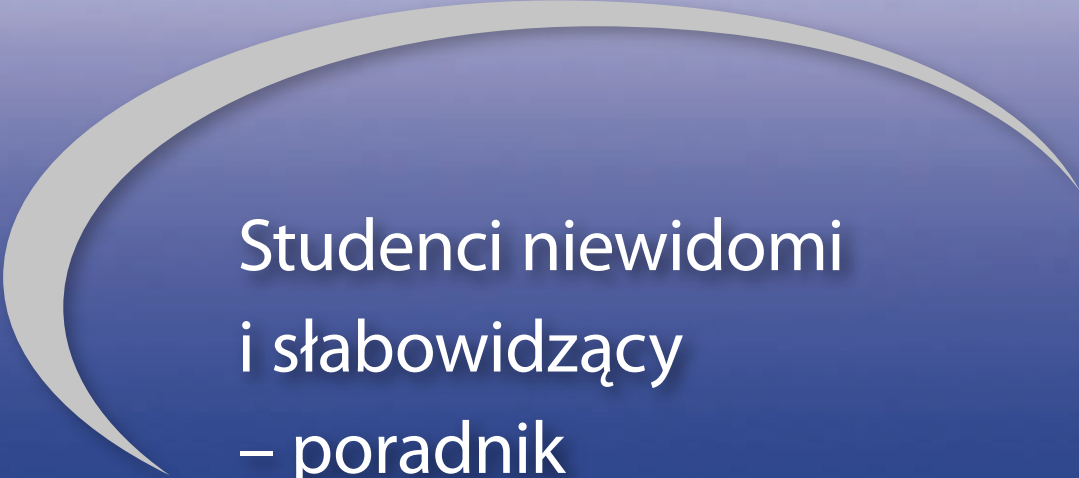


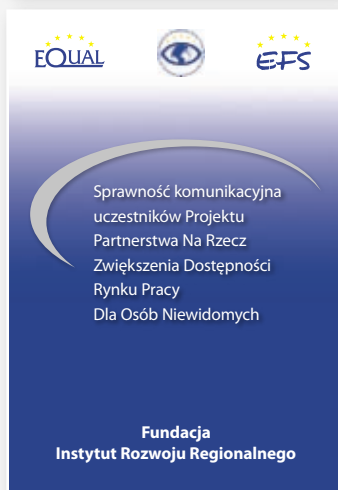
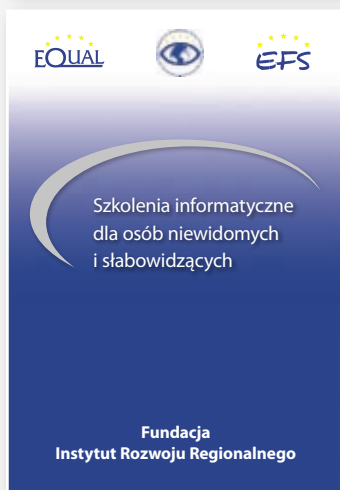
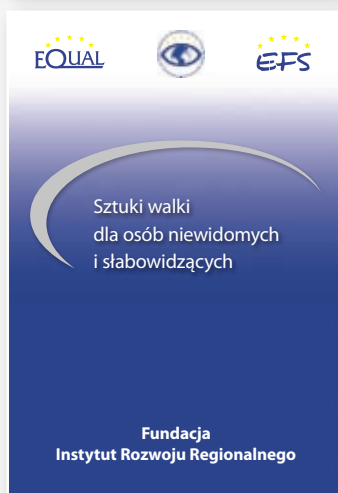
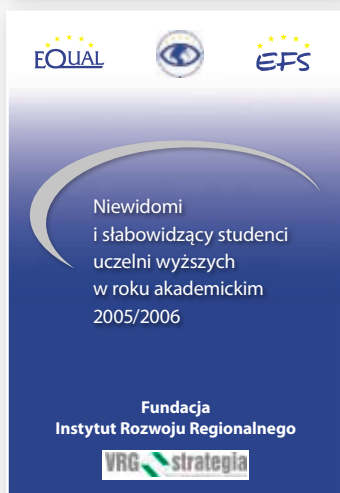
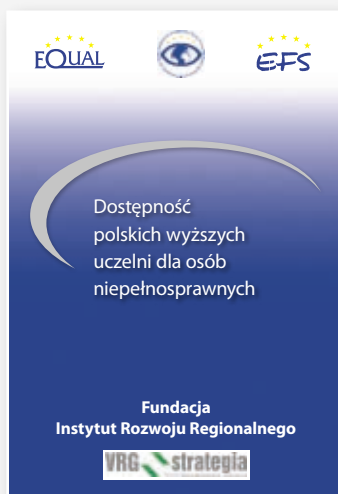
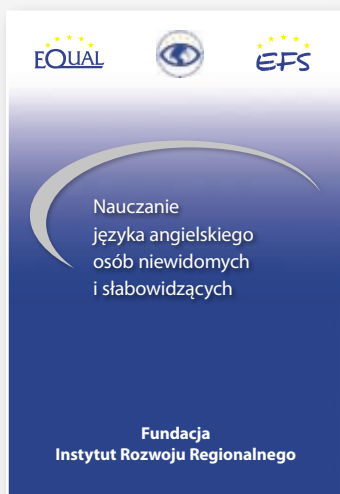


Studenci niewidomi i słabowidzący – poradnik dla wyższych uczelni

Dostępne WWW

**Fundacja
Instytut Rozwoju Regionalnego**

**W serii publikacji Fundacji Instytut Rozwoju Regionalnego
ukazą się również:**



Studenci niewidomi
i słabowidzący
– poradnik
dla wyższych uczelni

Dostępne WWW

**Fundacja
Instytut Rozwoju Regionalnego**

Kraków 2007

Wydawca

Fundacja Instytut Rozwoju Regionalnego
31-261 Kraków, ul. Wybickiego 3A,
tel.: 012 629 85 14, faks: 012 629 85 15
e-mail: biuro@firr.org.pl
<http://www.firr.org.pl>
Organizacja Pożytku Publicznego
KRS: 0000170802
Nr konta 77 2130 0004 2001 0255 9953 0005

Autor

Jacek Zadrożny

Konsultacja

Aleksander Waszkielewicz

Redakcja

Agnieszka Przednowek, Anna Maria Waszkielewicz

Opracowanie graficzne i skład

Studio Graficzne 4DTP
e-mail: 4DTP@ceti.pl

Druk

Omega Druk

ISBN

978-83-61170-04-4

Nakład

1200 egzemplarzy

© Kraków 2007

Wydanie I bezpłatne, nie przeznaczone do sprzedaży.

Opisane w niniejszej publikacji rezultaty powstały w ramach Projektu realizowanego przez Partnerstwo Na Rzecz Zwiększenia Dostępności Rynku Pracy Dla Osób Niewidomych w ramach Programu Operacyjnego Inicjatywy Wspólnotowej EQUAL.

Publikacja współfinansowana z Europejskiego Funduszu Społecznego w ramach Inicjatywy Wspólnotowej EQUAL oraz z budżetu państwa.

Spis treści

Wprowadzenie	5
Czym jest dostępność?	5
Dla kogo dostępność?	6
Po co dostępność?	7
Czy Twój serwis jest dostępny?	7
Dokument internetowy.....	9
Technologie stosowane w Internecie	9
Specyfikacje i ich lekceważenie	10
Struktura dokumentu i jego wygląd.....	10
Wskazówki	11
Informacja tekstowa.....	12
Akapity, nagłówki, listy... ..	12
Tabele	14
Formatowanie tekstu.....	16
Kolory stosowane na stronach.....	16
Czcionki na stronie	17
Wskazówki	18
Multimedia.....	19
Grafika	19
Mapy graficzne	20
Nagrania, filmy i animacje.....	21

4 | Spis treści

Flash i Java	22
Wskazówki	23
Formularze	24
Informacja kontekstowa	24
Grupowanie elementów	30
Automatyzacja w formularzach	31
Zabezpieczenia	32
Wskazówki	33
System nawigacji	34
Interfejs użytkownika	34
Skróty klawiaturowe	34
Język interfejsu i treści	35
Łączy	35
Ramki	36
Dla zagubionych	37
Wskazówki	37
Na zakończenie	38
Wskazówki	39

Wprowadzenie

Temat dostępności informacji, w tym także tej oferowanej w Internecie, jest coraz szerzej poruszany w świecie globalnej sieci. Wciąż jednak brakuje odpowiednich materiałów na ten temat, a dostępność (ang. *accessibility*) często jest mylona z użytecznością (ang. *usability*). Poradnik ten jest próbą odpowiedzi na to zapotrzebowanie i przedstawia temat dostępności zarówno od strony użytkowników, jak i projektantów stron internetowych.

Tych, którzy wzięli poradnik do ręki nie muszę przekonywać, jak wielką rolę we współczesnym społeczeństwie pełni Internet. Jednak zamierzam poświęcić kilka zdań na temat specjalnych grup użytkowników, dla których dostęp do tego medium jest szczególnie ważny, a tworzenie i serwowanie informacji w formie dostępnej stanowi warunek konieczny do korzystania z niego.

W poradniku są prezentowane przykłady kodów w języku HTML i zrzuty ekranów ilustrujące wykorzystanie technik udostępniających. Proszę, by Czytelnika nie zmyliła prostota tych przykładów, bo jest ona zastosowana jedynie w celu łatwiejszej prezentacji przykładów, a nie zalecania stosowania siermiężnego wyglądu. Teksty, tabele czy formularze nie są formatowane, co jednak nie ogranicza twórcy. Zrzuty wykonywane są z przeglądarki MS Internet Explorer z wykorzystaniem jej domyślnych ustawień.

Czym jest dostępność?

Aby przejść do dalszej części poradnika, niezbędne jest zdefiniowanie, co będziemy rozumieć pod pojęciem dostępności.

Dostępność stron internetowych to cecha wynikająca z ich budowy, umożliwiająca pełny, samodzielny, bezpieczny i efektywny dostęp do treści i funkcjonalności, uwzględniająca specyfikę pracy różnych grup użytkowników.

- I **Pełny dostęp** oznacza w tym przypadku, że każda informacja i usługa oferowana na stronie będzie możliwa do odczytania i wykorzystania przez każdego użytkownika korzystającego z powszechnie stosowanych technologii.
- I **Samodzielny dostęp** oznacza, że informacje lub usługi może wykorzystać każdy użytkownik bez potrzeby angażowania innych osób. W szczególności dotyczy to korzystania z Internetu przez osoby niepełnosprawne, dla

których brak samodzielnego dostępu stanowi naruszenie ich samodzielności i suwerenności.

