

Akademia Górniczo-Hutnicza im. St. Staszica

CSS

**Model blokowy, symbolizacja
identyfikatorów, klas i pseudoklas**

Tomasz Bartuś

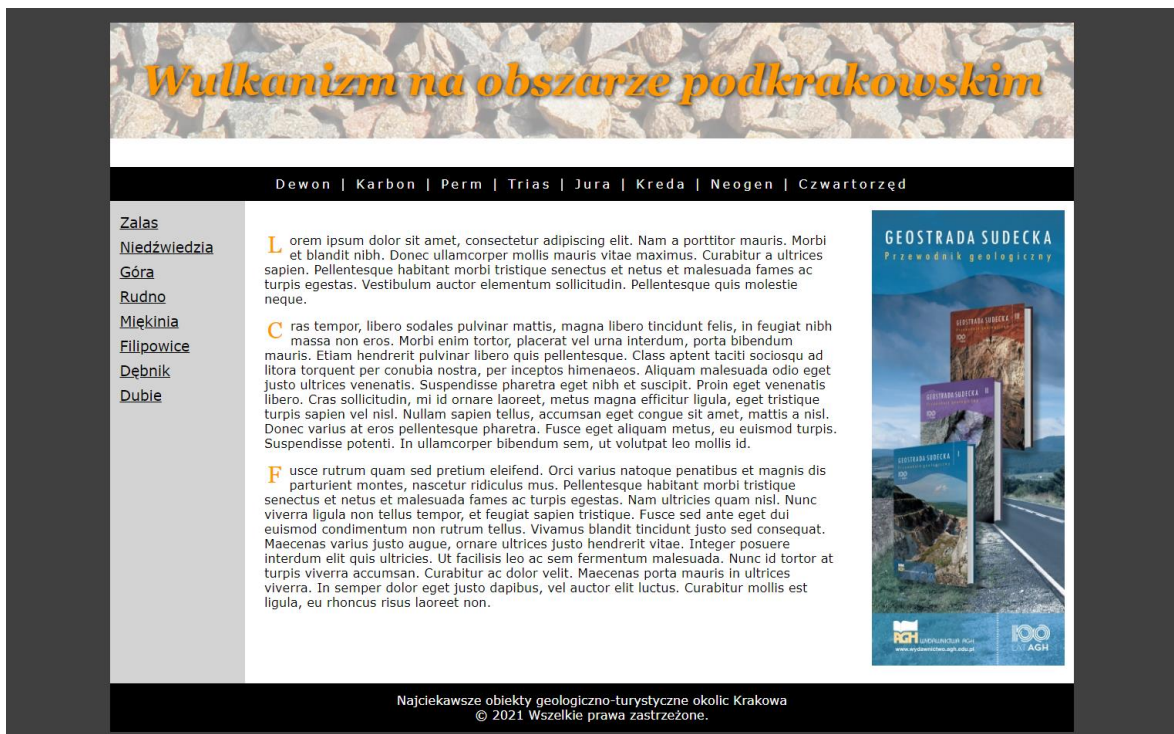
Wersja: 18.04.2024



2024

Wprowadzenie

W ramach kolejnego ćwiczenia utworzymy prostą stronę internetową opierającą się o kaskadowe arkusze styli (css) i model blokowy (pudełkowy). Tym razem poza przypomnieniem sobie sposobu tworzenia struktury modelu blokowego, zajmiemy się także stylizacją linków i kilku innych elementów stron www. Strona będzie miała wygląd przedstawiony na [Ryc. 1](#).

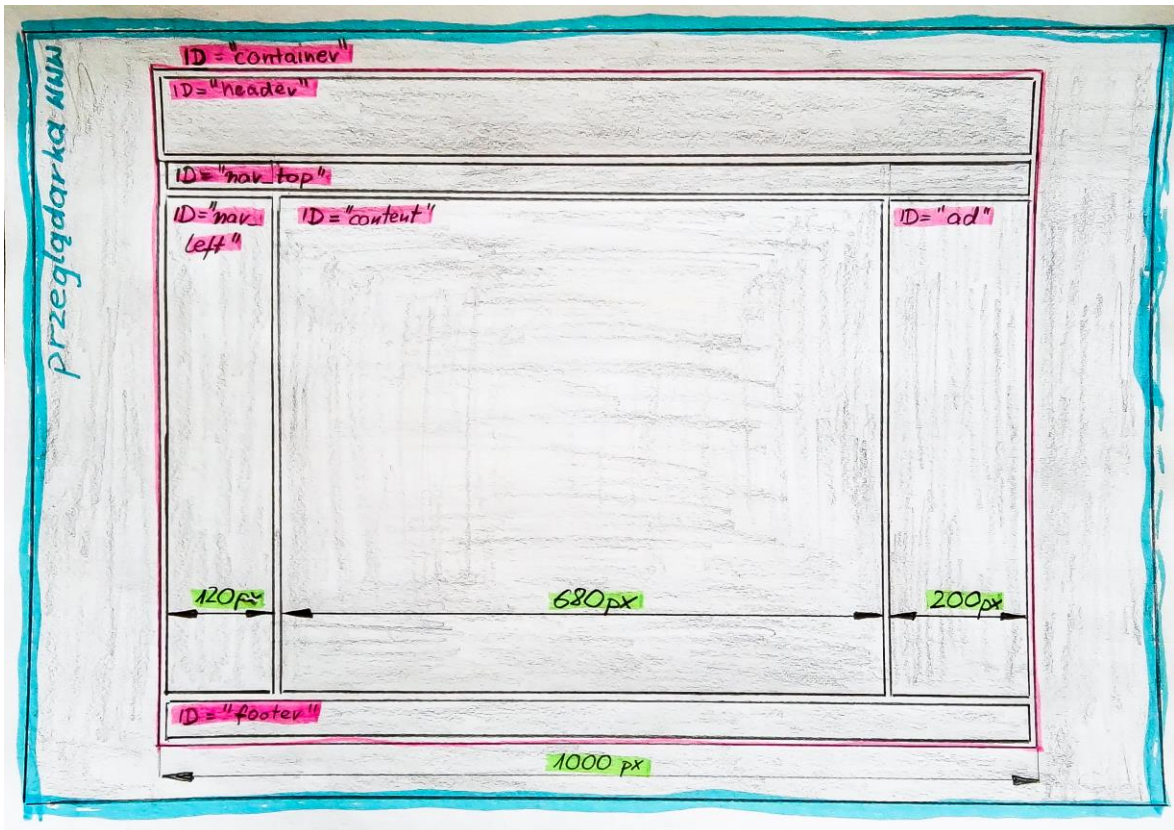


Ryc. 1. Wygląd projektowanej strony www

Założenia projektu strony www

Jak widać na [Ryc. 1](#) tym razem strona składa się z 6 bloków w kształtach prostokątów, które wypełniają niemal całą przestrzeń przeglądarki www. Strona jest wyśrodkowana, a po jej lewej i prawej stronie mamy białe marginesy dające wrażenie przestrzeni.

