio z obrazkiem. Przygotuj atrybuty alt tak, jakby miał je czytać człowiek, nie maszyna.

Wielkość liter

Stosuj małe litery w kodzie html zarówno do określania atrybutów jak i przy nazwach tagów. Podobnie postępuj, jeśli w kodzie html korzystasz z powiązania z javascript. Wbrew twierdzeniom, które można znaleźć w sieci, HTML rozróżnia wielkość liter, można jednak powiedzieć, że nie ma ona co do zasady większego znaczenia i duże litery również są poprawnie interpretowane. Przyjęto się jednak, że elementy jak atrybuty czy nazwy tagów zawsze zapisujemy małymi literami. Jednakże wielkość liter ma znaczenie w treści atrybutów i jeśli przykładowo wykorzystasz nazwę klasy `class="TwojaKlasa"`, jest to jak najbardziej poprawne, a w CSS pamiętaj, aby przy tworzeniu konkretnej deklaracji w nazwie selektora również zastosować duże i małe litery.

Kiedy stosować tło w CSS?

W nawiązaniu do poprzedniego akapitu, warto zastanowić się, kiedy umieszczać obrazki na stronie z pomocą ``, a kiedy robić to z wykorzystaniem CSS i właściwości `background`. W mojej ocenie, jeśli obrazek występuje wyłącznie jako tło jakiegoś elementu, powinniśmy osadzić go z pomocą CSS. Jeśli jednak jest on pewną odrębną całością znaczeniową na stronie i chcesz, aby znalazł uzasadnienie w jej strukturze semantycznej, wykorzystaj znacznik ``.

Sposób na logo

Logo jest zazwyczaj umieszczone w dobrze widocznej części na górze witryny. Istnieją dwa interesujące sposoby na osadzenie logo, które jednocześnie często jest nagłówkiem `h1` ważnym dla pozycjonowania treści. Dlatego po to, by miał on jakieś znaczenie dla robotów indeksujących, logo możemy osadzić w znaczniku `` dla którego nadajemy atrybut `alt` z nazwą firmy i słowem kluczowym. Drugi sposób to podłożenie obrazka jako tło w CSS dla kontenera, w którym jest logo, a w kontenerze umieszczenie słów kluczowych i usunięcie ich z pola widzenia z pomocą właściwości `display` lub `text-indent`. Przykładowy kod prezentuje się następująco:

```
<div id="logo">Twoja Firma - Słowo Kluczowe</div>
```

```
#logo {  
  background-image:url(logo.png) no-repeat;  
  text-indent: -666em;  
}
```

UWAGA! Pamiętaj o tym, aby w opisach `alt` bądź w zawartości kontenera, którą ukrywasz, znalazł się tekst, który widnieje na obrazku lub w sposób bezpośredni go opisuje. W żadnym przypadku nie stosuj listy słów kluczowych, ponieważ może zostać to potraktowane przez robota indeksującego jako spam!

Reguła @font-face

Teraz masz możliwość umieszczania własnych fontów na witrynie w ten sposób, że będą one widoczne jako tekst dla odwiedzającego, bez konieczności posiadania ich w systemie. Odpowiedzialna za ten proces jest tzw. reguła @font-face, która pozwala dołączyć do strony konwertowane wcześniej na odpowiedni format pliki z fontami. Składnia reguły @font-face prezentuje się następująco:

tip

```
@font-face { font-family: Delicious; src: url('Delicious.otf'); }
p { font-family: Delicious, sans-serif; }
```

Za każdym razem, kiedy konwertujemy font na taki, który można wykorzystać z pomocą @font-face, powinniśmy jednak upewnić się, że nie naruszamy w ten sposób warunków licencyjnych. Więcej informacji a także setki fontów, które możesz za darmo wykorzystać na swojej witrynie, znajdziesz na stronach:

<http://www.fontsquirrel.com/>

<http://www.typekit.com/>

<http://www.typhoteque.com/>

Google Fonts

Google udostępniło ciekawy sposób na wykorzystanie dowolnych krojów pisma na Twojej witrynie. Biblioteka Google Font API zawiera setki fontów, które możesz dosłownie w parę chwil zastosować na swojej stronie. Jest ona zlokalizowana pod adresem: <http://www.google.com/webfonts>

Po to, żeby osadzić na swojej stronie fonty, wystarczy, że posłużysz się odpowiednim linkiem do bibliotek google w sekcji <head>. Może to wyglądać na przykład w taki sposób:

```
<head>
  <link rel="stylesheet" type="text/css" href="http://fonts.googleapis.com/
css?family=Tangerine">
  <style>
    body {
      font-family: 'Tangerine', serif;
      font-size: 48px;
    }
  </style>
</head>
<body>
  <h1>Making the Web Beautiful!</h1>
</body>
```

Jak widzisz, z pomocą stylów CSS wykorzystaliśmy na stronie font Tangerine, który będzie poprawnie wyświetlany nawet u osób, które nie mają tego kroju zainstalowanego w systemie. Co więcej, podlinkowanie do zewnętrznych zasobów odciąża nasz serwer, a Google zapewnia, że metoda ta działa dla przeglądarek dostępnych od czasów IE 6 wwyż!

KORZYSTANIE Z PŁYTY DVD DOŁĄCZONEJ DO KSIĄŻKI

Na płycie DVD znajdziesz pełne szkolenie z zakresu Adobe Dreamweaver które jest dodatkiem do książki.

Prosimy o pobranie licencji, która jednocześnie jest dowodem zakupu oraz legalności kursu. Certyfikat jest dostępny na stronie <http://www.eduweb.pl> po zalogowaniu do Strefy Klienta.

INSTRUKCJA OBSŁUGI PŁYTY DVD

Po włożeniu płyty DVD do napędu proszę wybrać opcję Eksploruj (wyświetlić zawartość płyty), a następnie uruchomić plik index.html W razie problemów z odtwarzaniem filmów należy:

- 1/ Pobrać najnowszy Flash Player® (<http://get.adobe.com/pl/flashplayer/>)
- 2/ Zainstalować środowisko Java® (<http://www.java.com/pl/download/>)
- 3/ Wyczyścić pliki tymczasowe i uruchomić ponownie przeglądarkę, a jeśli to nie zadziała, w miarę możliwości spróbować uruchomić plik pod inną przeglądarką (np. Firefox®)
- 4/ W ostateczności zmienić rozszerzenie pliku .flv w folderze video na .mov i sprawdzić, czy film będzie odtwarzał się w QuickTime®

Kurs jest w sposób elektroniczny powiązany z właścicielem licencji. Kurs przeznaczony jest do użytku osoby kupującej. Z wyjątkiem sytuacji dopuszczalnych przez prawo, jakiegokolwiek powielanie, montaż, wypożyczanie czy inne rozpowszechnianie płyty lub jej zawartości czy fragmentów zawartości jest bezwzględnie zabronione.

Książka wraz z płytą DVD nie mogą być przedmiotem odsprzedaży czy sprzedaży handlowej bez uzyskania odpowiedniej licencji.