- I **Bezpieczny dostęp** oznacza, że korzystanie z informacji lub usług nie narusza zasad bezpieczeństwa. Sposób realizacji usług musi być zorganizowany w taki sposób, by alternatywne rozwiązania były równie bezpieczne jak podstawowe.
- I **Efektywny dostęp** oznacza, że nie jest wymagany znacząco wyższy wydatek pracy, czasu lub finansowy na osiągnięcie porównywalnego rezultatu.

Z ostatniej części definicji wynika, że dostępność musi być zaplanowana przez projektanta, a nie może być przypadkowa. To planowanie może pojawić się jako implementacja systemu CMS (ang. *Content Management System*), określonej polityki instytucji lub w inny sposób, byle skuteczny.

Dla kogo dostępność?

Odpowiedź na to pytanie jest prosta – dla wszystkich. Serwisy internetowe i działające w ich ramach usługi powinny być dostępne dla każdego użytkownika, niezależnie od używanego przez niego sprzętu, oprogramowania czy też posiadanej niepełnosprawności. Dlatego zdecydowanie złym rozwiązaniem jest tworzenie alternatywnych serwisów (na przykład wersji graficznych i tekstowych) oraz przygotowywanie ich do obsługi przez konkretne oprogramowanie (na przykład obsługiwane za pomocą Internet Explorer w wersji co najmniej 5.5). Projektant serwisu powinien kierować się zasadą tworzenia dla wszystkich, a nie dla największej grupy. Jeżeli bowiem przyjrzeć się tej największej grupie, to wśród niej też można znaleźć znaczące różnice w możliwościach korzystania z Internetu.

Warto w tym miejscu jednak wymienić także te grupy, które mogą mieć problem z korzystaniem z zasobów internetowych:

- I osoby niewidome i słabowidzące, z niedowładem rąk, głuche i niedosłyszące;
- I osoby o obniżonej normie intelektualnej;
- I osoby starsze;
- I daltoniści – wśród mężczyzn około 5% populacji;
- I użytkownicy bardzo starych komputerów i bardzo nowoczesnych rozwiązań typu PDA i SmartPhone.

Każdy z nich powinien mieć równy dostęp do oferowanych treści i usług.

Po co dostępność?

Powodów do tworzenia dostępnych serwisów jest wiele, poniżej przytoczymy kilka najważniejszych:

- Serwis powinien być dostępny dla jak najszerszego kręgu odbiorców, bez pomijania różnych grup, specyficznych ze względu na swoje potrzeby lub sposób dostępu do Internetu.
- Problemy z dostępnością mogą zniechęcać wielu użytkowników do korzystania z serwisu lub wręcz uniemożliwiać im to. To może drastycznie obniżać oglądalność lub liczbę klientów serwisu.
- Stosowanie standardów dostępności znacząco zwiększa możliwość automatycznego indeksowania serwisów przez systemy wyszukiwawcze.
- Serwis zgodny ze standardami jest o wiele łatwiejszy w konserwacji i modyfikacji.

Czy Twój serwis jest dostępny?

Po przeczytaniu wstępu może Cię zaciekać, czy Twój serwis jest dostępny. Odpowiedź na to pytanie nie jest prosta, więc proponujemy wykonanie kilku testów, które dadzą przynajmniej częściową odpowiedź na to pytanie.

Na początek można sprawdzić, czy Twój serwis jest zgodny ze **standardami internetowymi**, czyli czy poddaje się walidacji. Można do tego celu użyć walidatora dostępnego na stronach *World Wide Web Consortium* lub wtyczki HTML Validator dla przeglądarki Firefox. Polecamy szczególnie to drugie rozwiązanie, ponieważ surfowanie po Internecie i śledzenie paska statusu, na którym pojawia się liczba błędów i ostrzeżeń znalezionych na wyświetlanej właśnie stronie, jest bardzo pouczające.

Teraz w swojej ulubionej przeglądarce proszę **wyłączyć wyświetlanie grafiki**, animacji, Flasha i innych rozszerzeń, JavaScript, obsługę stylów i co tylko można znaleźć jeszcze w panelu konfiguracyjnym. Teraz można otworzyć w przeglądarce swoją stronę i przyrzuć się, czy można z niej korzystać. Jeżeli nie, to znaczy, że są w niej poważne błędy konstrukcyjne. Serwis powinien być przede wszystkim użyteczny, a nie ładny. Atrakcyjność powinna być podporządkowana funkcjonalności, a nie odwrotnie.

Kolejnym etapem może być sprawdzenie, czy stronę i usługi oferowane na niej można **obsłużyć za pomocą klawiatury**. Pomimo, że przeważająca większość użytkowników korzysta z myszy lub podobnego urządzenia wskazującego, to jednak są tacy, którzy preferują klawiaturę. Dla nich szczególnym utrudnieniem są menu rozwijane po najechaniu na nie kursorem oraz niektóre aktywne formularze.

Istnieją także walidatory dostępności i to w ogromnym wyborze. Na świecie problem dostępności został dostrzeżony znacznie wcześniej niż w Polsce i programiści pracowali nad odpowiednimi narzędziami. Najbardziej znany jest Bobby firmy WatchFire, który dokonuje walidacji pod kątem wymagań *Web Accessibility Initiative* i Section 508 – amerykańskiej regulacji ustawowej. Innym tego typu walidatorem jest dostępny pod adresem <http://relaxed.cz> program o nazwie Relaxed. Prosty raport o problemach z dostępnością można otrzymać także we wspominanej już wtyczce do Firefoxa o nazwie HTML Validator.

Pełne testowanie serwisu internetowego pod kątem jego dostępności ma zawsze trzy etapy:

1. Testowanie automatyczne z wykorzystaniem wspomnianych powyżej narzędzi. Efektem tego testu jest raport informujący o odnalezionych błędach i potencjalnych problemach. Ta faza pozwala na odnalezienie oczywistych błędów, w szczególności składniowych.
2. Testowanie półautomatyczne, którego elementem wejściowym jest wygenerowany wcześniej raport, a on z kolei musi być przeanalizowany przez osobę testującą. Tester powinien zwrócić uwagę na ostrzeżenia zawarte w raporcie, przeanalizować je i usunąć ewentualne błędy. Ta faza pozwala na usunięcie problemów, które automat sygnalizuje jako potencjalnie niedostępne, a których nie może samodzielnie jednoznacznie zakwalifikować.
3. Testowanie ręczne, które można wykonać w sposób analogiczny do testowania funkcjonalności, ale z wykorzystaniem grupy testerów o specyficznych potrzebach – osób niepełnosprawnych, starszych i dzieci, a także na mniej typowym sprzęcie i oprogramowaniu. Ta faza testowania pozwala na wykrycie błędów i ograniczeń, których nie da się wykryć w sposób zautomatyzowany.

Dokument internetowy

W dalszej części poradnika zostaną omówione szczegóły tworzenia dostępnych dokumentów internetowych. Pod tym pojęciem będzie rozumiane wszystko, co jest wyświetlone w przeglądarce internetowej – niezależnie od wykorzystanej technologii. Jednak faktycznie poradnik dotyczy stron przygotowywanych w języku HTML i związanych z nim tj. XHTML, CSS poziomu I, II i III.

Technologie stosowane w Internecie

Przeglądarki internetowe są przystosowane do wyświetlania wielu typów dokumentów. Podstawowym jest HTML, a w nowszych wersjach przeglądarek także jego rozszerzona wersja XHTML. Obydwa języki służą do opisu treści dokumentu poprzez oznaczenie jego fragmentów jako akapity, tytuły (nagłówki) różnego poziomu, listy wyliczeniowe i numerowane, akronimy i wiele innych. Dokumenty stworzone w tych językach zawierają także metainformacje, na przykład o stosowanym w nim języku narodowym, stronie kodowej, autorze czy prawach własności. Autorzy specyfikacji zalecają, by do określania wyglądu stosować arkusze stylów (CSS – *Cascading Style Sheets*), w celu oddzielenia treści i struktury dokumentu od jego graficznej prezentacji.

Dzięki rozszerzeniom przeglądarek w dokumentach można wyświetlać także inne rodzaje dokumentów, na przykład Flash, PDF (*Portable Document Format*), pliki pakietu MS Office i inne oraz osadzać aplikacje, na przykład w języku Java i kontrolki ActiveX. Zawsze trzeba przy tym pamiętać, że te elementy mogą być same w sobie niedostępne i to z kilku powodów:

- I oprogramowanie użytkownika może nie interpretować elementów osadzonych;
- I przeglądarka internetowa może nie mieć zainstalowanych odpowiednich wtyczek;
- I zabezpieczenia systemowe nie pozwalają na uruchamianie lub otwieranie niektórych rodzajów plików.

Mając na uwadze powyższe warto zastanowić się nad umieszczaniem podobnych rozwiązań, które są ściśle związane z oprogramowaniem posiadanym przez użytkownika. Jeżeli korzystanie z serwisu jest uzależnione od zainstalowania wirtualnej maszyny Javy, to wielu użytkowników może nie mieć do niego dostępu.

Warto poszukać alternatywnego rozwiązania opartego na standardach *World Wide Web Consortium* i zastąpić nimi te, które nie są pewne. Nawet najbardziej prymitywna przeglądarka internetowa z pewnością interpretuje język HTML, ale nie zawsze wczyta i wyświetli animacje Flash czy grafikę wektorową.

Specyfikacje i ich lekceważenie

Wydaje się oczywistym, że twórcy stron internetowych powinni dbać o ich zgodność ze standardami. Jednak doświadczenie pokazuje, że w procesie tworzenia serwisów internetowych bardzo łatwo popełnić błędy, które uniemożliwiają poprawną walidację. Błędy składniowe są przyczyną wielu problemów zarówno dla autorów, jak i użytkowników. Powodów, dla których należy stosować się do specyfikacji języków jest wiele:

- I błędy składniowe utrudniają konserwację i rozbudowę serwisów;
- I błędy składniowe utrudniają, a czasem wręcz uniemożliwiają stosowanie technik typu DOM (*Document Object Model*) i CSS (*Cascading Style Sheets*);
- I przeglądarka internetowa napotykając błąd składniowy zmuszona jest do jego interpretacji i każda może to robić w odmienny sposób, co powoduje różny sposób wyświetlania strony;
- I niezgodność ze specyfikacjami może uniemożliwić wyświetlenie części lub całości strony;
- I nieprawidłowo skonstruowane strony są gorzej indeksowane przez systemy wyszukiwawcze.

Struktura dokumentu i jego wygląd

Jednym z podstawowych zadań języków HTML i XHTML jest strukturalizacja informacji w dokumencie. Dzięki sześciu poziomom nagłówków można precyzyjnie określić wagę tytułów, a rozbudowana specyfikacja tabel opisuje nie tylko poszczególne komórki, rzędy i kolumny, ale także określa, które są nagłówkami, oraz podsumowanie tabeli.