Schematycznie rozrysujmy sobie układ projektu strony www (Ryc. 2). Na schemacie zaznaczono proponowane nazwy identyfikatorów elementów blokowych `<div></div>` oraz szerokości bloków.



Ryc. 2. Schemat układu bloków projektu strony www wraz z wymiarowaniem

Ćwiczenie

Przejdźmy zatem do wykonania zaprojektowanej strony.

1. W wybranej lokalizacji utwórz folder ćwiczeniowy `\css_02\`.



2. Jeśli go nie posiadasz utwórz w programie Notepad++ szablon dokumentów html.

```
<!DOCTYPE HTML>
<html lang="pl">
<head>
  <meta charset="utf-8">
  <title></title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>

</body>
</html>
```



3. Jeśli jeszcze nie masz szablonu plików html zachowaj plik pod nazwą `template.html` i utwórz w folderze ćwiczeniowym `\css_02\` jego kopię o nazwie `index.html`.



4. W folderze ćwiczeniowym `\css_02\` utwórz w programie Notepad++ pusty plik o nazwie `style.css`.



5. Do sekcji `<head></head>` dodaj tytuł strony oraz metainformacje dotyczące słów kluczowych oraz opisu strony.

```
<title>Wulkanizm permski w okolicach Krakowa</title>

<meta name="description" content="Najważniejsze stanowiska
geologiczno-turystyczne opisujące późnopaleozoiczną działalność
wulkaniczną w obszarze krakowskim">

<meta name="keywords" content="Kraków, geologia, wulkanizm,
Zalas, Miękinia, Rudno">
```



6. Wewnątrz tagu `<body>` utwórz element blokowy `<div>` o identyfikatorze `id="container"`. Tradycyjnie będzie on zawierał wszystkie bloki niższego rzędu.

```
<body>

<div id="container">
</div>
```

```
</body>
```



7. Wewnątrz utworzonego bloku kontenera utwórz blok o identyfikatorze `id="header"`, a w nim tag nagłówka `<h1></h1>`, a do jego wnętrza wpisz tekst: „Wulkanizm na obszarze podkrakowskim”.

```
<div id="container">
  <div id="header">
    <h1>Wulkanizm na obszarze podkrakowskim</h1>
  </div>
</div>
```



8. Poniżej bloku `header` dodamy blok z poziomą belką nawigacyjną o identyfikatorze `nav_top`.

```
<div id="container">
  <div id="header">
    <h1>Wulkanizm na obszarze podkrakowskim</h1>
  </div>

  <div id="nav_top">

  </div>
</div>
```



9. Do wnętrza bloku o identyfikatorze `nav_top` wprowadźmy teksty linków. Rozdzielmy je znakiem pipe („|”).

```
<div id="nav_top">
  Devon |
  Karbon |
  Perm |
  Trias |
  Jura|
  Kreda |
  Neogen |
  Czwartorzęd
```

```
</div>
```



10. Do utworzonych tekstów linków dodajmy tag `<nav></nav>`. Będzie on informował przeglądarkę www, że ma w tym miejscu do czynienia z **głównym** menu nawigacyjnym strony www.

```
<div id="nav_top">
  <nav>
    Dewon |
    Karbon |
    Perm |
    Trias |
    Jura |
    Kreda |
    Neogen |
    Czwartorzęd
  </nav>
</div>
```



11. Zapiszmy wprowadzone zmiany i wyświetlmy plik `index.html` w przeglądarce internetowej (Ryc. 3).



Ryc. 3. Wygląd strony `index.html` bez deklaracji css



12. Zajmijmy się teraz deklaracją wyglądu utworzonych elementów strony www. Otwórzmy pusty plik `style.css` w programie Notepad++.



13. Zmieńmy tło całej strony www. W tym celu wprowadźmy do pliku css nagłówek tagu `<body>`.

```
body{
```

}

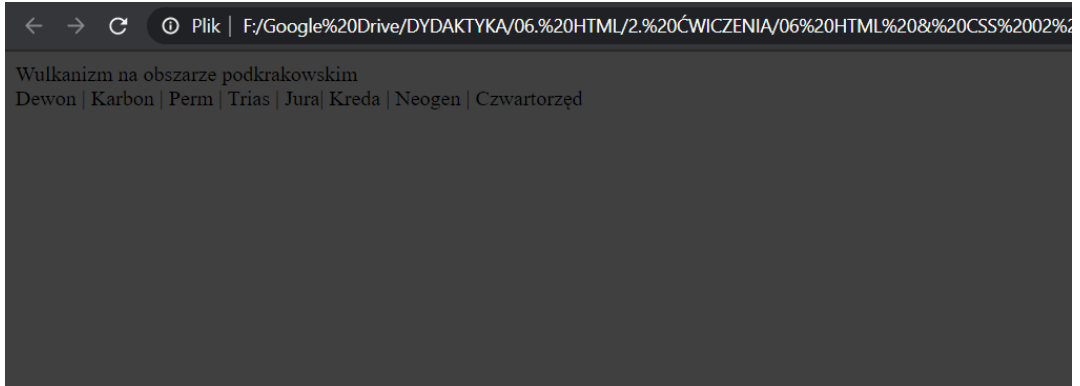


14. Do wnętrza deklaracji sekcji `<body>` wprowadź deklarację koloru tła.

```
body{  
    background-color: #404040;  
}
```



