

Utworzenie rozwijanego menu

Zadanie 1

Na podstawie przykładu omówionego poniżej proszę stworzyć **pliki html z indywidualnym** (innym niż w podanym przykładzie) **zestawem pozycji menu**: przesyłamy nauczycielowi plik strony głównej oraz podstrony, do których odsyłają pozycje menu, **a także plik css**. Plik css podłączamy do strony w sekcji head umieszczając znacznik:

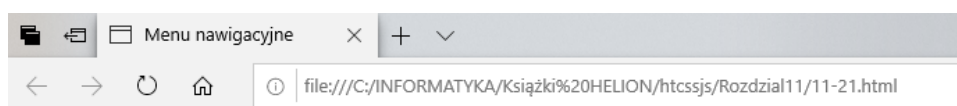
```
<LINK REL=stylesheet HREF="nazwa pliku css.css" TYPE="text/css">
```

Termin wykonania zadania : do czwartku 12.03.2020 godz. 23:59.

Brak przesłanego w terminie zadania skutkować będzie oceną niedostateczną.

Przykład do przeanalizowania:

Rozwijane menu to jedna z najczęściej używanych technik nawigacji w witrynach internetowych. Pozwala na zapewnienie szczegółowej nawigacji, nie wymagając przy tym zbyt dużej ilości miejsca na ekranie. Aby zaprezentować możliwości rozwijanego menu, na rysunku 11.12 pokazałem wygląd ukończonego menu, którego budowę się teraz zajmiemy. Wcześniej tego rodzaju rozwiązanie musiało opierać się na języku JavaScript, natomiast obecnie arkusze stylów CSS oferują na tyle potężne możliwości, że można je wykorzystać do budowy na przykład interfejsu opartego na rozwijanym menu. W omawianym tutaj przykładzie wykorzystałem wiele technik, które poznałeś już wcześniej.



Typowa witryna internetowa



Typowa treść witryny internetowej.

RYSUNEK 11.12. Menu nawigacyjne w postaci rozwijanego menu

Budowa paska menu zaczyna się od przygotowania zwykłej zagnieżdżonej listy HTML. Style wykorzystamy do konwersji tej listy na dynamiczny pasek nawigacyjny. Poniżej przedstawiłem kod źródłowy omawianej strony.

```
<!DOCTYPE html>
<html>
<head>
<title>Menu nawigacyjne</title>
<style type="text/css">
</style>
</head>
<body>
<h1>Typowa witryna internetowa</h1>
<nav id="navigationBar">
```

```

<ul>
<li><a href="/home.html">Strona główna</a></li>
<li><a href="/products.html">Produkty</a>
<ul>
<li><a href="/products/a.html">Wspaniały produkt</a></li>
<li><a href="/products/b.html">Kolejny wspaniały produkt</a></li>
<li><a href="/products/c.html">Nasz najlepszy produkt</a></li>
<li><a href="/products/d.html">Nasz tani produkt</a></li>
</ul>
</li>
<li><a href="/about.html">O firmie</a>
<ul>
<li><a href="/history.html">Historia</a></li>
<li><a href="/team.html">Zespół</a></li>
</ul>
</li>
<li><a href="/contact.html">Kontakt</a></li>
</ul>
</nav>
<p>Typowa treść witryny internetowej.</p>
</body>
</html>

```

Pierwszym krokiem na drodze prowadzącej do utworzenia menu jest poziome ułożenie nadrzędnego elementu listy wraz z odpowiednią ilością miejsca do ukrycia zagnieżdżonych list. Wspomniany element musi być pozycjonowany. Jak się okazuje, lista zewnętrzna musi być nie tylko pozycjonowana, ale również konieczne jest zastosowanie pozycjonowania bezwzględnego. Niestety, jak już wiesz, pozycjonowanie bezwzględne powoduje usunięcie elementu z przepływu normalnego. Jeżeli listę będziemy pozycjonować w taki sposób, zakłócimy przepływ normalny dla pozostałej części strony. Dlatego też zdecydowałem się na dodanie kolejnego elementu na stronie, który będzie mógł być pozycjonowany względnie. Tak więc pierwszym prawdziwym krokiem jest opakowanie listy znacznikiem `<nav>` z identyfikatorem - przykład możesz zobaczyć w poniższym fragmencie kodu.

```
<nav id="navigationBar"> ... listy menu ... </nav>
```

Skoro wszystko zostało przygotowane, możemy przystąpić do nadania stylu elementowi `<nav>` oraz zagnieżdżonym w nim listom. W trakcie pracy trzeba pamiętać o wymienionych poniżej kwestiach, aby otrzymać prawidłowo funkcjonujące rozwijane menu.

- Usunięcie znaków wypunktowania z list.
- Poziome ułożenie listy nadrzędnej.
- Podanie wielkości dla wszystkich list i ich elementów, aby tym samym ułatwić ich pozycjonowanie.
- Pozycjonowanie podmenu, aby było wyświetlane w odpowiednim miejscu.
- Domyślnie ukrywanie podmenu.
- Zastosowanie stylów dostosowujących wygląd menu do naszych potrzeb.
- Wyświetlenie podmenu po umieszczeniu kursora myszy nad elementem menu zawierającym podmenu.