Autor strony internetowej, szczególnie, jeżeli korzysta z wizualnych narzędzi do tworzenia, nie zawsze zdaje sobie sprawę z możliwości opisu struktury dokumentu. Dobrym przykładem może być tabela, w której nazwy kolumn zostały wyróżnione pogrubioną czcionką. Osoba przeglądająca taką tabelę może wzrokiem odnaleźć te pozycje i zinterpretować je jako nagłówki, ale nie robi

tego program komputerowy, użytkownik niewidomy lub użytkownik przeglądarki tekstowej. Języki HTML i XHTML oferują specjalny znacznik opisujący nagłówki kolumn lub wierszy, które można sformatować za pomocą stylów CSS. Takie zastosowanie wbudowanych możliwości rozwiązuje opisane powyżej problemy.

Wskazówki

1. Serwis internetowy powinien być dostępny dla każdego użytkownika.
2. Dodatkowe wersje dla specyficznych grup użytkowników powinny być tworzone tylko w ostateczności.
3. Należy przestrzegać standardów i specyfikacji technicznych, w szczególności opublikowanych przez *World Wide Web Consortium*.
4. Inne technologie (na przykład Flash, Java, PDF) powinny być stosowane z ostrożnością i tylko wówczas, gdy autor upewni się, że one także są dostępne.
5. Należy oddzielać strukturę i treść dokumentu od jego graficznej prezentacji.

Informacja tekstowa

U swoich narodzin Internet opierał się niemal wyłącznie na informacji tekstowej. Wraz z upływem lat, wzrostem przepustowości łączy i pojemności twardych dysków strony stały się coraz bardziej multimedialne. Jednak wciąż tekst jest podstawowym medium informacji oferowanej w Internecie. Dlatego warto stosować związane z nim elementy w sposób prawidłowy, co zapewni dokumentom logiczną strukturę, a użytkownikom ułatwi pracę.

Akapity, nagłówki, listy...

Podstawową logiczną częścią tekstu jest akapit (paragraf), który oznaczany jest w kodzie znacznikiem P. Taka zamknięta porcja tekstu może zawierać w sobie mniejsze elementy na przykład skróty, akronimy, cytaty i wiele innych. Te także można odpowiednio oznaczyć za pomocą znaczników języka HTML lub XHTML. Takie logiczne wydzielenie fragmentów tekstu znacząco ułatwia jego prawidłową interpretację. Warto pamiętać, że oznaczanie wyłącznie w sposób wizualny nie zawsze się sprawdzi. Niektóre przeglądarki mogą nie mieć możliwości wyświetlenia kursywy, wykorzystują więc inne mechanizmy do prezentacji cytatów, ale tylko wtedy, gdy ta informacja jest im prawidłowo dostarczona.

Powyższe uwagi dotyczą w takim samym stopniu innych elementów stosowanych w tekście na przykład list wyliczeniowych, numerowanych i definicji. Środowiska OL, UL i DL pozwalają na strukturalizowanie informacji tekstowej w stopniu znacznie wyższym niż za pomocą technik wyłącznie wizualnych. Umieszczenie grafiki przed akapitem nie czyni z niego automatycznie elementu listy, ponieważ przeglądarki nie zawsze będą ją wyświetlać. Podobnie odpowiednie oznaczenie definicji – podział na hasło i objaśnienie – pozwoli na jednoznaczny identyfikację obydwu elementów i poprawną interpretację.

Ważną rolę w dokumentach pełnią tytuły oznaczane za pomocą znaczników H1, H2, H3, H4, H5 i H6. Pozwalają one na tytułowanie poszczególnych części dokumentu, dzieląc go jednocześnie na mniejsze części. Cyfry w poszczególnych znacznikach określają wagę tytułu i kolejność zagnieżdżenia. Dlatego znacznik H2 powinien znajdować się po znaczniku H1, a poprzedzać znacznik H3. W przeciwnym razie struktura logiczna dokumentu zostanie zaburzona, co utrudni nie tylko odbiór informacji przez użytkowników, ale także utrudni automatyczne przetwarzanie. Poniżej znajduje się przykład prawidłowej struktury zagnieżdżenia nagłówków, a także konstrukcji definicji i list numerowanych i wyliczeniowych:

```

<body>
<h1>Tytuł całego dokumentu</h1>
<p>Tutaj znajduje się wstęp do całego dokumentu, choć
oczywiście nie jest obowiązkowy. Przecież prawie nikt
nie czyta wstępów.</p>
<h2>1. Pierwszy rozdział</h2>
<dl>
<dt>Dostępność stron internetowych</dt><dd> to cecha
wynikająca z ich budowy, umożliwiająca pełny,
samodzielny, bezpieczny i efektywny dostęp do treści
i funkcjonalności, uwzględniająca specyfikę pracy
różnych grup użytkowników.</dd>
</dl>
<h3>Podrozdział</h3>
<p>Tutaj jest jakiś mały podrozdział ilustrujący
zagnieżdżanie nagłówków.</p>
<h3>Drugi podrozdział</h3>
<p>Jak to w przykładach... Kończą mi się pomysły
na przykładowe teksty.</p>
<h2>2. Drugi rozdział</h2>
<p>Ten rozdział nie ma wydzielonych podrozdziałów, co
oszczędzi mi znaczną część czasu i inwencji na pisanie
tekstów, których nikt potem nie przeczyta... A może
jednak... Proszę o maila w tej sprawie.</p>
<h2>3. Trzeci rozdział</h2>
<p>Tutaj zaprezentuję także wykorzystanie list
numerowanych i wyliczeniowych.</p>
<h3>Listy numerowane</h3>
<ol>
<li>Internet jest dla wszystkich.</li>
<li>Informacja jest dla wszystkich, którzy zechcą
do niej sięgnąć.</li>
<li>Informacja powinna być dostępna dla wszystkich
niezależnie od używanego sprzętu, oprogramowania lub
posiadanej niepełnosprawności.</li>
</ol>
<h3>Lista wyliczeniowa</h3>
<ul>
<li>Internet jest dla wszystkich;</li>
<li>informacja jest dla wszystkich, którzy zechcą
do niej sięgnąć;</li>

```

informacja powinna być dostępna dla wszystkich niezależnie od używanego sprzętu, oprogramowania lub posiadanej niepełnosprawności.

</body>



Fot. 1. Przykładowe wykorzystanie struktury nagłówków i list.

Tabele

Tabele w języku HTML są wykorzystywane do dwóch celów: przedstawiania danych tabelarycznych i tworzenia układu strony. Pierwsze zastosowanie jest podstawowe i tabele do tego powinny być przede wszystkim używane. Wykorzystywanie tabel do tworzenia układu strony jest rozwiązaniem przestarzałym, ponieważ techniki formatowania za pomocą pozycjonowania elementów w arkuszach stylów są na tyle rozwinięte, że stosowanie tabel staje się niepotrzebne.

Układ tabelaryczny sprawia najwięcej kłopotów użytkownikom niewidomym, którzy odczytują jej zawartość w sposób liniowy, przechodząc po kolei przez komórki od lewej do prawej w kolejnych wierszach od góry do dołu. Dlatego, przygotowując tabelę z danymi, należy stosować kilka zasad: tabela powinna być (w miarę możliwości) jednorodna, to znaczy powinna mieć równą liczbę komórek w każdym wierszu i kolumnie. Oznacza to, że łączenie komórek powinno być realizowane z umiarem i tylko w razie konieczności. Należy przyjrzeć się, czy tabela odczytywana w sposób liniowy zachowuje sens. Mając tabelę zawierającą nazwy województw i liczbę ich mieszkańców lepiej zastosować układ pionowy, ponieważ wówczas użytkownik będzie otrzymywał informację o nazwie województwa, a po niej liczbę mieszkańców. Układ poziomy zaprezentuje mu najpierw wszystkie nazwy województw, a potem kolejne liczby, które nie dadzą się łatwo powiązać z konkretnym województwem.

Na potrzeby osób korzystających z oprogramowania udźwiękowiającego do specyfikacji HTML włączono specjalny atrybut SUMMARY, który zawiera opisową informację o tabeli. Informacja ta nie jest wyświetlana w przeglądarce internetowej, a przekazywana jedynie użytkownikowi niewidomemu. Atrybut SUMMARY jest czymś innym niż podpis tabeli umieszczany w znaczniku CAPTION i może wyglądać tak jak w poniższym kodzie:

```
<table summary="tabela zawiera informacje o liczbie
ludności w poszczególnych województwach. Pierwsza kolumna
zawiera nazwę województwa, a druga liczbę mieszkańców">
<caption>Liczba ludności w poszczególnych
województwach</caption>
<tr><th>Województwo</th><th>Ludność w tysiącach</th></tr>
<tr><td>dolnośląskie</td><td>2888</td></tr>
<tr><td>kujawsko-pomorskie</td><td>2068</td></tr>
<tr><td>lubelskie</td><td>2191</td></tr>
<tr><td>lubuskie</td><td>1009</td></tr>
<tr><td>łódzkie</td><td>2597</td></tr>
<tr><td>małopolskie</td><td>3253</td></tr>
<tr><td>mazowieckie</td><td>5135</td></tr>
<tr><td>opolskie</td><td>1055</td></tr>
<tr><td>podkarpackie</td><td>2097</td></tr>
<tr><td>podlaskie</td><td>1205</td></tr>
<tr><td>pomorskie</td><td>2189</td></tr>
<tr><td>śląskie</td><td>4715</td></tr>
<tr><td>świętokrzyskie</td><td>1292</td></tr>
<tr><td>warmińsko-mazurskie</td><td>1429</td></tr>
<tr><td>wielkopolskie</td><td>3372</td></tr>
<tr><td>zachodniopomorskie</td><td>1696</td></tr>
</table>
```

Województwo	Ludność w tysiącach
dolnośląskie	2888
kujawsko-pomorskie	2068
lubelskie	2191
lubuskie	1009
łódzkie	2597
małopolskie	3253
mazowieckie	5135
opolskie	1055
podkarpackie	2097
podlaskie	1205
pomorskie	2189
śląskie	4715
świętokrzyskie	1292
warmińsko-mazurskie	1429

Fot. 2. Przykładowa tabela z podpisem.

W powyższym przykładzie nie zastosowano żadnego specjalnego formatowania samej tabeli ani zawartych w niej danych. Formatowanie za pomocą stylów nie będzie miało żadnego wpływu na odbiór przez osoby niewidome, ale może mieć wpływ na odbiór przez osoby słabowidzące. Dlatego w tabelach należy stosować podobne zasady, jak w przypadku innych danych tekstowych.

Formatowanie tekstu

Jednym z podstawowych zaleceń *World Wide Web Consortium* jest oddzielanie treści internetowych od jej graficznej prezentacji. W szczególności chodzi tu o formatowanie za pomocą zewnętrznych arkuszy stylów CSS i stosowanie warstw DIV. Takie podejście do wizualizacji ma jeszcze tę zaletę, że dokonywanie zmian w zewnętrznym arkuszu stylów ma wpływ na każdy dokument, w którym jest odwołanie do niego. Nie trzeba więc modyfikować serwisu w wielu miejscach, lecz wystarczy w jednym.