15. Odświeżmy wygląd strony w przeglądarce www ([Ryc. 4](#)).



Ryc. 4. Wygląd strony www po zmianie koloru tła strony

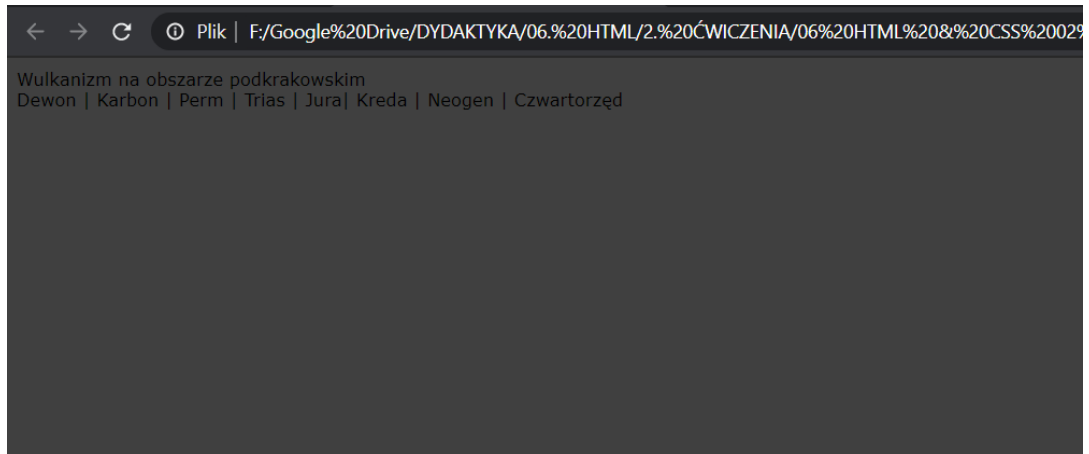


16. Zmieńmy teraz krój oraz domyślną wielkość czcionki. Do deklaracji wyglądu sekcji `body` dodajmy:

```
body{  
    background-color: #404040;  
    font-family: Verdana, Arial, sans-serif;  
    font-size: 13px;  
}
```



17. Odświeżmy wygląd strony w przeglądarce www ([Ryc. 5](#)).



Ryc. 5. Wygląd strony www po zmianie fontów i wielkości czcionki

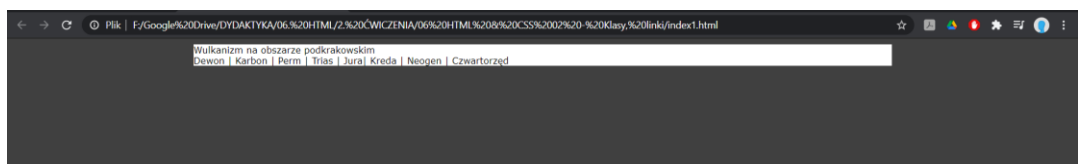


18. Dodajmy do pliku `style.css` deklarację wyglądu bloku kontenera. Zmieńmy szerokość bloku na 1000 pikseli, wyśrodkujmy blok i zmieńmy kolor jego tła na biały.

```
#container{
  width: 1000px;
  margin-left: auto;
  margin-right: auto;
  background-color: white;
}
```



19. Odświeżmy wygląd strony w przeglądarce www ([Ryc. 6](#)).



Ryc. 6. Wygląd strony www po dodaniu deklaracji wyglądu bloku `container`



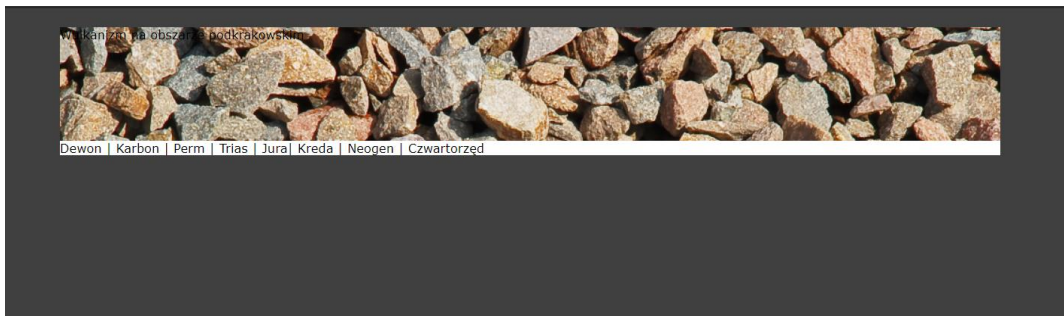
20. Zajmiemy się teraz wyglądem bloku `header`. W jego tle umieścimy obrazek (plik `header.png` w załącznikach do ćwiczenia). Wielkość obrazka to 1000 × 120 pikseli dlatego musimy ustawić minimalną wysokość bloku na 120 pikseli. Dla poprawienia wyglądu odsuniemy położenie bloku od

górnjej krawędzi przeglądarki o 20 pikseli.

```
#header{
  min-height: 120px;
  margin-top: 20px;
  background-image: url("../images/header.png");
}
```



21. Odświeżmy wygląd strony w przeglądarce www ([Ryc. 7](#)).



Ryc. 7. Wygląd strony www po dodaniu deklaracji wyglądu bloku header



22. Napis umieszczony wewnątrz bloku `header` przestał być widoczny. Musimy to naprawić. Wykorzystamy do tego deklarację bloku `<h1></h1>` znajdującego się wewnątrz bloku o identyfikatorze `header`. Dodajmy do pliku `style.css` deklarację o takiej treści:

```
#header h1{
}
```

Oznacza ona, że deklaracji poddajemy wygląd tagu `<h1></h1>` znajdujący się wewnątrz bloku o identyfikatorze `header`.



23. Do deklaracji nagłówka `<h1>` we wnętrzu bloku o identyfikatorze `header` dodamy deklarację fontów `Georgia`. Zmienimy wielkość czcionki napisu na 44 piksele i jej styl na kursywę.

```
#header h1{
```

```
font-family: Georgia;
font-size: 44px;
font-style: italic;
}
```



24. W tej samej deklaracji zmienimy też kolor czcionki na #ff9700, wycentrujemy napis wewnątrz bloku i dodamy napisowi cień. Względem napisu będzie on przesunięty w osiach X i Y o 1 piksel, będzie miał czarny, 5-cio pikselowy blur.

```
#header h1{
font-family: Georgia;
font-size: 44px;
font-style: italic;
color: #ff9700;
text-align: center;
text-shadow: 1px 1px 5px black;
}
```



25. Odświeżmy wygląd strony w przeglądarce www (Ryc. 8).



Ryc. 8. Wygląd strony www po dodaniu deklaracji wyglądu nagłówka

`<h1></h1>` wewnątrz bloku `header`



26. Napis nie jest wyrównany w pionie. Powinniśmy też nieco stonować kontrast obrazka ponieważ napis jest słabo czytelny i nie widać pod literami cienia. Nadamy tłu przezroczystość 40% (0,6).

```
#header h1{
font-family: Georgia;
font-size: 44px;
}
```

```
font-style: italic;
color: #ff9700;
text-align: center;
text-shadow: 1px 1px 5px black;
background: rgba(255,255,255,0.6);
line-height: 120px;
}
```



