

ZAAWANSOWANE TECHNIKI WWW (WFAIS.IF-C112)

(zajęcia 23.10.2013)

1) Tabele w html:

Jednym z podstawowych elementów na stronach do porządkowania i przechowywania informacji są tabele. Podstawowa deklaracja tabeli na stronie wygląda następująco:

```
<table>
  <tr>  <!-- Wiersz pierwszy -->
        <td> Komórka 1 </td> <td> komórka 2 </td>
  </tr>
  <tr>  <!-- Wiersz drugi -->
        <td> Komórka 1 </td> <td> komórka 2 </td>
  </tr>
</table>
```

W standardowej wersji jest tabela ma atrybut border ustawiony na 0. W celu jego zmiany należy ręcznie to zadeklarować: <table border="1">

Odstępy w komórkach można osiągnąć przez: <table cellpadding="10px">

Odstępny pomiędzy komórkami: <table cellspacing="10px">

Można również ustawić nagłówki w kolumnach przez dodanie dodatkowego wiersza na początku tabeli:

```
<tr>
  <th>nagłówek kolumny 1</th>      <th>Nagłówek kolumny 2</th>
</tr>
```

3) Kaskadowe arkusze stylów - CSS:

Pozwalają na oddzielenie warstwy formatu od warstwy danych (treści).

Stylizacja strony (witryny) odbywa się w osobnym pliku.

W arkuszach stylów stosujemy formułę:

```
selektor1, selektor2, ... { cecha1: wartość;
                           cecha2: wartość;
                           cecha3: wartość;
                           ...;
}
```

Selektor - jest to nazwa identyfikująca znacznik dla którego przeglądarka ma przypisać styl.

Cecha - Własność jaką chcemy określić dla danego selektora.

Wartość - ...

4) Deklaracja stylu CSS - trzy metody:

a) Style wbudowane (inline-styles) w danym znaczniku:

```
<h1 style="font-family: Arial;
          text-align: center;
```

```
        color: green;
        background: yellow;">
    To jeszcze jeden nagłówek 1-go stopnia ze stylem osadzonym.
</h1>
```

lub `<body style="background: blue;">` / kolorujemy tło sekcji body.

b) Style osadzone (embedded-styles) wewnątrz sekcji head:

```
<style rel="stylesheet" type="text/css">
    h1 {text-align: center;
        color: yellow;
        background: green;
        padding: 4px;}
    p.uwaga_12{font-size:16px;
        font-family:Arial;
        color:#804040;
        font-weight:bold;}
</style>
```

c) Deklaracja w pliku zewnętrznym - linked-style (najczęściej stosowana i najlepsza)

```
<head>
    <link rel="stylesheet" type="text/css" href="style1.css" />
</head>
```

5) Dziedziczenie:

Najistotniejszy z punktu widzenia dziedziczenia i stosowalności danej cechy-własności jest styl zadeklarowany "inline", w drugiej kolejności "osadzony, a na końcu "dołączony".

Dodatkowo można deklarować selektory tak aby zastosowanie danej własności miało miejsce tylko wtedy kiedy znajduje się wewnątrz innego selektora:

```
div p { text-align: center;}          // to dwie różne deklaracje
div, p { text-align: center;}        //
```

6) Klasy i identyfikatory selektorów w CSS:

a) Klasy stosujemy aby rozróżnić / nadać / tym samym selektorom występującym w danym dokumencie hipertekstowym różne cechy np. w witrynie często występują znaczniki `<h1>` ale dla zabawy chcemy aby w zależności od ważności danego paragrafu były pisane różnymi kolorami. CO WTEDY?? ----> Wtedy stosujemy klasy selektorów!! Ogólna deklaracja klasy selektora wygląda następująco:

```
selektor.klasa { cecha: wartość; } // klasy stowarzyszone z selektorem
```

aby odwołać się do konkretnej klasy w tekście to należy wybrać klasę selektora przy danym znaczniku:

```
<selektor class="klasa"> ... </selektor>
```

A więc w naszym przykładzie będzie:

```
<style type="text/css">
    h1 { font-family: Verdana; }
    h1.wazny { color: blue; }      // można deklarować HEXem
    h1.bwazny { color: red; }      // #FFFFFF - biały
    h1.bbwasny {color: green;}     // #000000 - czarny
</style>

<h1> Tekst standardowy pisany Verdana </h1>
<h1 class="wazny"> Tekst wazny </h1>
<h1 class="bwazny"> Tekst bardzo wazny </h1>
<h1 class="bbwasny"> Tekst bardzo bardzo wazny </h1>
```

W ten sposób można definiować klasy dla takich samych selektorów. Jak w tym przypadku jest z dziedziczeniem ???

