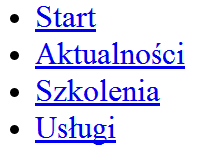
Temat: Menu pionowe i poziome w języku CSS z wykorzystaniem list.

Zaprojektowane na stronie internetowej menu umożliwia nawigację w jej obrębie. Paski nawigacyjne buduje się najczęściej na podstawie standardowej listy nieuporządkowanej. Lista jest zbiorem występujących po sobie pozycji; podobnie budowane jest menu. Za pomocą języka CSS można utworzoną listę zamienić w atrakcyjne menu w postaci pasków nawigacyjnych.

**Lista menu**

Budowanie menu zaczynamy od utworzenia listy w języku HTML. Lista taka powinna zawierać linki do odpowiednich miejsc wywoływanych przez menu.

**Przykład 2.90**



**Ćwiczenie 20.**

Utwórz listę z hiperłączami pokazaną w Przykładzie 2.90

Po zdefiniowaniu zawartości menu można przystąpić do jego formatowania z użyciem stylów CSS. Dołączenie deklaracji stylów pozwoli na stworzenie elastycznego i funk­cjonalnego menu nawigacyjnego. Sam wygląd menu zostanie zmieniony tylko przez dołączenie arkusza stylów z odpowiednimi deklaracjami.

Dobrym rozwiązaniem jest zdefiniowanie dla znacznika <ul>identyfikatora lub klasy, aby w wyniku formatowania tworzonego menu nie doprowadzić do przeformatowania zwykłych list, które mogą się znaleźć na stronie. Pozwoli to również na umieszczanie na stronie kilku różnie wyglądających zestawów menu.

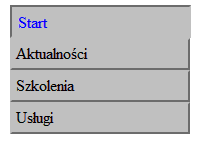
**Menu pionowe — deklaracje w arkuszu stylów**

Zanim przystąpimy do definiowania menu, z przygotowanej listy trzeba usunąć punk-tory oraz wyzerować jej marginesy. Należy również zadeklarować wyświetlanie listy w bloku oraz określić szerokość tego bloku.

Tak zapisana deklaracja w arkuszu stylów jest podstawą do tworzenia pionowych pasków nawigacyjnych.

Następne deklaracje dotyczą znacznika <a>występującego w zdefiniowanej liście oraz znacznika <ul>.

Przykładowe menu pionowe zostało pokazane na rysunku

****

Rysunek 2.46. Menu pionowe (wygląd menu z kursorem najechanym na START).

**Ćwiczenie 21.**

Zmodyfikuj poprzednie ćwiczenie dopisując w stylu osadzonym właściwości, tak aby otrzymać menu z Rysunku 2.46.

Jeśli masz problem z rozpoznaniem właściwości, pomogą Tobie poniższe wskazówki:

Wyłącz punktor w selektorze listy.

Dopisz pseudoklasy odsyłaczy: każdy z elementów listy ma być wyświetlony w bloku (długość bloku: 168px).Deklaracje a: link oraz a: visited mają mieć także zniesione podkreślenie, szary kolor #c0c0c0, kolor tekstu czarny, margines wewnętrzny 4px i ramka 2px z ustawionym stylem ramki (przycisk). Deklaracja a: hover ma mieć zmieniony styl obramowania (ramka), kolor tekstu niebieski oraz ustawienia marginesu wewnętrznego na 6px 3px 3px6px.

**Menu poziome - deklaracje w arkuszu stylów**

Przy tworzeniu menu poziomego również korzystamy ze zdefiniowanej wcześniej listy nieuporządkowanej.

Domyślnie lista jest tworzona w układzie pionowym. Ułożenie pozycji listy obok siebie można uzyskać na dwa sposoby:

* wyświetlając kolejne elementy listy w linii (atrybut display: inline);
* ustawiając pływanie dla każdej pozycji listy (atrybut float).

Podobnie jak w przypadku menu pionowego, z przygotowanego menu należy usunąć punktory oraz wyzerować jego marginesy.