27. Odświeżmy wygląd strony w przeglądarce www ([Ryc. 9](#)).



Ryc. 9. Wygląd strony www po modyfikacji deklaracji wyglądu nagłówka `<h1></h1>` wewnątrz bloku `header`



28. Zajmijmy się teraz blokiem `nav_top`. Zmieńmy jego tło na czarne, kolor czcionki zmienmy na biały, wycentrujmy cały tekst względem bloku, aby odsunąć tekst od krawędzi bloku dodajmy 10-pikselowy margines wewnętrzny, a na koniec zmienmy rozstęp pomiędzy literami do 3 pikseli.

```
#nav_top{
background-color: black;
color: white;
text-align: center;
padding: 10px;
letter-spacing: 3px;
}
```



29. Odświeżmy wygląd strony w przeglądarce www ([Ryc. 10](#)).



Ryc. 10. Wygląd strony www po modyfikacji wyglądu bloku `nav_top`



30. Przyszła kolej aby zająć się utworzeniem trzech kolumn bloków o identyfikatorach `nav_left`, `content` i `ad` (Ryc. 2). Przejdźmy zatem do edycji pliku `index.html` i po bloku `nav_top` dodajmy kolejno trzy bloki:

```
<div id="nav_left">

</div>

<div id="content">

</div>

<div id="ad">

</div>
```



31. Blok `nav_left` posłuży nam do utworzenia menu pionowego złożonego z kilku linków, blok `content` tradycyjnie będzie zawierał treść strony, a `ad` – baner reklamy. Dodajmy do bloku `nav_left` linki menu pionowego.

```
<div id="nav_left">
  Zalas<br>
  Niedźwiedzia Góra<br>
  Rudno<br>
  Miękinia<br>
  Filipowice<br>
  Dębnik<br>
```

```
Dubie
```

```
</div>
```



29. Wejdźmy na stronę [Lipsum generator](#) i wygeneruj trzy akapity łacińskich i quasi-łacińskich wyrazów. Dodamy je do utworzonego bloku `content`. Każdy wygenerowany akapit wstaw w osobny tag paragrafu (`<p></p>`).

```
<div id="content">
```

```
<p>
```

```
    Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Nam a porttitor mauris. Morbi et blandit nibh. Donec
ullamcorper mollis mauris vitae maximus. Curabitur a ultrices
sapien. Pellentesque habitant morbi tristique senectus et netus
et malesuada fames ac turpis egestas. Vestibulum auctor elementum
sollicitudin. Pellentesque quis molestie neque.
```

```
</p>
```

```
<p>
```

```
    Cras tempor, libero sodales pulvinar mattis, magna
libero tincidunt felis, in feugiat nibh massa non eros. Morbi
enim tortor, placerat vel urna interdum, porta bibendum mauris.
Etiam hendrerit pulvinar libero quis pellentesque. Class aptent
taciti sociosqu ad litora torquent per conubia nostra, per
inceptos himenaeos. Aliquam malesuada odio eget justo ultrices
venenatis. Suspendisse pharetra eget nibh et suscipit. Proin eget
venenatis libero. Cras sollicitudin, mi id ornare laoreet, metus
magna efficitur ligula, eget tristique turpis sapien vel nisl.
Nullam sapien tellus, accumsan eget congue sit amet, mattis a
nisl. Donec varius at eros pellentesque pharetra. Fusce eget
aliquam metus, eu euismod turpis. Suspendisse potenti. In
ullamcorper bibendum sem, ut volutpat leo mollis id.
```

```
</p>
```

```
<p>
```

```
    Fusce rutrum quam sed pretium eleifend. Orci varius
natoque penatibus et magnis dis parturient montes, nascetur
ridiculus mus. Pellentesque habitant morbi tristique senectus et
netus et malesuada fames ac turpis egestas. Nam ultricies quam
nisl. Nunc viverra ligula non tellus tempor, et feugiat sapien
tristique. Fusce sed ante eget dui euismod condimentum non rutrum
tellus. Vivamus blandit tincidunt justo sed consequat. Maecenas
varius justo augue, ornare ultrices justo hendrerit vitae.
Integer posuere interdum elit quis ultricies. Ut facilisis leo ac
sem fermentum malesuada. Nunc id tortor at turpis viverra
accumsan. Curabitur ac dolor velit. Maecenas porta mauris in
ultrices viverra. In semper dolor eget justo dapibus, vel auctor
elit luctus. Curabitur mollis est ligula, eu rhoncus risus
laoreet non.
```

```
</p>
```

```
</div>
```



32. Do wnętrza bloku `ad` dodajmy obrazek (dostępny w załączniku do ćwiczenia).

```
<div id="ad">
    
</div>
```



33. Odświeżmy wygląd strony w przeglądarce www (Ryc. 11).



Ryc. 11. Wygląd strony www po dodaniu treści bloków `nav_left`, `content` i `ad`



34. Jak widać bloki `nav_left`, `content` i `ad` ze znanych przyczyn ułożyły się jeden pod drugim. Przejdźmy teraz do edycji pliku `style.css` aby zdefiniować ich wygląd. Na początek ustawmy szerokości wszystkich trzech bloków (zob. Ryc. 2) oraz poinformujmy przeglądarki www, że mają je wyświetlać jeden obok drugiego.

```
#nav_left{
```



