

Narzędzia internetowe – laboratorium

CSS – Układ strony

Cel zajęć

Celem laboratorium jest poznanie zasad formatowania układu stron (ang. *page layout*) HTML z wykorzystaniem kaskadowych arkuszy stylów CSS3.

Zadanie 1

Utwórz nowy projekt HTML5. Pamiętaj o dodaniu pliku CSS, który będzie zawierał reguły formatowania układu strony. Następnie skopiuj i sprawdź działanie kodu umieszczonego na listingu 1 oraz listingu 2.

Listing 1. Kod HTML strony *index.html*

```
<!DOCTYPE html>
<html>
  <head>
    <title>Przygody Kubusia</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width">
    <link href="style.css" type="text/css" rel="stylesheet" />
  </head>
  <body>
    <header id="page_header">
      <h1>Przygody Kubusia</h1>
    </header>
    <nav>
      <ul>
        <li>Kubuś Puchatek</li>
        <li>Kubuś Fatalista</li>
      </ul>
    </nav>
    <article>
      <header>
        <h1>Kubuś Puchatek</h1>
      <p>
        Jestem Misiem o Bardzo Małym Rozumku i długie słowa sprawiają mi wielką trudność.
      </p>
    </header>
    <section id="extended">
      <p>
        Gdy Puchatek zobaczył, co jest w paczuszcze, o mało co nie przewrócił się z radości.
        Była to skrzyneczka kolorowych ołówków. Były tam ołówki oznaczone literami M. R. – to znaczy dla Misia
        Ratownika, na innych znów były litery D.M. – co oznaczało dla Dzielnego Misia. Był tam jeszcze
        szczyrek do ostrzenia ołówków i gumka do wycierania tego wszystkiego, co się źle napisało,
        i linia do rysowania linijek, żeby na nich równo pisać słowa, i podziałka, zrobiona na linii, żeby
        można nią było wszystko wymierzyć, i Niebieskie Ołówki, i Czerwone Ołówki, i Zielone Ołówki, żeby
        mówić o rozmaitych rzeczach na niebiesko, na czerwono i na zielono.
      </p>
    </section>
  </article>
  <aside>
    <span>
      Im bardziej Puchatek zaglądał do środka, tym bardziej tam Prosiaczka nie było.
    </span>
  </aside>
  <footer>Copyright © Stumilowy Las</footer>
</body>
</html>
```

Listing 2. Reguly formatowania pliku *style.css*

```
#page_header {
    background-color: deeppink;
    color:white;
    text-align:center;
    padding:5px;
}
nav {
    line-height:30px;
    background-color:#eeeeee;
    height:300px;
    width:20%;
    float:left;
    padding:5px;
}
article {
    width:50%;
    float:left;
    padding:10px;
}
aside {
    background-color: powderblue;
    height:300px;
    width:20%;
    float:right;
    padding:10px;
}
footer {
    background-color: dodgerblue;
    color:white;
    clear:both;
    text-align:center;
    padding:5px;
}
```

Po otwarciu w przeglądarce strona powinna wyglądać jak na rys. 1.



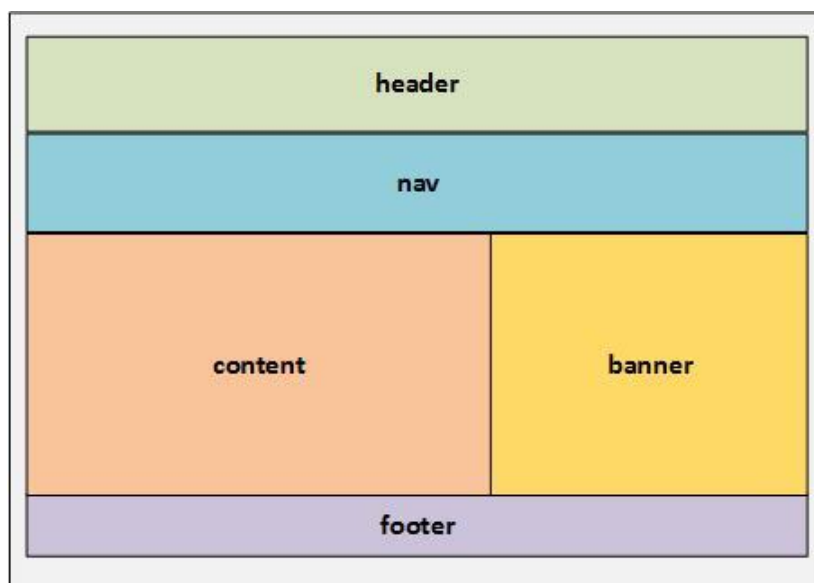
Rys. 1. Strona po uruchomieniu w przeglądarce

Zwróć uwagę na reguły najczęściej wykorzystywane podczas tworzenia układu strony:

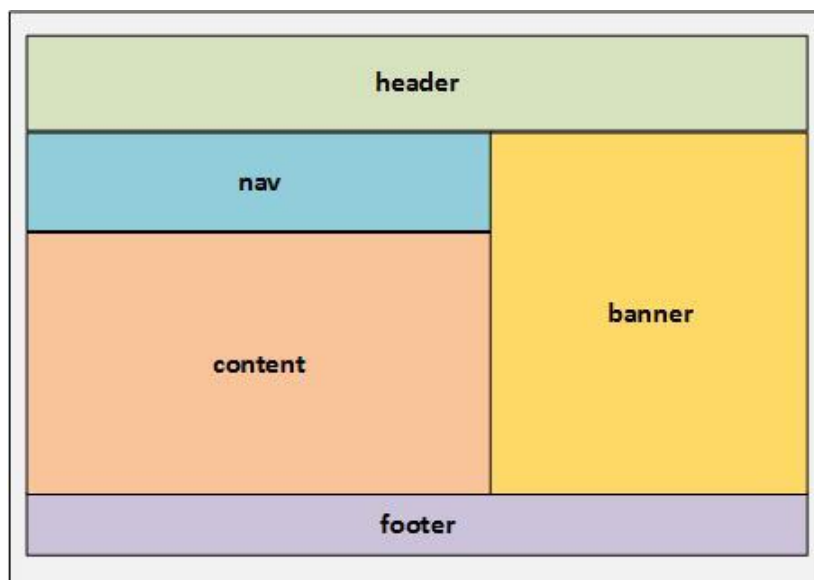
- height
- width
- float
- padding
- margin
- clear
- display

Zadanie 2

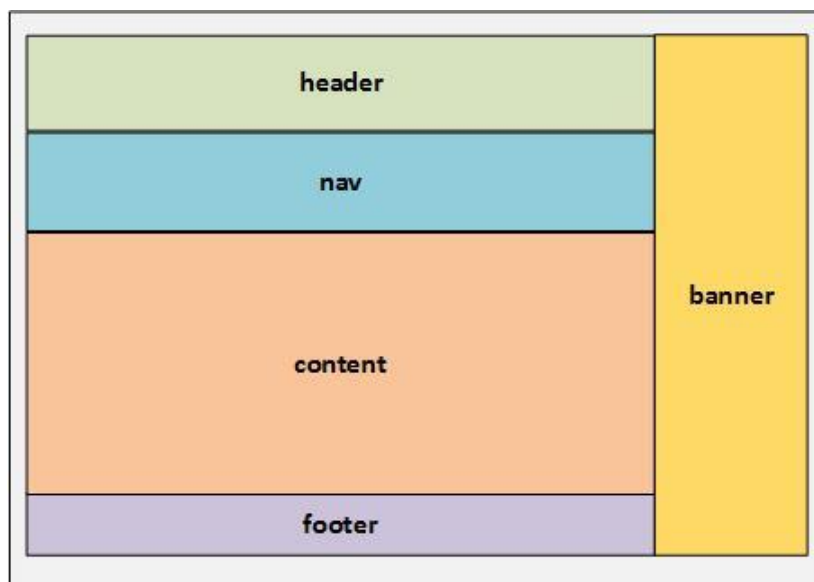
Utwórz nowy projekt HTML5. W projekcie zaprojektuj trzy szablony przedstawione na rys. 2, rys. 3, rys. 4 stosując pozycjonowanie oparte o kontenery tj. bloki header, nav, section, article, footer, div, aside. Poszczególne kontenery (bloki) wyróżnij kolorem tła.



Rys. 2. Układ strony A



Rys. 3. Układ strony B



Rys. 4. Układ strony C

Zastosuj układ elastyczny, szerokość bloków podawaj jako wartość procentową.

Wszystkie szablony zapisz na oddzielnych plikach HTML oraz CSS.

Zadanie 3

Utwórz nowy projekt HTML5, który będzie wykorzystywał układ *Flexible Box*. Następnie skopiuj i sprawdź działanie kodu umieszczonego na listingu 3.

Listing 3. Układ *Flexible Box Layout*

```
<!DOCTYPE html>
<html>
  <head>
    <title>Flexible box model</title>
    <style>
      .flex {
        /* basic styling */
        width: 350px;
        height: 200px;
        border: 1px solid #555;
        font: 14px Arial;

        /* flexbox setup */
        display: -webkit-flex;
        -webkit-flex-direction: row;

        display: flex;
        flex-direction: row;
      }

      .flex > div {
        -webkit-flex: 1 1 auto;
        flex: 1 1 auto;
        width: 30px;
        -webkit-transition: width 0.7s ease-out;
        transition: width 0.7s ease-out;
      }
    </style>
  </head>
  <body>
    <div class="flex">
      <div class="header">header</div>
      <div class="nav">nav</div>
      <div class="content">content</div>
      <div class="footer">footer</div>
      <div class="banner">banner</div>
    </div>
  </body>
</html>
```

```
}

.flex > div:nth-child(1){ background : #009246; }
.flex > div:nth-child(2){ background : #F1F2F1; }
.flex > div:nth-child(3){ background : #CE2B37; }

.flex > div:hover {
    width: 200px;
}

</style>
</head>
<body>
    <p>Flexbox</p>
    <div class="flex">
        <div>Blok 1</div>
        <div>Blok 2</div>
        <div>Blok 3</div>
    </div>
</body>
</html>
```

Więcej informacji o *Flexible Box Layout* na stronie dokumentacji: <http://www.w3.org/TR/css3-flexbox/>