Ze względu na czytelność należy unikać stosowania bardzo różnorodnych układów graficznych na przykład różnych krojów pisma, czy kolorów. Warto jednak wyraźnie oddzielić część główną strony od systemu nawigacji, aby użytkownik nie miał wątpliwości przy odnajdowaniu łącz.

Kolory stosowane na stronach

Dobór odpowiednich kolorów do serwisu internetowego jest sporym problemem do rozwiązania. Należy tu wyważyć pomiędzy zasadami dostępności, a estetyką i atrakcyjnością serwisu. Najważniejsze jednak, by kierować się kilkoma podstawowymi wskazówkami:

- l informacja nie powinna być przekazywana wyłącznie za pomocą koloru; jeżeli projektant zamierza wyróżnić fragment tekstu, to powinien to zrobić nie tylko kolorem, ale także za pomocą innych rozwiązań (np. pogrubienia, podkreślenia itp.);
- l należy stosować odpowiedni kontrast pomiędzy tłem, a elementami tekstowymi; przynajmniej w stosunku jasności 1:5;
- l należy unikać bardzo jaskrawych kolorów, szczególnie w tle strony;
- l tło powinno być jednolite, bez wzorków i deseni;
- l należy unikać migających i poruszających się elementów.

Nowoczesne przeglądarki internetowe wyposażono w rozwiązania techniczne pomagające w korzystaniu ze stron internetowych osobom słabowidzącym. Użytkownik może dostosować kolorystykę strony do własnych potrzeb poprzez wymuszenie stosowania innych stylów niż zaproponowane przez projektanta. Jednak nie każdy ma odpowiednie umiejętności, aby sobie samodzielnie poradzić z konfiguracją, więc to projektant powinien zadbać o to, by serwis był dostępny także przy domyślnych ustawieniach.

Czcionki na stronie

Ze względu na różnorodność schorzeń wzroku nie da się jednoznacznie określić, jaka wielkość czcionek stosowanych na stronie jest optymalna. Zbyt małe czcionki zmuszają użytkownika do wyężdżania wzroku, co nie jest zbyt przyjemne, zaś zbyt duże wyglądają nieestetycznie, a ich wykorzystanie drastycznie zmniejsza ilość informacji dostępnej na stronie. Najlepszym rozwiązaniem jest zatem stosowanie czcionek przeciętnej wielkości: **10-12 punktów**, jednak w taki sposób, by dało się je powiększyć za pomocą mechanizmów wbudowanych w przeglądarkę internetową. Mechanizmy takie posiadają najpopularniejsze przeglądarki: MS Internet Explorer, Opera, Firefox, Mozilla.

Warto w tym miejscu przypomnieć o istnieniu czcionek specjalnie zaprojektowanych do czytania na ekranie komputera, a więc także do stosowania na stronach internetowych. Należą do nich między innymi: **Verdana**, **Helvetica**, **Tahoma** i kilka innych. Odpowiednie dobranie wielkości oczek liter i kształtu sprawia, że są lepiej odbierane przez użytkowników i mniej męczą wzrok.

Jeżeli z jakiegoś powodu nie można użyć czcionek zalecanych, to przy doborze warto kierować się zasadą stosowania jak najpopularniejszych krojów, na przykład **Arial** i **Times**. Kroje te są doskonale znane wszystkim użytkownikom i mogą być czytane łatwiej, częściowo „na pamięć”. Zastosowanie nietypowego kroju, czy wręcz czcionki ozdobnej, może sprawić wiele trudności osobom słabowidzącym.

Wskazówki

1. Dokument powinien być skonstruowany w sposób logiczny z wykorzystaniem dostępnych znaczników opisu treści.
2. Tabele powinny być jednorodne, zachowywać sens po przekształceniu w sposób liniowy i zawierać dodatkową informację opisową.
3. Należy unikać nadmiernej różnorodności krojów czcionek i kolorystyki.
4. Warto wyraźnie oddzielić treść dokumentu od systemu nawigacyjnego.
5. Należy unikać stosowania jaskrawych kolorów oraz migających i poruszających się elementów.
6. Kontrast pomiędzy tłem a treścią powinien wynosić co najmniej 1:5.
7. Czcionki umieszczane na stronie powinny mieć wielkość co najmniej 10 punktów a konstrukcja strony powinna dawać możliwość ich powiększenia.
8. Najlepiej stosować kroje czcionek zaprojektowane do wyświetlania na ekranie monitora lub najbardziej znane.

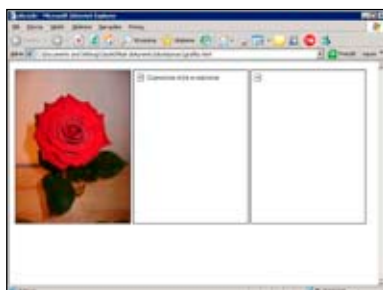
Multimedia

Trudno sobie wyobrazić serwis internetowy bez grafiki, animacji, filmów czy nagrań. Internet jest multimedialny, ponieważ taka jest tendencja w przekazywaniu informacji. Problem polega tylko na tym, jak taką formę informacji uczynić dostępną dla wszystkich użytkowników – udostępnić grafikę dla osób niewidomych, nagrania dla głuchych, a filmy dla użytkowników archaicznych przeglądarek. Należy w tym miejscu powiedzieć, że udostępnianie serwisów nie może polegać na usuwaniu tych multimedii i tworzeniu wersji wyłącznie tekstowych. Nie wolno zamykać użytkowników sieci w gettach specjalnych serwisów przeznaczonych dla wąskich grup odbiorców. O takiej wersji łatwo zapomnieć, a zapomnienie oznacza nieaktualność. Niech raczej projektant stworzy serwis, który będzie tak samo dostępny i użyteczny dla każdego odwiedzającego.

Grafika

Umieszczając na stronie internetowej grafikę trzeba pamiętać o tym, że nie każdy użytkownik jest w stanie ją wyświetlić w swojej przeglądarce. W takim przypadku przeglądarka wyświetli informację alternatywną, której treść zależy od autora strony. W tym celu w specyfikacji języka HTML znajduje się atrybut ALT, którego wartość stanowi opis grafiki. Oto przykład zastosowania:

```
<table><tr>
<td></td>
<td></td>
<td></td>
```



Fot. 3. Grafika wstawiona na stronę internetową z wykorzystaniem atrybutu ALT i bez niego.

Jak widać pierwsze zdjęcie wyświetlone jest prawidłowo. Jednak pozostałe dwa nie są widoczne. W przypadku środkowego zastosowanie opisu alternatywnego pozwala na zorientowanie się, co przedstawia grafika, co jest niemożliwe w wypadku ostatniej grafiki, ponieważ autor zapomniał o atrybucie ALT.

Priorytetowo należy traktować grafiki, które są aktywnymi elementami strony, na przykład linkami lub przyciskami. Bez alternatywnego opisu użytkownik nie jest w stanie zorientować się, jaki będzie skutek kliknięcia w taki przycisk albo dokąd doprowadzi go łącze. Grafika ilustracyjna powinna mieć opisy alternatywne, jeżeli autor chce poinformować użytkowników, co znajduje się na ilustracji. Odmienne należy jednak potraktować grafikę ornamentową, na przykład różnego rodzaju kropki, paski i ramki. Opisywanie ich w sposób alternatywny jest nie tylko bezcelowe, ale wprowadza nadmiar zbędnej informacji. Aby jednak strona przeszła poprawnie walidację dostępności, należy wprowadzić dla takiego elementu atrybut ALT z pustą wartością to znaczy dwa puste cudzysłowy.

Czasem może się zdarzyć, że precyzyjny opis rysunku jest bardzo ważny i nie wolno go pominąć. Atrybut ALT się tu do końca nie sprawdza, ponieważ może on zawierać tylko stosunkowo prostą informację tekstową. W takiej sytuacji można zastosować atrybut LONGDESC, który kieruje użytkownika do szczegółowego opisu umieszczonego na oddzielnej stronie:

```

```

Mapy graficzne

Szczególnym rodzajem grafiki są mapy graficzne, czyli grafiki, w których wydzielone rejony są aktywne i pełnią rolę linków. W języku HTML zdefiniowano dwa rodzaje takich map:

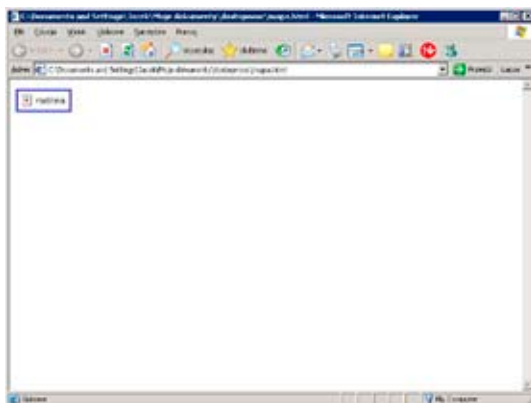
- I mapy wykonywane po stronie klienta realizowane za pomocą poleceń MAP i AREA,
- I mapy wykonywane po stronie serwera, gdy przeglądarka wysyła do serwera koordynaty miejsca kliknięcia.

Pierwszy rodzaj map jest stosunkowo prosty do udostępnienia i wymaga jedynie pamiętania o atrybutach ALT dla każdego aktywnego elementu oraz obiektu będącego mapą. Drugi rodzaj map nie jest możliwy do udostępnienia.

Na poniższym przykładzie można prześledzić technikę udostępniania mapy klienckiej z wykorzystaniem atrybutów ALT:

```

<map name="rodzina">
<area href="tata.html" shape="circle" coords="100,100,30"
alt="Tata, czyli głowa rodziny">
<area href="mama.html" shape="circle" coords="170,120,30"
alt="Mama, czyli prawdziwa głowa rodziny">
<area href="dziecko.html" shape="circle"
coords="200,130,30" alt="Dziecko, zazwyczaj cwańsze
od obu głów rodziny">
</map>
```



Fot. 4. Przykład udostępnionej mapy bitowej bez załadowanego obrazka.

Dzięki opisom alternatywnym użytkownik może korzystać z mapy bitowej, mimo, że obrazek nie został załadowany przez przeglądarkę. Również niewidomi użytkownicy mają swobodny dostęp do nawigacji.

Nagrania, filmy i animacje

Multimedia mają różne warstwy informacyjne:

- wizualną – w wypadku grafiki, animacji i filmów;
- dźwiękową – w wypadku filmów i nagrań.