Klasy nazwane nie przypisane do selektorów:

```
.klasa { cecha: wartość; } // klasy NIE-stowarzyszone z selektorem
```

Np:

```
<style type="text/css">
    .srodek { text-align: center;}
</style>

<p class="srodek"> tekst wysrodkowany za pomoca klasy srodek </p>
<h1 class="srodek"> Tekst wysrodkowany </h1>
```

b) Identyfikatory to w zasadzie identyczne obiekty jak klasy pozwalające na określenie cech dla danego selektora. Różnica jest taka że identyfikatory mogą być przypisane w danym dokumencie hipertekstowym tylko raz do jednego selektora. Ogólna deklaracja to:

```
selektor#identyfikator { cecha: wartość; } // identyfikator z selektorem
```

Nadanie cech dla danego selektora następuje przez atrybut `id=""`:

```
<selektor id=""> .. </selektor>
```

Przykład tym razem z h2:

```
<style type="text/css">
    h2 { font-family: Times; }
    h2#wazny {color: #ABCDEF; }
    h2#bwazny {color: #793732; }
    h2#bbwasny {color: #91AF72; }
</style>

<h2> Tesk pisny h2 </h2>
<h2 id="wazny"> Teskt wazny </h2>
```

```
<h2 id="bwazny"> Teskt b.wazny </h2>
<h2 id="bbwazny"> Teskt b.b.wazny </h2>
```

7) Stylizowanie odsyłaczy w zależności od danego stanu odsyłacza.

Możliwe statusy hiperłączy:

link – podstawowy odsyłacz;

hover – odsyłacz, nad którym zatrzymano kursor myszy;

focus – odsyłacz z tzw. fokusem, czyli miejscem zaznaczonym przy poruszaniu się po dokumencie przy użyciu klawisza tabulacji. Naciśnięcie klawisza Enter – spowoduje taką samą reakcję przeglądarki, jakby dany odsyłacz został kliknięty;

active – aktywny odsyłacz (kliknięty);

visited – odsyłacz, który prowadzi do wcześniej już odwiedzonego dokumentu HTML, tj. takiego, który znajduje się w historii przeglądarki.

Możemy definiować style dla odsyłaczy w zależności od przyjętego aktualnego stanu odsyłacza:

```
<style type="text/css">
  a:link {color: red;}
  a:visited {color: #9AB676;}
  a:active {color: green;}
  a:hover {color: yellow;}
</style>
```

```
<a href="http://www.onet.pl">ONET.PL</a>
```

Można również stosować klasy:

```
<style type="text/css">
  a:link.uj {color: #00FF00;}
  a:visited.uj {color: #800000;}
  a:active.uj {color: #00FFFF;}
  a:hover.uj {color: #808000;}
</style>
<a class="uj" href="http://www.uj.edu.pl">UJ.EDU.PL</a>
```

8) Formatowanie tekstu:

Typy czcionek: font-family: serif, Arial, Helvetica, etc...

Rozmiary czcionek:

font-size: xx-small|x-small|small|medium|large|x-large|xx-large|larger|smaller|n_pt|n_px|n%

font-size: 50px;

Parametry formatowania tekstu i bloków dopuszczalne w arkuszach stylów		
Charakterystyka	Znaczenie	Opis i możliwe wartości
font-family	Rodzina czcionek	Kroje czcionek dostępne w przeglądarkach. Jeśli dany krój nie