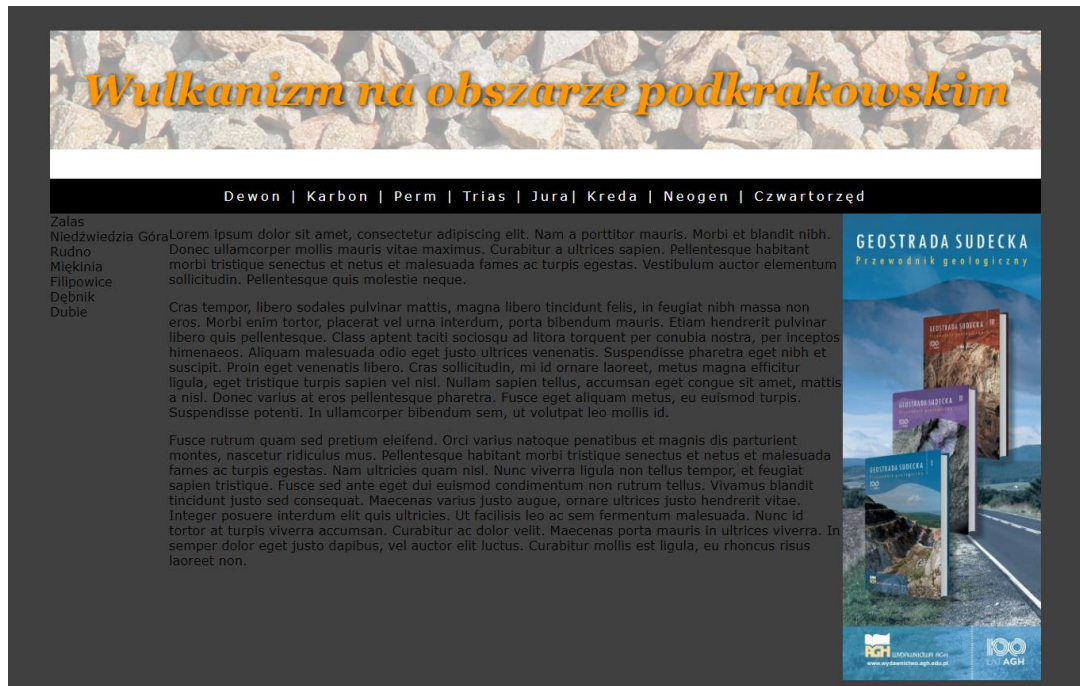
```

width: 120px;
float: left;
}
#content{
width: 680px;
float: left;
}
#ad{
width: 200px;
float: left;
}

```



35. Odświeżmy wygląd strony w przeglądarce www (Ryc. 12).



Ryc. 12. Wygląd strony www po modyfikacji szerokości i ułożenia bloków `nav_left`, `content` i `ad`



36. Dodajmy do bloków kolory tła.

```

#nav_left{
width: 120px;
float: left;
background-color: lightgray;
}

```

```
#content{
    width: 680px;
    float: left;
    background-color: #ffffff;
}
#ad{
    width: 200px;
    float: left;
    background-color: #ffffff;
}
```



37. Odświeżmy wygląd strony w przeglądarce www (Ryc. 13).



Ryc. 13. Wygląd strony www po modyfikacji kolorów tła bloków

nav_left, content i ad



38. Bloki sięgają w dół tylko do wysokości obejmowanej przez ich treść dlatego musimy do nich dodać deklarację minimalnej wysokości.

```
#nav_left{
    width: 120px;
    float: left;
    background-color: lightgray;
}
```



```

min-height: 480px;
}
#content{
width: 680px;
float: left;
background-color: #ffffff;
min-height: 480px;
}
#ad{
width: 200px;
float: left;
background-color: #ffffff;
min-height: 480px;
}

```



39. Odświeżmy wygląd strony w przeglądarce www (Ryc. 14).



Ryc. 14. Wygląd strony www po modyfikacji minimalnej wysokości bloków `nav_left`, `content` i `ad`



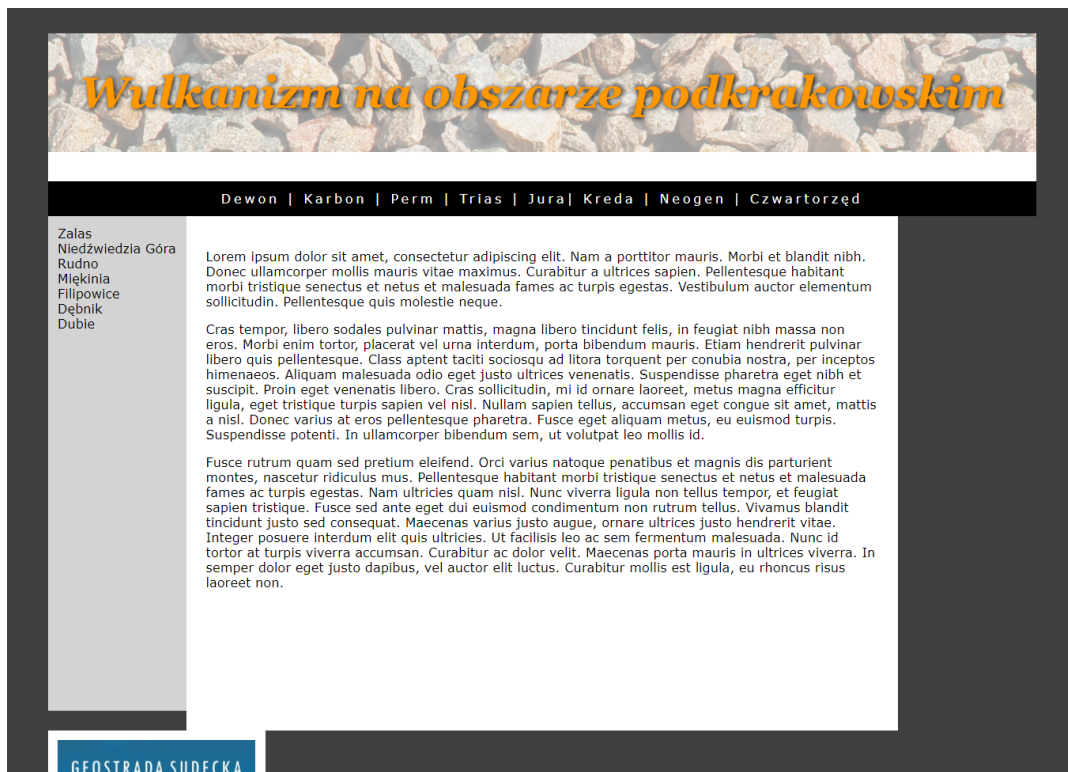
40. Treści występujące w blokach `nav_left`, `content` i `ad` brzydtko przylegają do krawędzi bloków, dodajmy im

marginesy wewnętrzne.

```
#nav_left{
  width: 120px;
  float: left;
  background-color: lightgray;
  min-height: 480px;
  padding: 10px;
}
#content{
  width: 680px;
  float: left;
  background-color: #ffffff;
  min-height: 480px;
  padding: 20px;
}
#ad{
  width: 200px;
  float: left;
  background-color: #ffffff;
  min-height: 480px;
  padding: 10px;
}
```



41. Odświeżmy wygląd strony w przeglądarce www ([Ryc. 15](#)).



Ryc. 15. Wygląd strony www po modyfikacji marginesów wewnętrznych bloków `nav_left`, `content` i `ad`