Do każdej z nich dostęp może być utrudniony. Do warstwy wizualnej nie mają dostępu osoby niewidome, a do dźwiękowej – osoby głuche. Jednak nie tylko osoby niepełnosprawne mogą spotkać się z tego rodzaju ograniczeniami. Trudno jest wysłuchać nagrania w dużym hałasie lub obejrzeć animację na ekranie

małego telefonu komórkowego nawet dla osoby całkowicie sprawnej. Obydwa rodzaje warstw informacyjnych powinny więc być deskrybowane do zwykłego tekstu, który jest dostępny dla wszystkich w największym stopniu.

Umieszczenie materiałów multimedialnych w serwisie internetowym jest możliwe co najmniej na dwa sposoby:

- I osadzenie na stronie za pomocą polecenia OBJECT lub w podobny sposób;
- I podlinkowanie do zasobu za pomocą polecenia A.

Pierwszy sposób pozwala znacząco podnieść atrakcyjność serwisu internetowego, wypełniając go obrazem i dźwiękiem. Jednak trzeba wziąć pod uwagę fakt znaczącego wzrostu zapotrzebowania na przepustowość łączy, gdy multimedia są pobierane automatycznie. Drugim poważnym problemem są zakłócenia generowane przez takie rozwiązania. Osoby korzystające z oprogramowania mówiącego (udźwiękawiającego) mogą mieć duży dyskomfort, gdy w tle strony słyszą muzykę odtwarzaną w zapętleniu. Poza tym, jeżeli użytkownik ma w przeglądarce wyłączone pobieranie i odtwarzanie multimediów, to nie będzie miał do nich dostępu.

Z tych powodów należy zalecić raczej **podlinkowywanie zasobów multimedialnych**, zamiast osadzania ich na stronie. Umieszczenie na stronie odpowiedniego łącza prowadzącego do zasobu multimedialnego pozwoli użytkownikowi na świadomy wybór, czy chce oglądać film lub odsłuchać nagranie, czy też nie. Dodatkowo może sam wybrać, z jakiego odtwarzacza skorzysta. Oprócz samego łącza, należy dostarczyć użytkownikowi informacje o formacie pliku, jego wielkości oraz długości trwania nagrania. Wreszcie trzeba dostarczyć opis deskrybujący zawartość multimedialną: obraz i dźwięk w sytuacjach, gdy treść jest istotna. Umieszczając na stronie nagranie przemówienia należy umieścić także tekstową wersję tego przemówienia, a w wypadku animacji ilustrującej jakieś zjawisko fizyczne – opis tego zjawiska. Niecelowe jest natomiast deskrybowanie informacji mało istotnych, ponieważ wymagają dodatkowego nakładu pracy, a są zbędne. W wypadku nagrania śpiewu skowronka wystarczy podać informację o tym w tekście łącza (na przykład „nagranie śpiewu skowronka”) i nie próbować opisywać tego bardziej szczegółowo.

Flash i Java

Pomimo szybkiego rozwoju oprogramowania dostępowego i wysiłków ze strony firm Adobe i Sun zarówno Flash, jak i Java są technologiami mało dostępnymi dla osób niewidomych. Umieszczanie ich na stronach internetowych, a szczególnie

uzależnianie od nich działania serwisu jest dość ryzykowne. Należy tu przyjąć założenie negatywne, że serwis może być niedostępny i ograniczyć się raczej do technologii zalecanych przez *World Wide Web Consortium* (W3C).

Jeżeli jednak Flash lub Java generują jedynie elementy ozdobne, które nie są elementami nawigacyjnymi, to można ich używać bez większych przeszkód, mając jednak świadomość, że nie będą widoczne dla wszystkich. Bardzo ważne jest jedynie, by takie osadzone elementy nie wymuszały odświeżania strony, ponieważ to bardzo utrudnia korzystanie z niej osobom niewidomym. Podobnie należy unikać szybko zmieniających się obrazów i migotania elementów, co może dezorientować osoby słabowidzące i utrudniać im korzystanie z oprogramowania powiększającego

Wskazówki

1. Aktywne elementy graficzne muszą posiadać opis alternatywny opisujący ich działanie lub cel, do którego prowadzą.
2. Grafika umieszczana na stronie powinna zawierać opis alternatywny, a w wypadku bardziej skomplikowanych opisów – długą deskrypcję.
3. Grafika ornamentowa, która nie niesie żadnej informacji powinna mieć pusty opis alternatywny.
4. Zasoby multimedialne powinny być zaopatrzone w tekstową deskrypcję warstw wizualnej i dźwiękowej.
5. Osadzanie na stronie technologii innych, niż zalecane przez W3C powinno być realizowane z największą ostrożnością.

Formularze

Tworzenie dostępnych formularzy jest stosunkowo proste, o ile zachowa się pewne podstawowe zasady – w tym przede wszystkim poprawność składniową. Wszystkie elementy formularza powinny być zamknięte w parze znaczników FORM i mieć własną, unikalną nazwę. W wypadku stosowania przycisków graficznych należy stosować zalecenia związane z grafiką opisane wcześniej. Poniżej można znaleźć bardziej szczegółowe zasady udostępniania formularzy z wykorzystaniem znaczników języka HTML.

Informacja kontekstowa

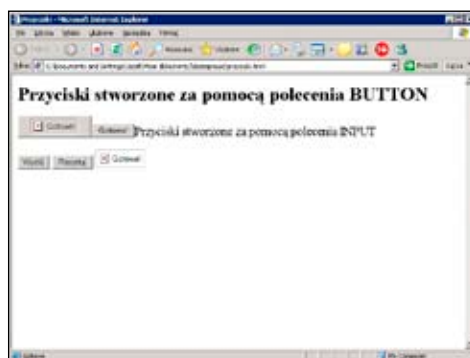
Na formularze składają się kontrolki kilku rodzajów:

- przyciski – tworzone za pomocą znacznika BUTTON lub INPUT;
- pola edycyjne – tworzone za pomocą polecenia INPUT (teksty jednowierszowe) i TEXTAREA (duże pola edycyjne);
- pola wyboru typu radio i *checkbox* – tworzone za pomocą polecenia INPUT;
- listy wyboru – tworzone za pomocą poleceń SELECT, OPTGROUP i OPTION;
- pola wyboru pliku.

Przyciski

Zadaniem przycisków w formularzach jest sterowanie działaniem danego formularza, na przykład przesłanie informacji do serwera, zaakceptowanie operacji lub uaktualnienie danych. Zazwyczaj mają formę prostokątów z napisem wewnątrz, choć często napis zastępowany jest grafiką. Na poniższym przykładzie pokazano dwa sposoby tworzenia przycisków w formularzach: za pomocą poleceń BUTTON i INPUT:

```
<form>
<button name="wyslij" value="Gotowe!"
type="submit"></button>
<button type="submit">Gotowe!</button>
<input type="submit" value="Wyślij">
<input type="reset" value="Resetuj">
<input type="image" src="gotowe.gif" alt="Gotowe!">
</form>
```



Fot. 5. Różne rodzaje przycisków.

Pierwszy przycisk w przykładzie został stworzony za pomocą polecenia `BUTTON` z wykorzystaniem grafiki. W takim wypadku dodatkową informację kontekstową o znaczeniu przycisku umieszcza się w atrybucie `ALT`. Jest to standardowa technika udostępniania grafiki opisana wcześniej (str. 19). Podobna sytuacja występuje w wypadku ostatniego przycisku stworzonego za pomocą polecenia `INPUT`, choć składnia jest nieco odmienna, ponieważ typ `IMAGE` polecenia `INPUT` oferuje możliwość umieszczenia odpowiednich atrybutów bezpośrednio.

W wypadku przycisków „Wyślij” i „Resetuj” informacja o ich działaniu, a jednocześnie napis na przycisku, umieszczane są w atrybucie `VALUE` i są dostępne bezpośrednio. Jak widać, przyciski są dość proste do udostępnienia, ponieważ w ich składni zaimplementowano informację kontekstową.

Pola edycyjne

Język HTML oferuje dwa rodzaje pól edycyjnych: jednoliniowe, które są tworzone za pomocą polecenia `INPUT`, oraz wieloliniowe, tworzone za pomocą polecenia `TEXTAREA`. Żadne z nich nie ma wewnętrznie zaimplementowanych informacji kontekstowych i dlatego celowe jest użycie polecenia `LABEL` do powiązania pól edycyjnych z ich etykietami tekstowymi.

Polecenie `LABEL` może być używane na dwa sposoby: przez otoczenie kontrolki wraz z jej opisem tekstowym (w poniższym przykładzie jest tak w wypadku pola wyboru na dole), a gdy nie jest to możliwe – przez określenie za pomocą atrybutu `FOR`, do jakiej kontrolki się odnosi dana etykieta tekstowa. Kontrolka jest identyfikowana za pomocą atrybutu `ID`, który musi mieć taką samą wartość, jak atrybut `FOR` odpowiadającej mu etykiety. Zaprezentowane zostało to na przykładzie pól edycyjnych. Ponieważ etykieta i odpowiadające jej pole edycyjne znajdują się w różnych komórkach tabeli, zastosowanie bezpośredniego otaczania jest niemożliwe.

```

<form>
<table>
  <tr>
<td><label for="imie">imie: </label></td>
<td><label for="nazwisko">nazwisko: </label></td>
<td><label for="email">e-mail: </label></td>
</tr>
<tr>
<td><input type="text" name="imie" id="imie"></td>
<td><input type="text" name="nazwisko" id="nazwisko"></td>
<td><input type="text" name="email" id="email"></td>
</tr>
<tr>
<td colspan="3"><label><input type="checkbox"
name="biuletyn" value="tak" id="biuletyn">zaznacz to pole,
aby otrzymywać biuletyn informacyjny</label></td>
</tr>
</table>
</form>

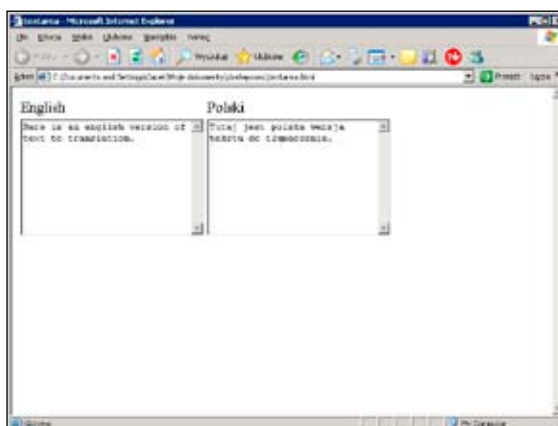
```

Polecenie TEXTAREA pozwala na wprowadzanie długiego tekstu w układzie zbliżonym do prostych edytorów tekstu. Poniżej przykład dwóch pól typu TEXTAREA umieszczonych w tabeli, której nagłówki są jednocześnie etykietami pól edycyjnych:

```

<form>
<table>
  <tr><td><label for="en">English</label></td>
<td><label for="pl">Polski</label></td></tr>
<tr>
<td><textarea name="en" id="en">Here is an English
version of text to translation.</textarea></td>
<td><textarea name="pl" id="pl">Tutaj jest polska wersja
tekstu do tłumaczenia.</textarea></td>
</tr>
</table>
</form>

```



Fot. 6. Przykład zastosowania pól edycyjnych typu TEXTAREA.