**Przykład 2.93**

ul, li { display: block; list-style: none; margin: 0; padding: 0; }

**Wyświetlanie kolejnych elementów listy w linii**

**Przykład 2.94**

ul li { display: inline; white-space: nowrap; }

ul a:link, ul a:visited { text-decoration: none;background-color: #c0c0c0;color: black; padding: 4px;border: 2px outset #c0c0c0; }

ul a:hover { border-style: inset; color: blue; padding: 6px 3px 3px 6px; }

Dodane polecenie white-space: nowrap; blokuje zawijanie tekstu w każdej pozycji menu (rysunek 2.47). Jeśli wszystkie pozycje menu nie zmieszczą się w jednej linii, po­dział nastąpi między elementami listy, a nie wewnątrz elementu.



Rysunek 2.47 Menu poziome – kolejne elementy w linii (wygląd menu z kursorem najechanym na START).

Metoda ta daje poprawne efekty w większości przeglądarek i nie wpływa na elementy znajdujące się poniżej menu. Nie można jednak za jej pomocą zdefiniować jednakowej szerokości pozycji menu.

**Ćwiczenie 22.**

Na podstawie pliku z poprzedniego ćwiczenia zaprojektuj menu poziome podobne do tego z rysunku 2.47.

Podpowiedź:

Zmodyfikuj styl usuwając szerokości długość bloków. Dopisz selektor potomka: ul li { display: inline; white-space: nowrap; }

**Float dla każdej pozycji listy**

**Przykład 2.95**

ul, li { float: left; }

ul a:link, ul a:visited {text-decoration: none;background-color: #c0c0c0;color: black;padding: 4px;border: 2px outset #c0c0c0; }

ul a:hover { border-style: inset; color: blue;padding: 6px 3px 3px 6px; }

Po zastosowaniu tej metody wynik w przeglą­darce Internet Explorer nie jest zadowalający (rysunek 2.48), natomiast inne przeglądar­ki wyświetlają menu prawidłowo (rysunek 2.48).



Rysunek 2.48 – Menu poziome w Google Chrome (pływanie)



Rysunek 2.49 – Menu poziome w IE

Wygląd menu w IE można poprawić, umiesz­czając elementy menu w blokach. Jeżeli ele­menty zostaną umieszczone w bloku, będzie można zdefiniować dla nich stałą szerokość (patrz przykład 2.96)

**Ćwiczenie 23.**

Zaprojektuj na stronie internetowej menu poziome podobne do tego z rysunku 2.48 drugą metodą zwaną PŁYWANIEM.

Podpowiedź:

Zastąp selektor ul li { display: inline; white-space: nowrap; } pływaniem.

**Przykład 2.96**

ul, ul li {display: block; list-style: none; margin: 0;padding: 0; }

ul li {float: left; margin-right: 10px; }

ul a:link, ul a:visited { display: block;position: relative; text-decoration: none; background-color: #c0c0c0; color: black; padding: 4px; border: 2px outset #c0c0c0; }

ul a:hover { border-style: inset; color: blue; padding: 6px 3px 3px 6px; }

Po umieszczeniu elementów menu w blokach wynik w przeglądarce Internet Explorer jest prawidłowy (rysunek 2.50).



Rysunek 2.50. Elementy menu umieszczone w blokach

Dodatkowo można określić stałą szerokość bloków i wyśrodkowanie tekstu menu (rysunek 2.51)



Rysunek 2.51. Elementy menu z określoną stałą szerokością

**Przykład 2.97**

ul a:link, ul a:visited { display: block; width: 100px;text-align: center; text-decoration: none; background-color: #c0c0c0; color: black; padding: 4px;border: 2px outset #c0c0c0; }

Umieszczenie elementów w blokach daje możliwość zdefiniowania jednakowej szerokości wszystkich pozycji menu oraz ustalenia odległości między pozycjami.

Mogą natomiast pojawić się problemy z prawidłowym wyświetlaniem w niektórych przeglądarkach, trzeba również pamiętać o przypisaniu właściwości *clear*następnemu elementowi strony w celu uniknięcia „przyklejenia" tego elementu do menu.