	k	występuje, używana jest czcionka domyślna.
font-size	Wielkość czcionki	Dowolna wartość w punktach [pt], calach [in], centymetrach [cm] albo w pikselach [px]. Dodatkowo dopuszczalne są wartości: larger, smaller, xx-small, x-small, small, medium, large, x-large, xx-large. Dopuszczalna jest również wartość procentowa względem rozmiaru czcionki macierzystej [%].
font-weight	Grubość czcionki	Dopuszczalne wartości: normal, bold, bolder, lighter.
font-style	Styl czcionki	Dopuszczalne wartości: normal, italic (kursywa), oblique (skośna).
font-variant	Odmiana czcionki	Dopuszczalne wartości: normal, small-caps.
color	Kolor czcionki.	Jeden z predefiniowanych kolorów lub kod heksadecymalny RGB.
background-attachment	Połączenie z tłem	Obraz przewija się razem z zawartością: scroll; obraz tła pozostaje nieruchomy: fixed.
background-color	Kolor tła	Przezroczyste tło: transparent; jeden z predefiniowanych kolorów lub kod heksadecymalny RGB.
background-image	Obrazek tła	Żaden obrazek: none; adres URL obrazka.
background-repeat	Powtarzanie tła	Powtarzanie tła tylko w kierunku poziomym: repeat-x; powtarzanie tła tylko w kierunku pionowym: repeat-y; powtarzanie tła w obu kierunkach: repeat; nie powtarza tła: no-repeat.
border-color	Kolor krawędzi	Jeden z predefiniowanych kolorów lub kod heksadecymalny RGB.
border-style	Styl krawędzi	żadna krawędź: none; linia przerywana: dashed; linia kropkowana: dotted; linia ciągła: solid; linia podwójna: double; wyłobione: groove; wypukłość:

		ridge; wkładka: inset; wypustka: outset.
border-bottom-width	Szerokość dolnej krawędzi	Cienka: thin; średnia: medium; dowolna liczba punktów [pt], cali [in], centymetrów [cm] albo pikseli [px].
border-left-width	Szerokość lewej krawędzi	Cienka: thin; średnia: medium; dowolna liczba punktów [pt], cali [in], centymetrów [cm] albo pikseli [px].
border-right-width	Szerokość prawej krawędzi	Cienka: thin; średnia: medium; dowolna liczba punktów [pt], cali [in], centymetrów [cm] albo pikseli [px].
bottom-top-width	Szerokość górnej krawędzi	Cienka: thin; średnia: medium; dowolna liczba punktów [pt], cali [in], centymetrów [cm] albo pikseli [px].
float	Przesunięcie	Przesuwa ustawianą zawartość do w lewo: left, w prawo: right albo nie przesuwa: none.
padding-bottom	Odstęp od dolnej krawędzi	Dowolna liczba punktów [pt], cali [in], centymetrów [cm] albo pikseli [px]; procent szerokości elementu nadrzędnego [%].
padding-left	Odstęp od lewej krawędzi	Dowolna liczba punktów [pt], cali [in], centymetrów [cm] albo pikseli [px]; procent szerokości elementu nadrzędnego [%].
padding-right	Odstęp od prawej krawędzi	Dowolna liczba punktów [pt], cali [in], centymetrów [cm] albo pikseli [px]; procent szerokości elementu nadrzędnego [%].
padding-top	Odstęp od górnej krawędzi	Dowolna liczba punktów [pt], cali [in], centymetrów [cm] albo pikseli [px]; procent szerokości elementu nadrzędnego [%].
text-align	Wyrówn	Lewostronny: left; wyśrodkowany:

	anie tekstu	center; prawostronny: right; wyjustowany: justify.
text-decoration	Efekty dodatko we	Żadne: none; podkreślenie: underline; linia nad tekstem: overline; przekreślenie: line-through; miganie: blink.
text-indent	Wcięcie tekstu	Dowolna liczba punktów [pt], cali [in], centymetrów [cm] albo pikseli [px]; procent szerokości elementu nadrzędnego [%].
text-transform	Przeksz tałcenie tekstu	Kapitałiki: capitalize; duże litery: uppercase; małe litery: lowercase; żadne: none.
line-height	Wysoko ść wiersza	Normalna: normal; dowolna liczba punktów [pt], cali [in], centymetrów [cm] albo pikseli [px]; procent wartości rozmiarów czcionki [%].
letter-spacing	Rozsta w liter	Normalny: normal; dowolna liczba punktów [pt], cali [in], centymetrów [cm] albo pikseli [px].
word-spacing	Odstępy miedzy wyraza mi	Normalny: normal; dowolna liczba punktów [pt], cali [in], centymetrów [cm] albo pikseli [px].
margin-left	Lewy margine s	Automatyczny: auto; dowolna liczba punktów [pt], cali [in], centymetrów [cm] albo pikseli [px]; procent wartości z szerokości elementu nadrzędnego [%].
margin-right	Prawy margine s	Automatyczny: auto; dowolna liczba punktów [pt], cali [in], centymetrów [cm] albo pikseli [px]; procent wartości z szerokości elementu nadrzędnego [%].
margin-top	Górny margine s	Automatyczny: auto; dowolna liczba punktów [pt], cali [in], centymetrów [cm] albo pikseli [px]; procent wartości z wysokości elementu nadrzędnego [%].
margin-bottom	Dolny margine s	Automatyczny: auto; dowolna liczba punktów [pt], cali [in], centymetrów [cm] albo pikseli [px]; procent wartości z wysokości elementu nadrzędnego [%].

