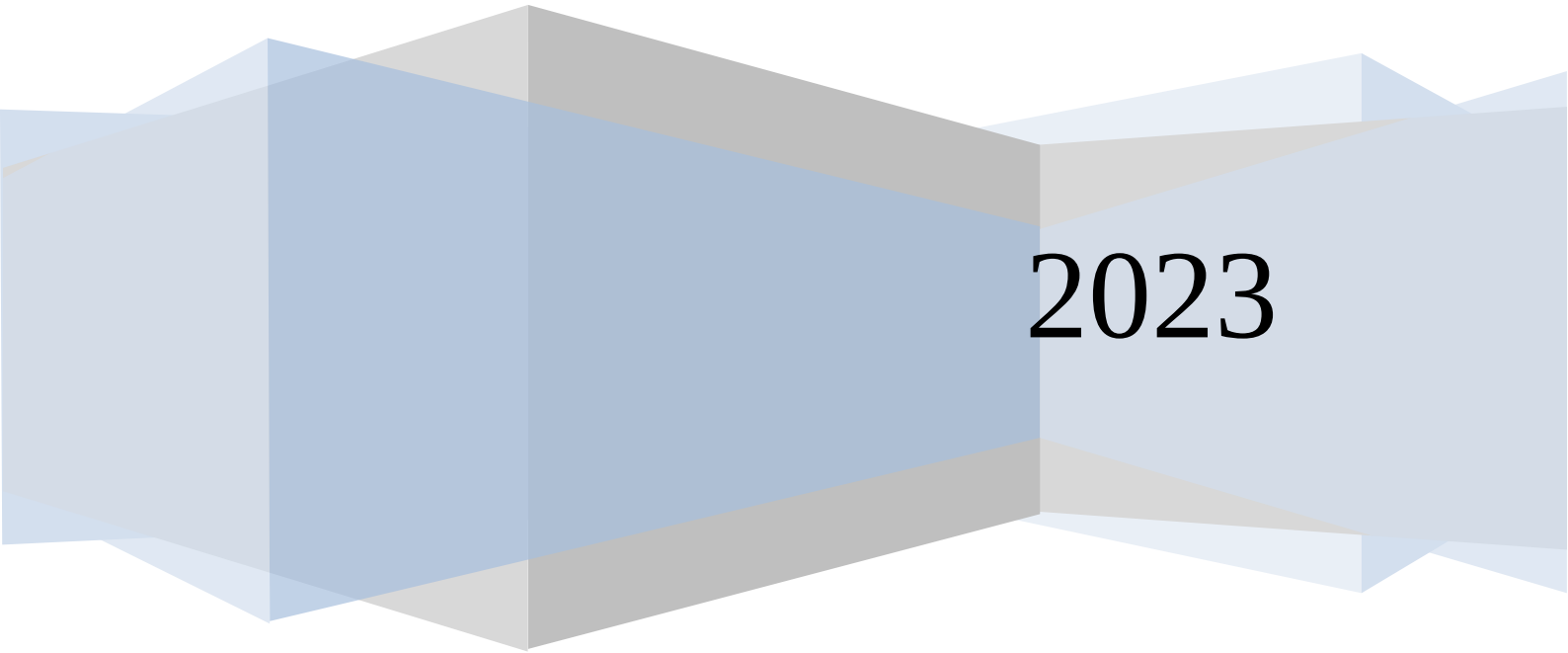
Akademia Górniczo-Hutnicza im. St. Staszica