42. Po dodaniu paddingów pojawia się znany nam już problem. Bloki `nav_left`, `content` i `ad` nie mieszczą się wewnątrz 1000-pikselowego bloku `content`. Jesteśmy zmuszeni zmniejszyć ich szerokość. Sumaryczna szerokość lewych i prawych marginesów wewnętrznych wszystkich trzech bloków wynosi 80 pikseli (10+10+20+20+10+10). Taką wartość musimy odjąć np. od szerokości bloku `content`. Nowa szerokość bloku `content` będzie więc wynosiła 600 pikseli. Pozostaje jeszcze problem zmiany wysokości bloków po dodaniu paddingów. W tym przypadku blok `content` sięga o 20 pikseli niżej niż pozostałe bloki. Skorygujemy to odejmując tę wartość od minimalnej wysokości bloku `content`. Nowy parametr `min-height` dla tego obiektu będzie

więc teraz wynosił 460 pikseli.

```
#nav_left{
  width: 120px;
  float: left;
  background-color: lightgray;
  min-height: 480px;
  padding: 10px;
}
#content{
  width: 600px;
  float: left;
  background-color: #ffffff;
  min-height: 460px;
  padding: 20px;
}
#ad{
  width: 200px;
  float: left;
  background-color: #ffffff;
  min-height: 480px;
  padding: 10px;
}
```



43. Odświeżmy wygląd strony w przeglądarce www ([Ryc. 16](#)).



Ryc. 16. Wygląd strony www po modyfikacji szerokości i wysokości bloku content



44. Ostatnią rzeczą jaką zmienimy w wyglądzie edytowanych bloków będzie zwiększenie pionowych odstępów pomiędzy kolejnymi linkami menu lewego w bloku `nav_left`. Dodajmy 25 pikselową odległość pomiędzy kolejnymi elementami menu.

```
#nav_left{
  width: 120px;
  float: left;
  background-color: lightgray;
  min-height: 480px;
  padding: 10px;
  line-height: 25px;
}
```



45. Odświeżmy wygląd strony w przeglądarce www (Ryc. 17).



Ryc. 17. Wygląd strony www po modyfikacji odległości pomiędzy kolejnymi elementami bloku `nav_left`



46. Dodajmy teraz do pliku `index.html` blok stopki.

```
<div id="footer">
    Najciekawsze obiekty geologiczno-turystyczne okolic
    Krakowa<br>&copy; 2021 Wszelkie prawa zastrzeżone.
</div>
```



47. Do pliku `style.css` dodajmy definicję wyglądu bloku stopki `footer`. Jako, że jest to pierwszy blok po zmianie domyślnego ułożenia bloków `nav_left`, `content` i `ad` na jeden obok drugiego, musimy odwołać tę zmianę i tym samym przywrócić domyślny sposób ich układania jeden pod drugim. Służy do tego deklaracja `clear: both;`. Poza nią zmienimy kolor tła stopki na czarny, kolor czcionki na biały, wyśrodkujemy napis wewnątrz bloku i dodajmy 10-pikselowy margines wewnętrzny.

```
#footer{
```



```
clear: both;
background-color: black;
color: white;
text-align: center;
padding: 10px;
}
```



48. Odświeżmy wygląd strony w przeglądarce www (Ryc. 18).



Ryc. 18. Wygląd strony www po dodaniu deklaracji wyglądu bloku

footer



49. Strona wygląda już całkiem dobrze. Wyróżnimy w tekstach akapitów w bloku `content` pierwszą literę. Skorzystamy tu z deklaracji pseudoklasy `first-letter` dla tagu paragraf `<p></p>`. Na początek zwiększymy czcionkę pierwszej litery każdego paragrafu do 25 pikseli, nadajmy jej kolor `#ff9700`, zmieńmy czcionkę na Georgia.

```
p:first-letter{
font-size: 25px;
```

```
color: #ff9700;
font-family: Georgia;
}
```



50. Odświeżmy wygląd strony w przeglądarce www ([Ryc. 19](#)).

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam a porttitor mauris. Morbi et blandit nibh. Donec ullamcorper mollis mauris vitae maximus. Curabitur a ultrices sapien. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum auctor elementum sollicitudin. Pellentesque quis molestie neque.

Cras tempor, libero sodales pulvinar mattis, magna libero tincidunt felis, in feugiat nibh massa non eros. Morbi enim tortor, placerat vel urna interdum, porta bibendum mauris. Etiam hendrerit pulvinar libero quis pellentesque. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Aliquam malesuada odio eget justo ultrices venenatis. Suspendisse pharetra eget nibh et suscipit. Proin eget venenatis libero. Cras sollicitudin, mi id ornare laoreet, metus magna efficitur ligula, eget tristique turpis sapien vel nisl. Nullam sapien tellus, accumsan eget congue sit amet, mattis a nisl. Donec varius at eros pellentesque pharetra. Fusce eget aliquam metus, eu euismod turpis. Suspendisse potenti. In ullamcorper bibendum sem, ut volutpat leo mollis id.

Fusce rutrum quam sed pretium eleifend. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Nam ultricies quam nisl. Nunc viverra ligula non tellus tempor, et feugiat sapien tristique. Fusce sed ante eget dui euismod condimentum non rutrum tellus. Vivamus blandit tincidunt justo sed consequat. Maecenas varius justo augue, ornare ultrices justo hendrerit vitae. Integer posuere interdum elit quis ultricies. Ut facilisis leo ac sem fermentum malesuada. Nunc id tortor at turpis viverra accumsan. Curabitur ac dolor velit. Maecenas porta mauris in ultrices viverra. In semper dolor eget justo dapibus, vel auctor elit luctus. Curabitur mollis est ligula, eu rhoncus risus laoreet non.

Ryc. 19. Wygląd bloku `content` po dodaniu deklaracji wyglądu pseudoklasy `p:first-letter`



51. Odsuńmy nieco literę od lewej krawędzi oraz tekstu pierwszego wiersza. Dodajmy jej lewy i prawy `padding`.

```
p:first-letter{
font-size: 25px;
color: #ff9700;
font-family: Georgia;
padding-right: 8px;
padding-left: 3px;
}
```



52. Odświeżmy wygląd strony w przeglądarce www ([Ryc. 20](#)).

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam a porttitor mauris. Morbi et blandit nibh. Donec ullamcorper mollis mauris vitae maximus. Curabitur a ultrices sapien. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum auctor elementum sollicitudin. Pellentesque quis molestie neque.

Cras tempor, libero sodales pulvinar mattis, magna libero tincidunt felis, in feugiat nibh massa non eros. Morbi enim tortor, placerat vel urna interdum, porta bibendum mauris. Etiam hendrerit pulvinar libero quis pellentesque. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Aliquam malesuada odio eget justo ultrices venenatis. Suspendisse pharetra eget nibh et suscipit. Proin eget venenatis libero. Cras sollicitudin, mi id ornare laoreet, metus magna efficitur ligula, eget tristique turpis sapien vel nisl. Nullam sapien tellus, accumsan eget congue sit amet, mattis a nisl. Donec varius at eros pellentesque pharetra. Fusce eget aliquam metus, eu euismod turpis. Suspendisse potenti. In ullamcorper bibendum sem, ut volutpat leo mollis id.