Pola wyboru

Pola wyboru w formularzach służą do dokonywania prostych wyborów i występują w dwóch rodzajach: CHECKBOX i RADIO.

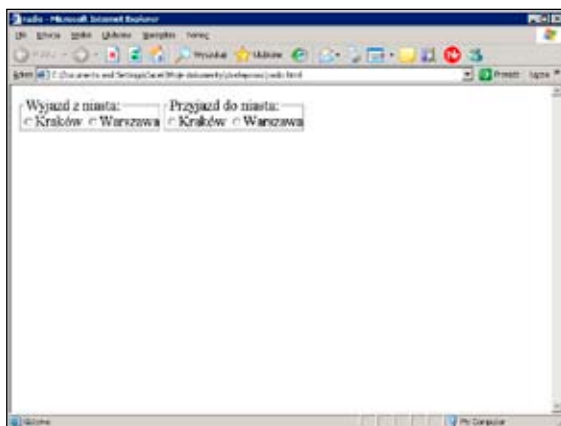
Pole typu CHECKBOX – służące do wybierania tylko pomiędzy dwoma stanami najczęściej ma postać kwadracika z „v” w środku (jeżeli jest zaznaczone) lub pustego (kiedy nie jest zaznaczone). W formularzach opis znajduje się najczęściej na prawo od kontrolki i można wtedy użyć znacznika LABEL do otoczenia kontrolki i opisu. Czasem jednak nie da się tego zrobić i wówczas należy użyć znacznika LABEL wraz z atrybutem FOR. Obie techniki ilustruje poniższy przykład:

```
<form>
<label><input type="checkbox" name="zgoda" id="zgoda">
Wyrażam zgodę na przetwarzanie moich danych osobowych
</label>
<hr>
<input type="checkbox" name="licencja" id="licencja"
value="zgoda">
<label for="licencja"> Zgadzam się na warunki licencji
</label>
</form>
```

Pole typu RADIO pozwala na dokonanie wyboru między co najmniej dwoma możliwościami. W obrębie tego wyboru może być wybrana tylko jedna opcja, a pozostałe są wyłączane. Kontrolka najczęściej jest wyświetlana w postaci kółek, przy czym aktualnie wybrana opcja oznaczona jest kropką wewnątrz kółka. Warto zauważyć, że użytkownik musi mieć dostęp do dwóch informacji:

czego dotyczy wybór i jaki jest wybór. Jeżeli kupuje bilet, nie wystarczy mu informacja, że wybiera pomiędzy Warszawą, a Krakowem, ale musi też wiedzieć, że dokonuje wyboru miasta, z którego będzie jechał. W tym celu – oprócz etykiet LABEL – należy zastosować grupowanie za pomocą polecenia FIELDSET, które opisane jest w dalszej części poradnika. Poniżej przykład ilustrujący realizację tego zadania:

```
<form>
<table>
<tr>
<td>
<fieldset>
<legend>Wyjazd z miasta:</legend>
<label><input type="radio" name="wyjazd"
value="krakow"> Kraków</label>
<label><input type="radio" name="wyjazd"
value="warszawa">Warszawa</label>
</fieldset></td>
<td><fieldset>
<legend>Przyjazd do miasta:</legend>
<label><input type="radio" name="przyjazd"
value="krakow"> Kraków</label>
<label><input type="radio" name="przyjazd"
value="warszawa">Warszawa</label>
</fieldset></td>
</tr>
</table>
```



Fot. 7. Dwie grupy przełączników typu RADIO.

Wykorzystanie polecenia LABEL w wypadku pól wyboru ma jeszcze jedną zaletę, którą jest znaczące **powiększenie powierzchni aktywnej kontrolki**. Użytkownik nie musi już wycelować kursorem dokładnie w kwadracik pola *checkbox* lub w kółko polecenia radio, lecz wystarczy kliknąć na napis związany z danym polem. Pozwala to na znacznie szybszą i wygodniejszą pracę wszystkim użytkownikom.

Listy wyboru

Jeżeli do wyboru jest znacząco więcej możliwości, zamiast przełącznika typu RADIO stosuje się listy wyboru tworzone za pomocą poleceń SELECT i OPTION. Mają one jeszcze jedną zaletę, którą jest możliwość wybrania kilku opcji zamiast jednej. Listy wyświetlane są na dwa sposoby: jako lista rozwijana lub prostokąt z wieloma pozycjami do wyboru. Pierwszy sposób jest odpowiedniejszy, gdy dajemy użytkownikowi możliwość pojedynczego wyboru, a drugi, gdy wybieramy kilka pozycji.

Poniższy przykład ilustruje sposób tworzenia dostępnej listy, która jest powiązana z etykietą za pomocą znacznika LABEL. Dzięki temu użytkownik ma pewność, czego dotyczy wybór. Warto zwrócić także uwagę, że zastosowano tu także grupowanie opcji za pomocą polecenia OPTGROUP. Systemy operacyjne podzielono na dwie grupy: Windows i Linux. Polecenie zawiera obowiązkowy atrybut LABEL, który jest etykietą grupy. Polecenia OPTION mogą zawierać także atrybut LABEL, który może być skróconą etykietą pełnej nazwy opcji, jednak jest to w tym wypadku opcjonalne.

```
<form>
<table><tr>
<td><label for="os">Wybierz system operacyjny</label></td>
<td><select name="os" id="os" size="12">
<optgroup label="Windows">
<option>Windows 95</option>
<option>Windows 98</option>
<option>Windows 2000</option>
<option>Windows XP</option>
<option label="Vista">Windows Vista</option>
</optgroup>
<optgroup label="Linux">
<option>Debian</option>
<option>RedHat</option>
<option>Ubuntu</option>
<option>Mandriva</option>
<option>Fedora</option>
```

```

</optgroup>
</select></td></tr></table>
</form>

```



Fot. 8. Użycie polecenia `SELECT` i grupowania opcji wyboru.

Pole wyboru pliku

Do rozważenia pozostała ostatnia kontrolka – pole wyboru pliku. Dzięki niej użytkownik może na swoim komputerze odnaleźć plik i przesłać go na serwer w celu dalszej obróbki. Kontrolka ta jest renderowana w sposób całkowicie automatyczny, co oznacza jej dostępność. Jak zwykle pozostaje problem powiązania kontrolki z jej etykietą, co jest realizowane – podobnie jak we wcześniej opisanych przykładach – za pomocą znacznika `LABEL`. W poniższym przykładzie zaprezentowany jest ten mechanizm:

```

<form>
<input type="file" id="dokument">
<label for="dokument">Wybierz dokument do wysłania</label>
</form>

```

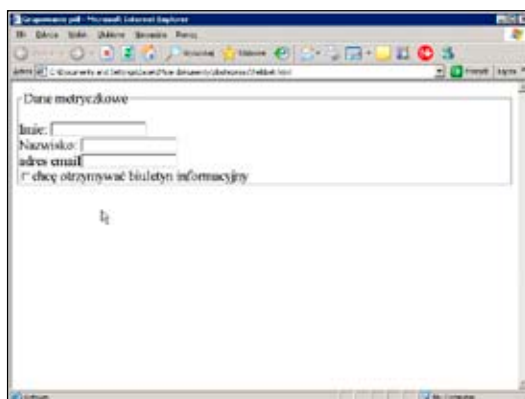
Grupowanie elementów

Bardziej rozbudowane formularze składają się zazwyczaj z kilku bloków tematycznych, które warto wyodrębnić dla zachowania odpowiedniej struktury. Do tego celu są stosowane dwa znaczniki:

- `FIELDSET` – służy do zamykania zestawu kontrolki w jeden blok;
- `LEGEND` – służy do nadawania tytułu blokowi.

Poniżej przedstawiono przykład zastosowania tego rozwiązania:

```
<form>
<fieldset>
<legend>Dane metryczkowe </legend>
<br><label>Imię: <input type="text" name="imie"></label>
<br><label>Nazwisko: <input type="text" name="nazwisko"></label>
<br><label>adres email<input type="text" name="email"></label>
<br><label><input type="checkbox" name="biuletyn"
value="tak">chcę otrzymywać biuletyn informacyjny</label>
</fieldset>
</form>
```



Fot. 9. Przykład grupowania kontroltek za pomocą polecenia FIELDSET.

W ten sposób powstał blok pól opisujących imię, nazwisko, adres e-mail oraz pole wyboru, które razem tworzą dane metryczkowe. Takich bloków można w formularzu stworzyć wiele i każdy będzie miał swoją nazwę. Znacznik LEGEND może mieć także opcjonalny atrybut ALIGN, czyli wyrównania napisu i może przyjmować wartości „TOP” (domyślna), „BOTTOM”, „LEFT” lub „RIGHT”.

Automatyzacja w formularzach

Inteligentne formularze są bardzo wygodną formą wprowadzania danych. Dzięki wykorzystaniu takich technologii, jak JavaScript można zaprojektować formularz, w którym lista dostępnych miast będzie się zmieniała w zależności od wybranego województwa albo wartość koszyka będzie się automatycznie aktualizować, gdy kupujący zmieni liczbę towarów. Jednak w takich formularzach można popełnić wiele błędów, które będą skutkować poważnymi ograniczeniami w dostępności.

Jednym z najczęściej spotykanych, a jednocześnie najuciążliwszych jest stosowanie rozwijanych list wyboru (*combobox*), w których wybranie danej pozycji powoduje automatyczne przejście do odpowiedniej strony. Ponieważ niektóre osoby posługują się klawiaturą zamiast myszki, są zmuszone do przechodzenia po kolei przez pozycje listy za pomocą klawiszy strzałek. Po każdej zmianie wyboru przeglądarka automatycznie wyświetla stronę odpowiadającą danej pozycji, a użytkownik jest zmuszony do powrotu i wybrania kolejnej pozycji z listy i tak aż do skutku. Zamiast takiego rozwiązania lepiej jest stworzyć taką samą listę i dodatkowy przycisk na formularzu z poleceniem przejścia.

Podobny problem występuje w wypadku formularzy, w których kolejne opcje są uzależnione od wybranych wcześniej. Takie formularze mają zaimplementowany mechanizm włączania i wyłączania odpowiednich kontroltek w zależności od stanu innych kontroltek. Takie mechanizmy są dość ryzykowne, więc lepszym rozwiązaniem wydaje się stosowanie tzw. wizzardów, w których wyświetlane są kolejne ekrany z polami formularzy w zależności od wyborów dokonanych na wcześniejszych.