CSS

Prosta strona oparta na modelu blokowym CSS

Tomasz Bartuś

Wersja: 18.04.2023



Wprowadzenie

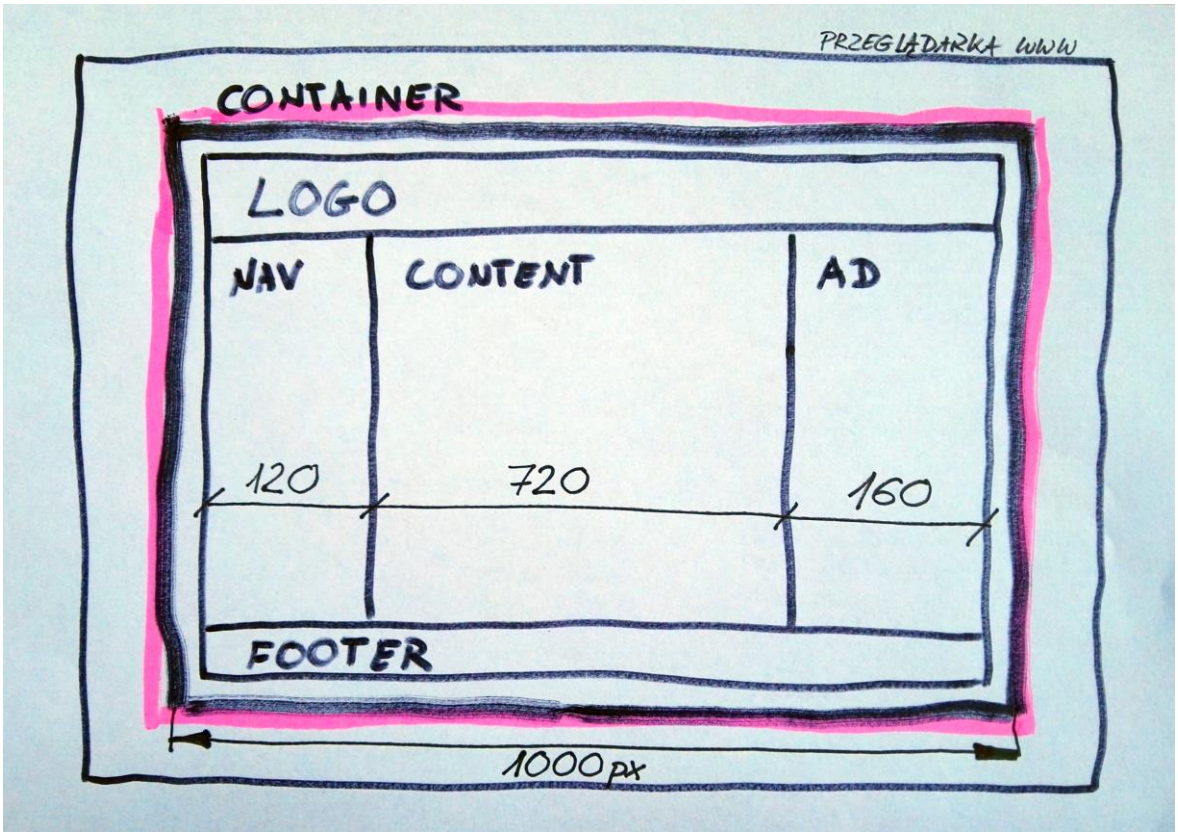
W ramach ćwiczenia utworzymy prostą stronę internetową opierającą się o kaskadowe arkusze styli (css) i model blokowy (pudełkowy). Strona będzie miała wygląd przedstawiony na Ryc. 1.

Ryc. 1. Wygląd projektowanej strony www

Założenia projektu strony www

Jak widać na Ryc. 1 strona składa się z 5 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_01`.

2. Jeśli go nie posiadasz utwórz w programie Notepad++ szablon dokumentów html. Będziemy go używali w dalszym ciągu ćwiczeń.

```
<!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. Zachowaj plik pod nazwą `template.html` i utwórz jego kopię pod nazwą `index.html` w folderze ćwiczeniowym `\css_01`

4. W folderze ćwiczeniowym `\css_01` utwórz w programie Notepad++ pusty plik o nazwie `style.css`.

5. W kodzie html Zmień tytuł dokumentu html `index.html` na:

```
<title>Geoturystyka - subiektywne TOP 5 podkrakowskich
geostanowisk</title>
```

6. Do sekcji `<head>` dodaj metainformacje dotyczące słów kluczowych i opisu strony.

```
<meta name="description" content="Serwis zajmuje się
popularyzacją obiektów geoturystycznych w Małopolsce">
<meta name="keywords" content="geoturystyka, geologia, Kraków,
amonity, skałki, jura">
```

7. Wewnątrz tagu `<body>` utwórz element blokowy `<div>` o identyfikatorze `id=”container”`. Będzie on w przyszłości zawierał wszystkie bloki niższego rzędu.

```
<body>


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


</body>
```

8. Wewnątrz utworzonego bloku kontenera utwórz blok o identyfikatorze `id=”logo”`, a następnie wstaw do niego nagłówek `<h1></h1>` wraz z napisem „`Najlepsze podkrakowskie geostanowiska`”.

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

```
    <div id="logo">
        <h1>Najlepsze podkrakowskie geostanowiska</h1>
    </div>
</div>
```

9. Teraz poniżej bloku o identyfikatorze `id="logo"` utworzymy kolejno trzy bloki, które docelowo będą rozmieszczone obok siebie. Będą to kolejno bloki o identyfikatorach:

    - `id="nav"`
    - `id="content"`
    - `id="ad"`

```
<div id="container">
    <div id="logo">
        <h1>Najlepsze podkrakowskie geostanowiska</h1>
    </div>

    <div id="nav">
    </div>

    <div id="content">
    </div>

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

10. Do wnętrza elementu blokowego o identyfikatorze `id="nav"` (pierwsza kolumna) dodajmy treść przyszłych linków:

```
    <div id="nav">
        Zakrzówek<br>
        Zalas<br>
        Rudno<br>
        Dębnik<br>
        Bonarka<br>
    </div>
```