Fusce rutrum quam sed pretium eleifend. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Nam ultricies quam nisl. Nunc viverra ligula non tellus tempor, et feugiat sapien tristique. Fusce sed ante eget dui euismod condimentum non rutrum tellus. Vivamus blandit tincidunt justo sed consequat. Maecenas varius justo augue, ornare ultrices justo hendrerit vitae. Integer posuere interdum elit quis ultricies. Ut facilisis leo ac sem fermentum malesuada. Nunc id tortor at turpis viverra accumsan. Curabitur ac dolor velit. Maecenas porta mauris in ultrices viverra. In semper dolor eget justo dapibus, vel auctor elit luctus. Curabitur mollis est ligula, eu rhoncus risus laoreet non.

Ryc. 20. Wygląd bloku content po dodaniu do deklaracji wyglądu pseudoklasy `p:first-letter` właściwości `padding-left` i `padding-right`



53. Pierwsza litera cały czas przylega do pierwszego wiersza, a gdyby cały paragraf potraktować jako blok i przyłożyć pierwsze litery akapitów do bloków?... Spróbujmy... Do deklaracji `p:first-letter` dodajmy właściwość `float: left;`.

```
p:first-letter{
  font-size: 25px;
  color: #ff9700;
  font-family: Georgia;
  padding-right: 8px;
  padding-left: 3px;
  float: left;
}
```



54. Odświeżmy wygląd strony w przeglądarce www (Ryc. 21).

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam a porttitor mauris. Morbi et blandit nibh. Donec ullamcorper mollis mauris vitae maximus. Curabitur a ultrices sapien. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum auctor elementum sollicitudin. Pellentesque quis molestie neque.

Cras tempor, libero sodales pulvinar mattis, magna libero tincidunt felis, in feugiat nibh massa non eros. Morbi enim tortor, placerat vel urna interdum, porta bibendum mauris. Etiam hendrerit pulvinar libero quis pellentesque. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Aliquam malesuada odio eget justo ultrices venenatis. Suspendisse pharetra eget nibh et suscipit. Proin eget venenatis libero. Cras sollicitudin, mi id ornare laoreet, metus magna efficitur ligula, eget tristique turpis sapien vel nisl. Nullam sapien tellus, accumsan eget congue sit amet, mattis a nisl. Donec varius at eros pellentesque pharetra. Fusce eget aliquam metus, eu euismod turpis. Suspendisse potenti. In ullamcorper bibendum sem, ut volutpat leo mollis id.

Fusce rutrum quam sed pretium eleifend. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Nam ultricies quam nisl. Nunc viverra ligula non tellus tempor, et feugiat sapien tristique. Fusce sed ante eget dui euismod condimentum non rutrum tellus. Vivamus blandit tincidunt justo sed consequat. Maecenas varius justo augue, ornare ultrices justo hendrerit vitae. Integer posuere interdum elit quis ultricies. Ut facilisis leo ac sem fermentum malesuada. Nunc id tortor at turpis viverra accumsan. Curabitur ac dolor velit. Maecenas porta mauris in ultrices viverra. In semper dolor eget justo dapibus, vel auctor elit luctus. Curabitur mollis est ligula, eu rhoncus risus laoreet non.

Ryc. 21. Wygląd bloku `content` po dodaniu do deklaracji wyglądu pseudoklasy `p:first-letter` właściwości `float: left;`



55. Wygląda to znacznie lepiej niż wcześniej. Mamy jednak problem, że ktoś mógłby próbować czytać 2-gi wiersz paragrafów razem z pierwszą literą. Powinniśmy jednoznacznie przypisać pierwszą literę akapitu do pierwszego wiersza. Wykorzystajmy do tego właściwość `line-height`: `25px`; (czyli tyle ile pikseli ma pierwsza litera).

```
p:first-letter{
  font-size: 25px;
  color: #ff9700;
  font-family: Georgia;
  padding-right: 8px;
  padding-left: 3px;
  float: left;
  line-height: 25px;
}
```



56. Odświeżmy wygląd strony w przeglądarce www (Ryc. 22).

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam a porttitor mauris. Morbi et blandit nibh. Donec ullamcorper mollis mauris vitae maximus. Curabitur a ultrices sapien. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum auctor elementum sollicitudin. Pellentesque quis molestie neque.

Cras tempor, libero sodales pulvinar mattis, magna libero tincidunt felis, in feugiat nibh massa non eros. Morbi enim tortor, placerat vel urna interdum, porta bibendum mauris. Etiam hendrerit pulvinar libero quis pellentesque. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Aliquam malesuada odio eget justo ultrices venenatis. Suspendisse pharetra eget nibh et suscipit. Proin eget venenatis libero. Cras sollicitudin, mi id ornare laoreet, metus magna efficitur ligula, eget tristique turpis sapien vel nisl. Nullam sapien tellus, accumsan eget congue sit amet, mattis a nisl. Donec varius at eros pellentesque pharetra. Fusce eget aliquam metus, eu euismod turpis. Suspendisse potenti. In ullamcorper bibendum sem, ut volutpat leo mollis id.

Fusce rutrum quam sed pretium eleifend. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Nam ultricies quam nisl. Nunc viverra ligula non tellus tempor, et feugiat sapien tristique. Fusce sed ante eget dui euismod condimentum non rutrum tellus. Vivamus blandit tincidunt justo sed consequat. Maecenas varius justo augue, ornare ultrices justo hendrerit vitae. Integer posuere interdum elit quis ultricies. Ut facilisis leo ac sem fermentum malesuada. Nunc id tortor at turpis viverra accumsan. Curabitur ac dolor velit. Maecenas porta mauris in ultrices viverra. In semper dolor eget justo dapibus, vel auctor elit luctus. Curabitur mollis est ligula, eu rhoncus risus laoreet non.

Ryc. 22. Wygląd bloku content po dodaniu do deklaracji wyglądu pseudoklasy `p:first-letter` właściwości `line-height`



57. Zajmijmy się teraz wystylizowaniem linków. Na pierwszy ogień weźmy linki poziomej belki nawigacyjnej z bloku `nav_top`. Otwórzmy w programie Notepad++ plik `index.html` i dodajmy do istniejących tekstów tagi ``.