**Ćwiczenie 24.**

Zaprojektuj korzystając z poprzedniego pliku menu poziome podobne do tego z rysunku 2.51.

Wskazówki:

Dopisz do ul li {float: left; }margines zewnętrzny prawy: 10px.

Dopisz także do pseudoklas ul a:link, ul a:visited właściwości:

display: block; width: 100px; text-align: center;

Zmodyfikuj kolory i czcionki wg własnych preferencji.

**Menu zagnieżdżone**

Podobnie jak w przypadku menu pionowego lub poziomego, budując menu zagnieżdżone, należy zacząć od utworzenia listy. Lista powinna zawierać linki do odpowiednich miejsc wywoływanych przez menu.

**Przykład 2.98**

<ul id="blok">

<li><a href="aktualności.html">Aktualności</a></li>

<li><a href="szkolenia.html">Szkolenia</a>

<ul>

<li><a href="kursl.html">HTML</a></li>

<li><a href="kurs2.html">CSS</a></li>

<li><a href="kurs3 .html">JavaScript</a></li>

</ul>

</li>

<li><a href="usługi .html">Osługi</a></li>

</ul>

Po zdefiniowaniu zawartości menu można przystąpić do tworzenia menu zagnieżdżonego. Ponieważ zaproponowana lista ma podlistę tylko na drugiej pozycji, nasze menu będzie się rozwijało tylko na tej właśnie pozycji.

Podobnie jak w przypadku menu pionowego i poziomego, z przygotowanego menu trzeba usunąć punktory oraz wyzerować jego marginesy.

**Przykład 2.99**

ul, ul li {margin: 0; padding: 0;}

#blok li {list-style: none;}

W definicji stylów został zdefiniowany identyfikator #blok, który w treści dokumentu HTML został przypisany do znacznika listy <ul>.

Następnym etapem jest zdefiniowanie wyglądu menu.

**Przykład 2.100**

#blok, #blok ul {width: 140px;}

#blok li {padding-bottom: 1px; position: relative;}

#blokul { position: absolute; top: 0; left: 160px; padding-left: 1px; visibility: hidden; }

#blokli:hoverul {visibility: visible; width: auto; height: auto;}

Listom została nadana szerokość *14 0px,* a elementy listy zostały rozdzielone przerwą *(padding-bottom: 1px).* Zdefiniowano położenie względne dla listy głównej w sto­sunku do strony internetowej oraz położenie podlisty w stosunku do listy głównej.

Dla wewnętrznej listy *(#blok ul)* zdefiniowano pozycjonowanie absolutne w stosunku do listy głównej oraz jej ukrycie *(visibility: hidden;).*

Gdy ustawimy kursor myszy na elemencie menu, wyświetli się podmenu.

Pozostało jeszcze zdefiniować, jak ma wyglądać element listy, gdy ustawimy nad nim kursor myszy. Aby element menu głównego był nadal podświetlony, gdy zostanie wybrany element podmenu, należy odpowiednio wykorzystać zasady dziedziczenia.

**Przykład 2.101**

/\*definiuje wygląd każdego elementu menu\*/

#blok a, #blokli:hover li a {

display: block;

font: 12px/30px verdana, sans-serif;

text-decoration: none;

padding: 0 10px;

width: 140px;

color: black;

background-color: #c0c0cf;

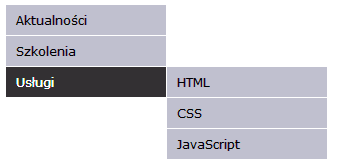
}

/\*definiuje wygląd elementu wskazanego kursorem myszy  
jeżeli będzie to element podmenu, element z menu głów­nego też pozostanie podświetlony.\*/

#blokli:hover a, #blokli:hoverli:hover a {

background-color: #333033;

color: white; }



Rysunek 2.52 Menu zagnieżdżone

**Ćwiczenie 25.**

Otwórz plik cwiczenie25.html, sprawdź jak wygląda w przeglądarce. Dopisz po trzy rozwinięcia do Aktualności, Szkolenia i Usługi podobnie jak na rysunku 2.52. Uwaga! Nie musisz modyfikować stylów.