11. Do wnętrza elementu blokowego o identyfikatorze `id="content"` (środkowa kolumna) dodajmy treść:

```
    <div id="content">
        <h1>Kopalnia porfiru w Zalasie</h1>
        <p>
            Zalas jest niewielką miejscowością położoną na zachód
od Krakowa i na południe od Krzeszowic. Istnieje tam duża
odkrywkowa kopalnia permskich skał wulkanicznych - porfirów. W
nadkładzie występują osady jury środkowej i górnej, wśród których
powszechnie występuje fauna amonitów, ramienionogów, małży,
belemnitów i inne.
        </p>
    </div>
```

12. Do wnętrza elementu blokowego o identyfikatorze `id="ad"` (trzecia kolumna) dodajmy obrazek (dostępny w załącznikach do ćwiczenia).

```
    <div id="ad">
        <img src="./images/baner_620wys96dpi.jpg" alt="Geostrada
Sudecka - przewodnik geologiczny" width="160">
    </div>
```

13. Do wnętrza elementu blokowego kontenera `id="container"` dodajmy ostatni, występujący na samym dole strony blok stopki.

```
    <div id="footer">
        Geoturystyka - subiektywne TOP 5 podkrakowskich
geostanowisk<br>
        &copy; Wszelkie prawa zastrzeżone
```

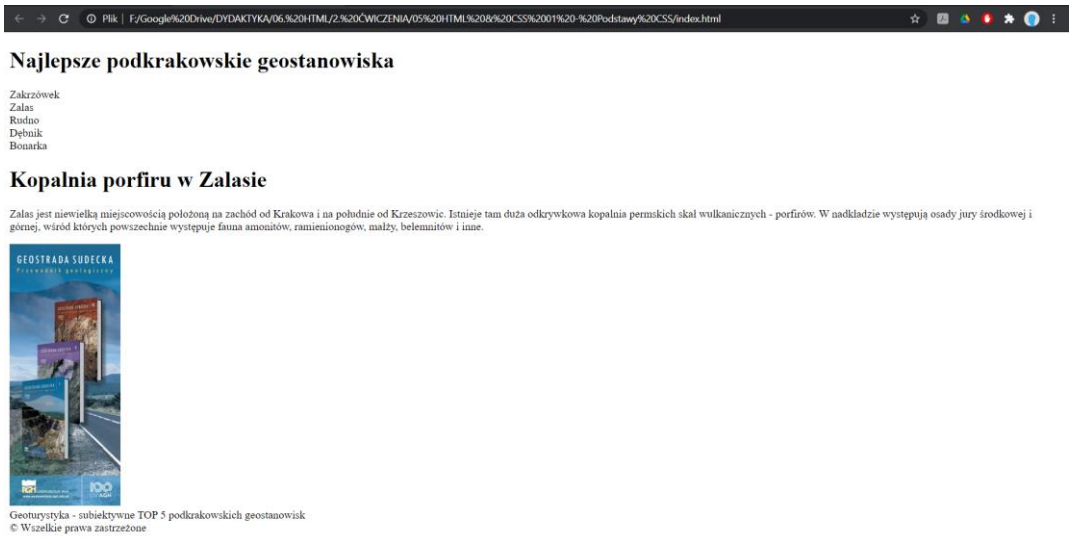
```
    </div>
```

Tym samym ukończyliśmy projekt kodu strony html. Strona powinna wyglądać tak jak w poniższym listingu:

```
<!DOCTYPE HTML>
<html lang="pl">
<head>
    <meta charset="utf-8">
    <title>Geoturystyka - subiektywne TOP 5 podkrakowskich geostanowisk</title>
    <meta name="description" content="Serwis zajmuje się popularyzacją obiektów geoturystycznych w Małopolsce">
    <meta name="keywords" content="geoturystyka, geologia, Kraków, amonity, skałki, jura">
    <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
<div id="container">
    <div id="logo">
        <h1>Najlepsze podkrakowskie geostanowiska</h1>
    </div>

    <div id="nav">
        Zakrzówek<br>
        Zalas<br>
        Rudno<br>
        Dębnik<br>
        Bonarka<br>
    </div>
    <div id="content">
        <h1>Kopalnia porfiru w Zalasie</h1>
        <p>
            Zalas jest niewielką miejscowością położoną na zachód od Krakowa i na południe od Krzeszowic. Istnieje tam duża odkrywkowa kopalnia permskich skał wulkanicznych - porfirów. W nadkładzie występują osady jury środkowej i górnej, wśród których powszechnie występuje fauna amonitów, ramienionogów, małży, belemnitów i inne.
        </p>
```

```
    </div>
    <div id="ad">
        <img src="./images/baner_620wys96dpi.jpg" alt="Geostrada
Sudecka - przewodnik geologiczny" width="160">
    </div>


    <div id="footer">
        Geoturystyka - subiektywne TOP 5 podkrakowskich
geostanowisk<br>
        &copy; Wszelkie prawa zastrzeżone
    </div>


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

14. Utworzony plik `index.html` obejrzyjmy w przeglądarce www Ryc. 3. Jak widać wszystkie elementy blokowe zostały poukładane kolejno jeden pod drugim. Jest to ustawienie domyślnie. Jeżeli chcemy aby bloki zostały ułożone jeden obok drugiego będziemy to musieli odpowiednio oznaczyć.



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

Teraz możemy przejść do deklaracji wyglądu strony www w pliku `style.css`.

CSS

15. Otwórzmy pusty plik `style.css` w programie Notepad++.

16. Utwórzmy w nim pustą deklarację wyglądu elementu blokowego kontenera (`<div id="container"></div>`). Elementy z identyfikatorami (`id=""`) oznaczamy za pomocą znaku „#”