Zabezpieczenia

Z formularzami bardzo często jest związany problem bezpieczeństwa – zarówno dla właściciela serwisu, jak i użytkownika. Formularze są zazwyczaj interfejsem do pewnych usług oferowanych w serwisie, na przykład poczty, konta bankowego czy też sklepu internetowego. Zapewnienie odpowiedniego poziomu bezpieczeństwa jest bardzo istotne, ale nie wolno przy tym zapomnieć o dostępności tego zabezpieczenia dla użytkowników.

Ze względu na bezpieczeństwo serwisów internetowych, nagminnie stosowane są zabezpieczenia w postaci tokenów graficznych. Po wypełnieniu formularza rejestracyjnego, na przykład konta poczty elektronicznej, użytkownikowi wyświetla się obrazek, na którym znajduje się kod do wpisania w odpowiednie pole w celu ostatecznej akceptacji. Takie zabezpieczenie zapobiega masowemu tworzeniu kont przez aplikacje internetowe, ponieważ wymaga odczytania kodu przez człowieka. W wypadku takiego zabezpieczenia nie można zastosować technik udostępniających grafikę, ponieważ rozwiązanie stałoby się bezużyteczne. Niektóre serwisy internetowe dostrzegły ten problem i próbowały go rozwiązać w ten sposób, że zamiast tokena graficznego użytkownik może otrzymać token dźwiękowy, czyli ten sam kod w postaci nagrania. Jednak z doświadczeń niewidomych użytkowników wynika, że to rozwiązanie także nie jest doskonałe, ponieważ nagranie jest celowo zniekształcane i trudne do zrozumienia. Pozostaje także problem osób głuchoniewidomych, dla których obie formy są w równym stopniu niedostępne.

Projektując systemy zabezpieczeń warto przemyśleć, czy nie są one zbyt rozbudowane. W jednym z banków elektronicznych wprowadzono token graficzny, na którym wyświetlany był numer kodu, który należy przepisać z karty kodów jednorazowych. Z kolei jedna z firm telekomunikacyjnych, oferując dostęp do konta użytkownika wymaga – oprócz loginu i hasła – jednorazowego kodu sesji wysyłanego poprzez SMS. W obu przypadkach zabezpieczenia wydają się niecelowe i utrudniają dostęp do usług, nie zapewniając dodatkowego bezpieczeństwa.

Programiści powinni pamiętać także o tym, by uprzedzić użytkownika o konieczności potwierdzenia operacji kodem z tokena graficznego lub w innej formie, która może być niedostępna. Można się spotkać bowiem z sytuacją, gdy po długim i żmudnym wypełnianiu danych w formularzu pojawia się prośba o przepisanie kodu z obrazka, a osoba niewidoma akurat nie ma kogo poprosić o pomoc. Poziom irytacji jest wówczas tak duży, że może całkowicie zniechęcić do korzystania z serwisu internetowego.

Jeżeli wprowadzenie zabezpieczeń specjalnych (tokeny graficzne, dźwiękowe, SMS, elektroniczne) jest niezbędne, to należy zapewnić także indywidualną ścieżkę dla użytkownika. Jedną z opcji autoryzacyjnych może być na przykład poproszenie administratora serwisu o dokonanie autoryzacji za użytkownika. Nie jest to rozwiązanie idealne, ponieważ wymaga dodatkowego czasu i ogranicza samodzielność użytkownika, ale czasem jest jedynym możliwym do zastosowania.

Wskazówki

1. Każdy element formularza powinien mieć informację kontekstową w postaci etykiety ściśle związanej z danym elementem.
2. Grupy kontrolek formularza należy łączyć w sposób logiczny.
3. Formularz musi być skonstruowany w taki sposób, by użytkownik mógł go wypełnić za pomocą klawiatury, bez użycia myszy.
4. Użytkownik powinien mieć zawsze możliwość sprawdzenia wszystkich podanych informacji przed ostatecznym zatwierdzeniem.
5. Zastosowane techniki zabezpieczeń powinny przewidywać korzystanie z nich przez osoby niepełnosprawne.

System nawigacji

Sprawne poruszanie się po rozbudowanych serwisach internetowych w znacznej mierze zależy od prawidłowego zaprojektowania systemu nawigacji. Temat ten jest bardzo obszerny i wiąże się bardziej z zasadami użyteczności niż dostępności serwisów internetowych, jednak poniżej przedstawiono kilka praktycznych rad, które pozwolą na podniesienie poziomu dostępności oferowanych usług.

Interfejs użytkownika

System nawigacyjny powinien być spójny w obrębie całego serwisu internetowego. Użytkownik powinien mieć możliwość łatwego zapamiętania położenia poszczególnych menu i usług, a więc powinny mieć one stałe miejsce w interfejsie. System nawigacyjny można także wydzielić przez zastosowanie nieco innej kolorystyki tła i łączy, co pozwoli na wyraźne oddzielenie go od treści głównej.

Dobrym rozwiązaniem jest nadawanie nazw poszczególnym blokom systemu nawigacyjnego, umieszczając je w znacznikach nagłówkowych (na przykład H2), co znacząco ułatwia pracę osobom niewidomym. Należy przy tym pamiętać o prawidłowym zagnieżdżaniu nagłówków, aby zachować strukturę dokumentu. Takie rozwiązanie jest ułatwieniem także dla osób widzących, które mają przejrzystą informację o tym, czego dotyczy dany blok nawigacyjny. Opisywana metoda jest coraz częściej wykorzystywana w różnych serwisach internetowych i rozwiązaniach typu CMS.

Skróty klawiaturowe

Interfejs serwisu internetowego można dodatkowo wyposażać w system skrótów klawiaturowych analogicznych do interfejsów aplikacji. Umieszczenie atrybutu ACCESSKEY przy niektórych znacznikach pozwala na szybkie odszukiwanie linków lub pól formularzy. Skróty klawiaturowe można skojarzyć z łączami, przyciskami, polami edycyjnymi i wyborami oraz grupami kontrolkami. Wartością atrybutu jest pojedynczy znak, jak w poniższym przykładzie:

```
<a href="mainpage.html" accesskey="h">Strona główna</a>
```

gdzie skrót **Alt+H** przenosi użytkownika do łącza do strony głównej. Zachowanie się elementów oznaczonych skrótami może być różne w zależności

od rodzaju elementu. W wypadku pól edycyjnych przenosi kursor do danego pola, a w wypadku pól wyboru zmienia ich wartość.

Język interfejsu i treści

Jednym z najtrudniejszych problemów w tworzeniu dostępnych serwisów jest rodzaj używanego języka. Z Internetu korzystają nie tylko ludzie młodzi i wykształceni, ale także osoby starsze i o obniżonej normie intelektualnej. Dlatego język stosowany w serwisie powinien być dostosowany raczej do tej słabszej części społeczności sieciowej. W pracy nad dostosowaniem serwisu warto zadbać o kilka elementów:

- tekst powinien być pisany w sposób możliwie prosty;
- wyrażenia trudne lub obcojęzyczne powinny być wyjaśniane;
- serwis powinien być zaopatrzony w pomoc kontekstową.

Łączy

Łączy, nazywane również linkiem, jest jednym z najważniejszych elementów na stronie internetowej. Dzięki łączom strony są hipertekstowe i pozwalają na wygodne poruszanie się w globalnej sieci. Dlatego tak ważne jest tworzenie ich w sposób możliwie dostępny.

Jeżeli łączy jest stworzone za pomocą obrazka (łączy graficzne), to należy stosować techniki opisane w rozdziale poświęconym multimediom, czyli przede wszystkim atrybut ALT. Warto jeszcze raz podkreślić, jak bardzo jest to istotne dla dostępności całego serwisu.

W wypadku łączy tekstowych warto pamiętać o kilku prostych zasadach:

- treść tekstu, na który klika użytkownik powinna wyraźnie określać, jaki będzie efekt kliknięcia, czyli dokąd go doprowadzi lub jaką akcję wywoła;
- tekst będący łączem powinien być wyraźnie wyróżniony spośród reszty tekstu na przykład przez podkreślenie, pogrubienie, inny kolor itp., przy czym należy pamiętać, by nie wyróżniać go wyłącznie kolorem;
- tekst łączy powinien być zrozumiały także wówczas, gdy zostanie wyrwany z kontekstu i dlatego należy unikać stosowania łączy typu „kliknij” lub „tutaj”, a także znaków nietekstowych, na przykład „>>”, ponieważ poza kontekstem nie mają one sensu.

W niektórych serwisach można znaleźć dodatkowe sposoby wyróżniania łączy przez ujmowanie ich w nawiasy kwadratowe [], co dodatkowo ułatwia odnajdowanie ich w treści strony. Innym prostym rozwiązaniem jest zmiana atrybutów (na przykład koloru) tekstu po najechaniu na łączy kursorem myszki. Każde tego typu rozwiązanie jest dobre, o ile nie stoi w sprzeczności z zasadami dostępności.

Ramki

W języku HTML istnieje możliwość łączenia kilku stron w jedną całość z wykorzystaniem ramek (FRAME). Mechanizm ten powstał w celu ułatwienia tworzenia systemu nawigacji w czasach, gdy strony internetowe były statyczne, a nie generowane dynamicznie. Było to stosunkowo proste rozwiązanie do umieszczania stałego menu nawigacyjnego i zmieniającej się treści. Wykorzystanie ramek ma jednak bardzo wiele wad i dlatego projektanci serwisów odchodzą od nich na rzecz systemów nawigacji opartych o warstwy. Jest to krok w dobrą stronę, ponieważ ramki często powodowały problemy z dostępem do treści.

Czasem jednak ktoś może zechcieć nadal korzystać z tego mechanizmu i wówczas powinien zadbać o przestrzeganie trzech prostych zasad:

- liczba ramek powinna być ograniczona do minimum;
- każda strona w ramce powinna mieć jednoznacznie identyfikującą nazwę;
- należy zadbać o dostęp do treści dla przeglądarek, które nie obsługują ramek.

Liczba ramek powinna być ograniczona do absolutnego minimum, co oznacza gruntowne przemyślenie przez autora, które ramki można zlikwidować lub włączyć do innych, by ich treść była wyświetlana wspólnie.

Jednoznacznie identyfikujące nazwy ramek są umieszczone wewnątrz znacznika TITLE w nagłówkach dokumentów wczytywanych do danej ramki. Dodatkowo w treści dokumentu można umieścić jego tytuł w znaczniku H1.