Przechodzimy do stylów css wymaganych do prawidłowego działania budowanego menu. Zastosujemy kolejność coraz większej szczegółowości. Na początek nadajemy styl kontenerowi `<nav>`.

```

#navigationBar {
position: relative;
margin: 0;

```

```
padding: 0;
height: 30px;
width: 90%;
}
```

Te style powodują względne pozycjonowanie paska menu, aby umożliwić w nim zastosowanie pozycjonowania bezwzględnego. Wyłączyłem marginesy i dopełnienie, a także podałem wielkość paska nawigacyjnego. Kolejnym krokiem jest nadanie stylu liście najwyższego poziomu w pasku nawigacyjnym.

```
#navigationBar ul {
margin: 0;
padding: 0;
position: absolute;
height: 100%;
width: 100%;
list-style-type: none;
background-color: #eeeeee;
overflow: visible;
}
```

Także w tym przypadku pozbyłem się marginesu i dopełnienia. Jednak tym razem zastosowałem pozycjonowanie bezwzględne listy względem jej kontenera. Wysokość i szerokość zdefiniowałem na 100%, aby lista wypełniła cały kontener. Ponadto wyłączyłem wyświetlanie znaków wypunktowania. Zdefiniowałem kolor tła i zezwoliłem na wyświetlenie treści powodującej przepełnienie listy. Kolejnym krokiem jest nadanie stylu elementom listy w menu najwyższego poziomu. Zastosowałem selektor >, gwarantując tym samym, że style będą miały wpływ jedynie na elementy listy najwyższego poziomu, a nie elementy list potomnych.

```
#navigationBar ul > li {
display: block;
float: left;
height: auto; /* Szerokość elementów nagłówka menu. */
width: 120px;
}
```

Elementy są dosunięte do lewej strony, mają postać bloków, a także zdefiniowaną szerokość. Następnym krokiem jest nadanie stylu łączom na liście najwyższego poziomu.

```
#navigationBar li > a {
display: block;
padding: 0 0 0 10px;
height: 100%;
width: 109px;
border-right: 1px black solid;
line-height: 30px;
font-family: sans-serif;
font-size: 17px;
text-decoration: none;
background-color: #eeeeee;
color: #5d5636;
}
```

Zdefiniowałem wiele stylów dla łącza, aby zapewnić mu odpowiedni wygląd i wpasować do budowanego układu. Kolejnym krokiem jest rozpoczęcie pracy nad zagnieżdżonymi listami. Najpierw sama zagnieżdżona lista.

```
#navigationBar ul ul {
margin-top: 2px;
display: none;
position: static;
height: auto;
width: 160px;
border: 1px #666666 solid;
```

```
background-color: #dddddd;  
}
```

Właściwości display przypisałem wartość none, więc lista pozostaje ukryta po wczytaniu strony. Położenie zdefiniowałem jako static, co jest wartością domyślną. Ponadto dodałem obramowania i kolor tła. Teraz możemy przejść do nadania stylu elementom w podmenu.

```
#navigationBar ul ul li {  
position: relative;  
float: none;  
display: block;  
height: 28px;  
width: 100%;  
border: none;  
}
```

W omawianym przykładzie elementy listy mają przypisane pozycjonowanie względne, pływanie elementu zostało wyłączone, a same elementy mają określoną wielkość. Wiele zastosowanych tutaj stylów nadpisuje style dla elementów na liście nadrzędnej. Kolejnym krokiem jest zdefiniowanie stylów dla łącza na liście.

```
#navigationBar ul ul li > a {  
height: 100%;  
width: 96%;  
padding: 0 0 0 4%;  
line-height: 28px;  
background-color: transparent;  
border: none;  
color: #000000;  
font-size: 12px;  
font-style: normal;  
}
```

W tym przypadku zdefiniowaliśmy wysokość i szerokość łącza, dodaliśmy nieco dopełnienia oraz określiliśmy wielkość linii, aby łącze zostało prawidłowo umieszczone. Ponadto nieco zmniejszyłem wielkość czcionki, co ułatwia zmieszczenie łącza w przeznaczonym dla nich miejscu. Po zakończeniu pracy nad wyglądem łącza pozostało już tylko dodanie stylów dla sytuacji, gdy kursor myszy znajdzie się nad elementem listy. W ten sposób zapewniamy dynamiczność strony.

```
#navigationBar li:hover > a {  
color: #220000;  
background-color: #eeeabe;  
}
```

```
#navigationBar li:hover > ul {  
display: block;  
}
```

Kiedy kursor myszy znajdzie się na elementem listy nawigacyjnej, właściwość display listy potomnej tego elementu otrzymuje wartość block. Zmianie ulega kolor tła oraz łącza, aby wyglądało na wybrane. W ten sposób zakończyliśmy pracę nad listą. W efekcie otrzymaliśmy system dynamicznego menu zaimplementowany za pomocą wyłącznie arkuszy stylów CSS.

Na podstawie: Html, CSS i JavaScript dla każdego

Autor: Rafe Colburn

Wydawnictwo HELION 2017