vertical-align	Wyrównanie w pionie	Do linii bazowej: baseline; do góry: top; do góry tekstu: text-top; do środka: middle; do dołu tekstu: text-bottom; procentowa wartość bieżącej szerokości wiersza [%]; do wysokości indeksu dolnego: sub; do wysokości indeksu górnego: super.
----------------	---------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

9.) Tworzenie menu za pomocą listy uporządkowanej.

Aby stworzyć menu można do tego celu wykorzystać listę uporządkowaną `` wraz z zastosowaniem odnośników - hiperłączy `<a>`. Najprostsza konstrukcja:

```
<ul id="menu">
  <li><a href="http://www.onet.pl"> Hiperłącze do Onet.pl </a></li>
  <li><a href="http://www.wp.pl"> Hiperłącze do Wp.pl </a></li>
  <li><a href="http://www.gazeta.pl"> Hiperłącze do Gazeta.pl </a></li>
</ul>
```

W ten prosty sposób powstaje najprostsze menu z odnośnikami do innych stron. Aby lepiej wyglądało można go ostylezować za pomocą CSS. (cały kod w załączeniu oraz `koza/~marcin/TWWW/index1.html`) Aby ostylezować musimy ostylezować element ``, `` oraz `<a>` które składają się na najprostsze menu.

Tu o marginesach na stronie - model pudełkowy:

- margines zewnętrzny: `margin: u b l r;`
`margin-left: 10px;`
- obramowanie: ramka: `border: width style color;`
`border-style: solid;`
`border-color: #XXXXXX;`
`border-width: 10px;`
`border-top-style: dashed;`
`border-bottom-color: #XXXXXX;`
`border-left-width: 2px;`
- margines wewnętrzny: `padding: u b l r;`
`padding-right: 2px;`

Ważniejsze cechy:

- `list-style: none;` -> zmiana punktów na liście ``
- `display: block;` -> jak przeglądarka ma wyświetlać elementu: blok tekstu, linia tekstu, element listy. Możliwe wartości: `block`, `inline`, `none` (w tym kontekście `visibility:hidden`);
- `text-decoration: none;` - brak podkreślenia odnośnika.

Podobnie tworzymy menu poziome: z tą różnicą że elementom `` musimy nadać własność `float:left`; (`right`, `none`).

To powoduje że elementy z tą cechą będą dryfować w zdefiniowaną stronę.

3) Model pudełkowy / przykład strony z nagłówkiem, sidebarem i stopką.

Mamy już menu teraz można zrobić szkielet strony w oparciu o `<div>`.

Różnica pomiędzy `<div>` a `` ??

Zwykle do budowy stron używamy elementów blokowych <div>.

```
<div style="clear:both;"></div>
```

Aby uniknąć tego efektu pływania z opcji float należy jakiemuś elementowi **za** elementem z **float** nadać właściwość **clear** — powoduje ona ponowne „złączenie” float z biegiem dokumentu.

left

dany element będzie zsunięty poniżej wszystkich z **float:left**, ale pozwoli na opływanie go z prawej.

right

element będzie zsunięty poniżej wszystkich z **float:right**, ale pozwoli na opływanie go z lewej.

both

element będzie zsunięty poniżej wszystkich poprzedzających go floatów

none

Domyślna wartość oznaczająca brak wpływu na floaty (wyłącza działanie **clear**).