W związku z tym, że za chwilę będziemy chcieli zmienić domyślny, nieciekawy wygląd linków i będziemy je chcieli odróżnić od linków menu pionowego z bloku `nav_left`, nadajmy im klasę np. o nazwie `nav_top_link`. Posłużymy się klasą, a nie identyfikatorem ponieważ na stronie będą się znajdowały więcej niż jeden element o tym identyfikatorze.

```
<div id="nav_top">
  <nav>
    <a href="" class="nav_top_link">Dewon</a> |
    <a href="" class="nav_top_link">Karbon</a> |
```

```

<a href="" class="nav_top_link">Perm</a> |
<a href="" class="nav_top_link">Trias</a> |
<a href="" class="nav_top_link">Jura</a> |
<a href="" class="nav_top_link">Kreda</a> |
<a href="" class="nav_top_link">Neogen</a> |
<a href="" class="nav_top_link">Czwartorzęd</a>

</nav>
</div>

```



58. Odświeżmy wygląd strony w przeglądarce www (Ryc. 23).



Ryc. 23. Wygląd linków bloku `nav_top` bez stylizacji za pomocą css



59. Jak widać, domyślny wygląd linków bez stylizacji za pomocą css nie jest zbyt ciekawy. Linki na zastosowanym czarnym tle nie są czytelne. Zajmijmy się ich stylizacją. W pliku `style.css` dodajmy deklaracje klasy `nav_top_link`. Do stylizacji zastosujemy zwykłą klasę `nav_top_link` (wygląd linku kiedy kursor jest poza linkiem) oraz dwie pseudoklasy dynamiczne klasy `nav_top_link:nav_top_link:hover` (wygląd linku po najechaniu na niego kursorem) i `nav_top_link:active` (wygląd linku w momencie kliknięcia na niego). Dodajmy do pliku odpowiednie nagłówki deklaracji.

```

.nav_top_link{
}
.nav_top_link:hover{
}
.nav_top_link:active{

```

}



60. Utwórzmy teraz następujący wygląd linków:

- Stan z kursorem poza linkiem: kolor linku biały, tekst bez podkreślenia,
- Stan `hover`: kolor linku `#ff9700`, tekst z podkreśleniem u góry i u dołu, kolor linii taki jak linku,
- Stan `active`: kolor linku czerwony, tekst z podkreśleniem u góry i u dołu, kolor linii czerwony.

```
.nav_top_link{
    color: white;
    text-decoration: none;
}
.nav_top_link:hover{
    color: #ff9700;
    text-decoration: underline overline #ff9700;
}
.nav_top_link:active{
    color: red;
    text-decoration: underline overline red;
}
```



61. Odświeżmy wygląd strony w przeglądarce www ([Ryc. 24](#)).



Ryc. 24. Wygląd linków bloku `nav_top` bez po stylizacji za pomocą css



62. Na koniec zajmijmy się linkami menu pionowego z bloku `nav_left`. Otwórzmy do edycji plik `index.html` i dodajmy do istniejących tekstów tagi ``. W związku z tym,

że chcemy je odróżnić od wyglądu linków z bloku `nav_top`, nadajmy im klasę np. o nazwie `nav_left_link`.

```
<div id="nav_left">
  <a href="" class="nav_left_link">Zalas</a><br>
  <a href="" class="nav_left_link">Niedźwiedzia
Góra</a><br>
  <a href="" class="nav_left_link">Rudno</a><br>
  <a href="" class="nav_left_link">Miękinia</a><br>
  <a href="" class="nav_left_link">Filipowice</a><br>
  <a href="" class="nav_left_link">Dębnik</a><br>
  <a href="" class="nav_left_link">Dubie</a>
</div>
```

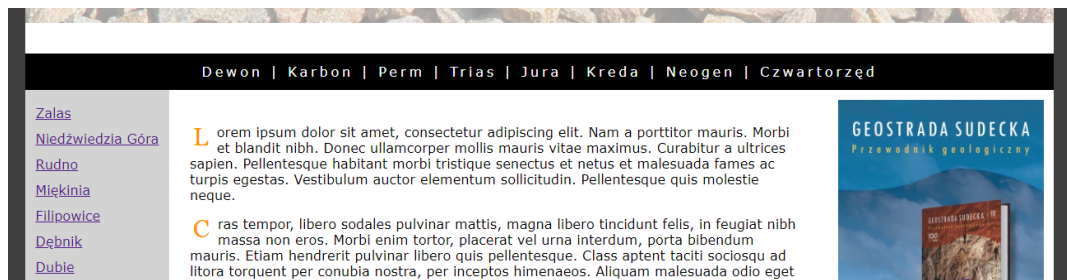


63. W pliku `style.css` dodajmy deklaracje klasy `nav_left_link`. Do stylizacji zastosujemy zwykłą klasę `nav_left_link` (wygląd linku kiedy kursor jest poza linkiem) oraz dwie pseudoklasy dynamiczne klasy `nav_left_link`: `nav_left_link:hover` (wygląd linku po najechaniu na niego kursorem) i `nav_left_link:active` (wygląd linku w momencie kliknięcia na niego). Dodajmy do pliku odpowiednie nagłówki deklaracji.

```
.nav_left_link{
}
.nav_left_link:hover{
}
.nav_left_link:active{
}
```



64. Odświeżmy wygląd strony w przeglądarce [www](#) (Ryc. 25).



Ryc. 25. Wygląd linków bloku `nav_left` bez po stylizacji za pomocą CSS



65. Przejdźmy do stylizacji wyglądu klasy `nav_left_link` oraz pseudoklas: `nav_left_link:hover` oraz `nav_left_link:active`. Utwórzmy teraz następujący wygląd linków:

- Stan z kursorem poza linkiem: kolor linku czarny, wielkość czcionki 15 pikseli,
- Stan `hover`: kolor linku biały, kolor tła `#ff9700`,
- Stan `active`: kolor linku `black`, kolor tła czerwony.

```
.nav_left_link{
    color: black;
    font-size: 15px;
}
.nav_left_link:hover{
    color: white;
    background-color: #ff9700;
}
.nav_left_link:active{
    color: black;
    background-color: red;
}
```



66. Odświeżmy wygląd strony w przeglądarce www (Ryc. 26).



Ryc. 26. Wygląd linków bloku `nav_left` po stylizacji za pomocą css



67. Na tym zakończymy projekt szablonu naszej nowej strony www. Zobaczmy jak w całości wygląda (Ryc. 27).



Ryc. 27. Gotowy projekt szablonu strony www