Pomimo, że ogromna większość przeglądarek internetowych potrafi obsługiwać mechanizm ramek, to wciąż istnieją takie, które tego nie potrafią. Użytkownikom tego typu oprogramowania należy zapewnić wersję alternatywną interfejsu nawigacyjnego i dostępu do treści i umieścić go w znaczniku NOFRAMES. Poważnym błędem jest pominięcie tej możliwości, szczególnie że nie jest to zbyt trudne. Biorąc jednak pod uwagę konieczność stworzenia bezramkowej alternatywy, warto pomyśleć, czy nie zrezygnować z ramek w ogóle jako z rozwiązania przestarzałego.

Dla zagubionych

Podstawowym celem tworzenia systemu nawigacji jest sprawienie, by użytkownik nie czuł się w nim zagubiony i potrafił odnaleźć potrzebną mu informację. Czasami jednak może się zdarzyć, że nie podoła wyzwaniu i zgubi się w gąszczu informacji. Należy zadbać o to, by zawsze mógł znaleźć koło ratunkowe w postaci łącza do strony głównej lub menu głównego. Ważne jest także, by w każdym momencie miał możliwość sprawdzenia, w którym miejscu serwisu się znajduje, co można zapewnić przez umieszczenie odpowiedniej informacji w belce tytułowej strony. Odpowiednie tytułowanie stron jest istotne także ze względów praktycznych, ponieważ tytuły stron są wykorzystywane przez przeglądarki do automatycznego nazywania plików zapisywanych na dyskach twardych oraz tworzeniu nazw ulubionych stron.

Na koniec warto wspomnieć o problemach, jakie może sprawić otwieranie stron w nowym oknie po kliknięciu łącza. Osoba słabowidząca może przeoczyć otwarcie nowego okna przeglądarki i poruszać się w nim z pełnym zaufaniem. Tymczasem takie nowe okno nie przejmuje historii odwiedzanych stron z poprzednio otwartej strony i nie pozwala na wygodny powrót do poprzedniego stanu. Jeżeli więc po kliknięciu łącza strona ma się otworzyć w nowym oknie, to należy o tym uprzedzić użytkownika odpowiednią informacją umieszczoną obok łącza.

Wskazówki

1. Interfejs użytkownika powinien mieć spójny wygląd w obrębie serwisu.
2. Język stosowany w serwisie powinien być w miarę możliwości prosty, a trudniejsze wyrażenia powinny być objaśniane.
3. Łącze powinno w sposób jednoznaczny informować dokąd prowadzi lub jaką akcję wykona.
4. Łącze powinno mieć sens poza kontekstem i powinno być wyraźnie wyróżnione w treści strony.
5. Ramki powinny być wykorzystywane z ostrożnością i umiarem, a każda z nich powinna mieć jasno zdefiniowany tytuł.

Na zakończenie

Temat dostępności jest niezwykle rozległy i poradnik nie jest w stanie go wyczerpać. Powstająca od pewnego czasu specyfikacja *Web Content Accessibility Guidelines 2.0* ma już objętość ponad 350 stron, a jeszcze wciąż nie jest ukończona (stan na październik 2007 roku.) Dlatego gorąco zachęcamy do dalszego poszukiwania informacji, które zostały jedynie zasygnalizowane w tej publikacji. W szczególności polecamy przeczytanie dokumentów przygotowanych przez *Web Accessibility Initiative* – grupę roboczą do spraw dostępności *World Wide Web Consortium*. Dokumenty te to przede wszystkim:

- I *WCAG (Web Content Accessibility Guidelines) 1.0*;
- I *WCAG 2.0* (wersja robocza);
- I *Checklist of Checkpoints for Web Content Accessibility Guidelines 1.0*;
- I dokumenty o charakterze technicznym i szkoleniowym – wszystkie można znaleźć pod adresem internetowym <http://www.w3.org/wai>, a w języku polskim kilka z nich na stronie <http://www.dwww.pl>.

Firmy promujące inne technologie (np. Adobe i Sun) także starają się rozwijać swoje produkty w obszarze dostępności. Technologia Flash może być niemal równie dostępna jak HTML pod warunkiem zapoznania się ze sposobami tworzenia zasobów w sposób dostępny. Bliższe informacje na ten temat można znaleźć na stronie <http://www.adobe.com/accessibility>. Znajduje się tam m.in. dokument pt. *Best Practices for Accessible Flash Design* autorstwa Boba Regana, opisujący sposoby udostępniania technologii Flash. Na tej samej stronie znajdują się informacje na temat przygotowywania dostępnych dokumentów w formacie PDF (*Portable Document Format*).

Firma Sun Microsystems, Inc. pracuje od wielu lat nad udostępnieniem stworzonego przez siebie środowiska Sun Java. W języku Java są pisane samodzielne aplikacje oraz aplety osadzone na stronach internetowych. Technologia jest wciąż w fazie rozwoju i oprogramowanie specjalistyczne dla osób niepełnosprawnych nie współpracuje z nią zbyt dobrze. Więcej na temat *Sun Microsystems Accessibility Program* można przeczytać pod adresem <http://www.sun.com/access/>.

W trakcie przygotowywania serwisów lub programowania systemów zarządzania treścią (CMS) mogą pojawić się kolejne wątpliwości i pytania. Zapraszamy więc do kontaktu z Fundacją Instytut Rozwoju Regionalnego przy ul. J. Wybickiego 3a, 31-261 Kraków, email: biuro@firr.org.pl. Wspólnie możemy stworzyć warunki do powstania nowego, bardziej dostępnego Internetu.

Dokument internetowy

1. Serwis internetowy powinien być dostępny dla każdego użytkownika.
2. Dodatkowe wersje dla specyficznych grup użytkowników powinny być tworzone tylko w ostateczności.
3. Należy przestrzegać standardów i specyfikacji technicznych, w szczególności opublikowanych przez *World Wide Web Consortium*.
4. Inne technologie (na przykład Flash, Java, PDF) powinny być stosowane z ostrożnością i tylko wówczas, gdy autor upewni się, że one także są dostępne.
5. Należy oddzielać strukturę i treść dokumentu od jego graficznej prezentacji.

Informacja tekstowa

1. Dokument powinien być skonstruowany w sposób logiczny z wykorzystaniem dostępnych znaczników opisu treści.
2. Tabele powinny być jednorodne, zachowywać sens po przekształceniu w sposób liniowy i zawierać dodatkową informację opisową.
3. Należy unikać nadmiernej różnorodności krojów czcionek i kolorystyki.
4. Warto wyraźnie oddzielić treść dokumentu od systemu nawigacyjnego.
5. Należy unikać stosowania jaskrawych kolorów oraz migających i poruszających się elementów.
6. Kontrast pomiędzy tłem a treścią powinien wynosić co najmniej 1:5.
7. Czcionki umieszczane na stronie powinny mieć wielkość co najmniej 10 punktów a konstrukcja strony powinna dawać możliwość ich powiększenia.
8. Najlepiej stosować kroje czcionek zaprojektowane do wyświetlania na ekranie monitora lub najbardziej znane.

Multimedia

1. Aktywne elementy graficzne muszą posiadać opis alternatywny opisujący ich działanie lub cel, do którego prowadzą.
2. Grafika umieszczana na stronie powinna zawierać opis alternatywny, a w wypadku bardziej skomplikowanych opisów – długą deskrypcję.
3. Grafika ornamentowa, która nie niesie żadnej informacji powinna mieć pusty opis alternatywny.
4. Zasoby multimedialne powinny być zaopatrzone w tekstową deskrypcję warstw wizualnej i dźwiękowej.
5. Osadzanie na stronie technologii innych, niż zalecane przez W3C powinno być realizowane z największą ostrożnością.

Formularze

1. Każdy element formularza powinien mieć informację kontekstową w postaci etykiety ściśle związanej z danym elementem.
2. Grupy kontrolek formularza należy łączyć w sposób logiczny.
3. Formularz musi być skonstruowany w taki sposób, by użytkownik mógł go wypełnić za pomocą klawiatury, bez użycia myszy.
4. Użytkownik powinien mieć zawsze możliwość sprawdzenia wszystkich podanych informacji przed ostatecznym zatwierdzeniem.
5. Zastosowane techniki zabezpieczeń powinny przewidywać korzystanie z nich przez osoby niepełnosprawne.

System nawigacji

1. Interfejs użytkownika powinien mieć spójny wygląd w obrębie serwisu.
2. Język stosowany w serwisie powinien być w miarę możliwości prosty, a trudniejsze wyrażenia powinny być objaśniane.
3. Łącze powinno w sposób jednoznaczny informować dokąd prowadzi lub jaką akcję wykona.
4. Łącze powinno mieć sens poza kontekstem i powinno być wyraźnie wyróżnione w treści strony.
5. Ramki powinny być wykorzystywane z ostrożnością i umiarem, a każda z nich powinna mieć jasno zdefiniowany tytuł.

Informacje o autorze



Jacek Zadrozny

Absolwent Wydziału Dziennikarstwa i Nauk Politycznych Uniwersytetu Warszawskiego. Od 2006 roku pracownik Fundacji Instytut Rozwoju Regionalnego, a wcześniej Polskiego Związku Niewidomych.

Fundacja Instytut Rozwoju Regionalnego (FiRR) jest organizacją pozarządową typu non-profit działającą od 26 sierpnia 2003 r. w dziedzinach nauki, transferu innowacji, przedsiębiorczości oraz kompleksowego i wielopoziomowego wspierania osób najbardziej dyskryminowanych w społeczeństwie. Podstawową misją Fundacji jest zrównoważony rozwój przedsiębiorczości, transfer innowacyjnych rozwiązań, a także wspieranie przedsiębiorstw i innych podmiotów w zdobywaniu i wykorzystywaniu nowoczesnych technologii. Współpracuje z Ministerstwem Nauki i Informatyzacji, Polską Agencją Rozwoju Przedsiębiorczości oraz władzami lokalnymi województwa małopolskiego, a w ostatnim okresie szczególnie intensywnie kooperuje z licznymi organizacjami działającymi na rzecz osób niepełnosprawnych, w szczególności niewidomych. Od lutego 2007 Fundacja posiada status Organizacji Pożytku Publicznego.

Projekt realizowany przez Partnerstwo na Rzecz Zwiększenia Dostępności Rynku Pracy dla Osób Niewidomych, w skład którego wchodzi – oprócz Fundacji – także Polski Związek Niewidomych, Akademia Górniczo – Hutnicza, Fundacja Na Rzecz Osób Niewidomych i Słabowidzących, Instytut Badań Marketingowych i Społecznych VRG Strategia oraz Agencja Handi-Soft współfinansowany jest ze środków Europejskiego Funduszu Społecznego w ramach Inicjatywy EQUAL. Jego głównym celem jest kompleksowa pomoc ludziom z dysfunkcją wzroku w znalezieniu i utrzymaniu pracy bądź podjęciu i ukończeniu studiów wyższych. Wzięcie udziału w projekcie dało szansę 60 osobom w wieku 16-35 lat z terenu województwa małopolskiego na zwiększenie swoich kompetencji i umiejętności.