10) Jeszcze na temat selektorów i reguł w CSS:

a) selektor uniwersalny: * {cecha: wartość;}

```
* {  
    color:red;    }
```

Stosuje się często do zerowania np paddingów i marginesów na początku dokumentu:

```
* { margin:0 ;  
    padding: 0;  
}
```

b) selektor potomka: zadziała tylko jeśli jest potomkiem <div>. Potomek znaczy znajduje się w tym selektorze na dowolnym poziomie zagnieżdżenia (potomek w dowolnym pokoleniu).

```
div span {  
    color:red;  
}
```

```
<div>  
    <span> tttt </span>  
</div>
```

```
<div>  
    <p>  
        <span> ttt </span>  
    </p>  
</div>
```

```
<p> <span> tekst 3 </span></p>
```

c) selektor dziecka: zadziała tylko dla pierwszego zagnieżdżenia:

```
div > span {
    color: green;
}
```

Może być zbudowany z większej ilości selektorów: `div > p > span {}`

d) Selektor brata: obejmuje swoim działaniem jeden selektor znajdujący się bezpośrednio za danym selektorem:

```
div + span { color: blue; }
<div> </div> <span> tekst </span>
```

Można również składać selektory braci:

```
div + p + span { color: red; }
<div> </div> <p> </p> <span> xxxx </span>
```

e) selektor braci: zadziała dla wszystkich elementów danego typu za danym selektorem:

```
div ~ span { color: red; }
```

```
<div> </div>
<span> czerwony </span>
<span> czerwony</span>
<span>czerwony</span>
```

```
<div> <span> czarny </span></div>
```

```
<hr/>
```

```
<span> czerwony </span>
```

```
<div> <p> <span> czarny </span></p> </div>
```

Selektory atrybutów:

a) selektor atrybutu:

```
selektor[attribut] {cecha: wartosc;}
```

```
div[title] {color:red;}
```

```
<div title="www"> czerwony </div>
<hr/>
<p title="www"> czarnyy </p>
```

lub

```
[attribut] {cecha: wartosc;}
```

```
[title] {color:red;}

<div title="www"> czerwony </div>
<hr/>
<p title="www"> czerwony </p>
```

b) selektor atrybutu z wartością:

```
selektor[atribut="wartość"] {cecha: wartosc;}
    [atribut="wartość"] {cecha: wartosc;}
    [title="www"] { color: red;
                    }

<div title="www"> tttttt </div>
<hr/>
<p title="qqq"> kkk </p>
```

c) selektor dla atrybutu zawierającego dany wyraz:

```
selektor[atribut~="wartość"] {cecha: wartosc;}
    [atribut~="wartość"] {cecha: wartosc;}

[title~="www"] { color: red;
               }

<div title="www"> tttttt </div>
<hr/>
<p title="www qqq"> kkk </p>
```

d) selektor atrybutu z wyrazem z myślnikiem (łącznikiem):

```
selektor[atribut|="wartość"] {cecha: wartosc;}
    [atribut|="wartość"] {cecha: wartosc;}

[title|="www"] { color: red;
               }

<div title="www-kkkkkkk"> tttttt </div>
<hr/>
<p title="www-qqq"> kkk </p>
<hr/>
<p title="www-dddddqqq"> kkk </p>
```

e) selektor atrybutu zawierający dany ciąg znaków:

```
selektor[atribut*="wartość"] {cecha: wartosc;}
    [atribut*="wartość"] {cecha: wartosc;}

[title*="www"] { color: red;
               }
```

```

<div title="ww-kkkkkkk"> tttttt </div>
<hr/>
<p title="www-qqq"> kkk </p>
<hr/>
<p title="kkkkkk kkkww-ddddddqqq"> kkk </p>

```

f) selektor atrybutu którego wartość zaczyna się od konkretnego ciągu znaków:

```

selektor[atribut^="wartość"] {cecha: wartosc;}
[atribut^="wartość"] {cecha: wartosc;}

```

```

[title^="www"] { color: red;
}
<div title="www-kkkkkkk"> tttttt </div>
<hr/>
<p title="www-qqq"> kkk </p>
<hr/>
<p title="kkkkkk kkkww-ddddddqqq"> kkk </p>

```

g) selektor atrybutu którego wartość kończy się na dany ciąg znaków:

```

selektor[atribut$="wartość"] {cecha: wartosc;}
[atribut$="wartość"] {cecha: wartosc;}

```

```

[title$="www"] { color: red;
}
<div title="www-kkkkkkkwww"> tttttt </div>
<hr/>
<p title="www-qqq www"> kkk </p>
<hr/>
<p title="kkkkkk kkkww-ddddddqqq"> kkk </p>

```

i) łączenie ze sobą warunków na kilka atrybutów dla bardziej precyzyjnego wyboru treści której chcemy nadać odpowiednie wartości:

```

selektor[atribut][atribut] {cecha: wartość;}
[atribut][atribut] {cecha: wartość;}

```

```

[title$="www"][lang="pl"] { color: red;
}
<div title="www-kkkkkkkwww" lang="pl"> tttttt </div>
<hr/>
<div title="www-kkkkkkkwww" lang="en"> tttttt </div>
<hr/>
<p title="www-qqq www"> kkk </p>

```

Selektory pseudo-elementów:

Czasem zachodzi potrzeba dostępu do określonych elementów na każdej stronie ale do których nie da się bezpośrednio odnieść: np. pierwsza litera, pierwsza linijka etc. Można by za każdym razem kiedy chcemy ostylizować pierwszą linijkę np. obejmować ją

elementem span i stylizować ten element ale jest to czasochłonne. Można je również wykorzystać do dodawania konkretnej treści. Ale aby rozwiązać ten problem można użyć tzw. pseudo-elementów:

a) zaczniemy od pierwszej linii: **first-line**

```
selektor:first-line { color: red;
    }
:first-line { color: red;
    }
```

Można z jednym dwukropkiem też zadziałać ale dla odróżnienia od pseudo-klas lepiej pisać z dwoma dwukropkami.

b) podobnie można zastosować: **first-letter**

```
selektor:first-letter{ color: red;
    }
:first-letter { color: red;
    }
```

c) pseudo element: **before** - dodaje tekst do każdego elementu określonym selektorem:

```
div:before {content:" --Dodaj ten tekst --";
    color: red;
    }
```

d) podobny efekt możemy osiągnąć za pomocą: **after**

```
div:after {content:" --Dodaj ten tekst --";
    color: red;
    }
```

e) zaznaczenie (nie działa we wszystkich przeglądarkach np. dopiero w IE9):

```
div::selection {
    color:red;
    }
```

wtedy zaznaczając tekst myszką powinien być on podświetlony na czerwono.

11) HTML 5

```
<!DOCTYPE html>
<html lang="pl">
<head>
<title>Witamy w HTML 5</title>
<meta charset="utf-8" />
<script src="js_file.js"></script>
<link rel="stylesheet" href="css_file.css" />
</head>
<body>
<!-- Tutaj będą m.in. nowe elementy HTML 5 -->
<h1>Witamy w HTML 5</h1>
```

```
</body>
</html>
```

Podstawowe nowe znaczniki ułatwiające tworzenie struktury / konspektu strony:

```
<header> - do wyodrębnienia nagłówka strony
<nav>,    - do wyodrębnienia nawigacji strony
<article> - do wyodrębnienia części strony - np post na blogu
<section>, - też do grupowania treści
<footer>, - do tworzenia stopki
<aside> - do tworzenia treści dodatkowej (np na prawo strony)
```

```
<!DOCTYPE html>
<html lang="pl">
<head>
<title>Witamy w HTML 5</title>
<meta charset="utf-8" />
<style rel="stylesheet">
article, aside, nav {display: block;}
article, aside {float: left;}
article {width: 500px;}
nav {width: 200px;}
</style>
</head>
<body>
<!-- Tutaj będą m.in. nowe elementy HTML 5 -->
<article>
<header>
  
  <h1>Witamy w HTML 5</h1>
</header>
<p style="font-family: Arial;">
<span class="html">HTML 5</span> oferuje wiele nowych możliwości. Jednak
minie
trochę czasu, zanim możliwości te będą w pełni dostępne.
</p>
<footer>
<small>&copy; Projekt HTML 5</small>
</footer>
</article>
<aside>
  <h2>Polecane linki</h2>
  <nav>
    <ul>
      <li><a href="#">Strona główna</a></li>
      <li><a href="#">O witrynie</a></li>
      <li><a href="#">Projekt</a></li>
      <li><a href="#">Wiadomości</a></li>
      <li><a href="#">Kontakt</a></li>
    </ul>
  </nav>
</aside>
```

```
</body>  
</html>
```