```
#container{
}
```

17. Zdefiniujmy szerokość bloku kontenera na 1000 pikseli. Robimy to za pomocą pary deklaracji – właściwości i jej wartości. Każdą deklarację kończymy znakiem średnika.

```
#container{
    width: 1000px;
}
```

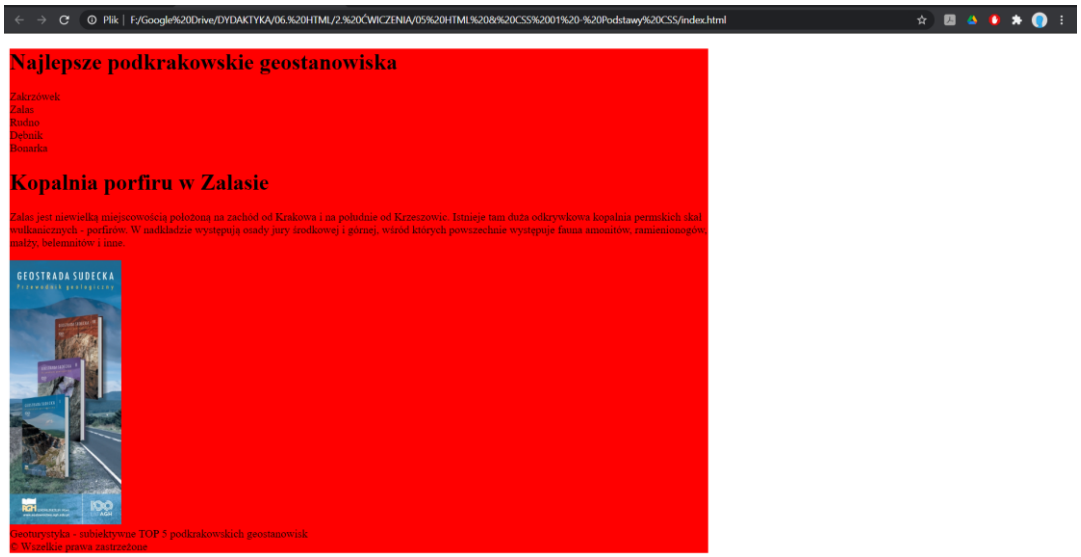
18. Podejrzyjmy wykonaną modyfikację (Ryc. 4). Póki co jedynym efektem jaki uzyskaliśmy jest ograniczenie szerokości bloków z prawej strony.

Ryc. 4. Wygląd strony www po zmianie szerokości bloku kontenera na 1000px

19. Aby lepiej zrozumieć co się wydarzyło tymczasowo dodajmy do deklaracji wyglądu bloku kontenera kolor jego tła.

```
#container{
    width: 1000px;
    background-color: red;
}
```

20. Ponownie obejrzyjmy wygląd strony w przeglądarce www (Ryc. 5).



Ryc. 5. Wygląd strony www po zmianie koloru tła bloku kontenera

Teraz wyraźnie widać, że szerokość bloku kontenera została właściwie ograniczona do 1000px.

21. Zajmijmy się teraz wyśrodkowaniem bloku kontenera względem lewej i prawej krawędzi przeglądarki www. Robimy to za pomocą pary deklaracji autoustawienia lewego i prawego marginesu:

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

```
}
```

22. Po zachowaniu zmian w pliku `style.css`, odświeżmy wygląd strony w przeglądarce www (Ryc. 6).

Ryc. 6. Wygląd strony www po wyśrodkowaniu bloku kontenera

23. Usuńmy z deklaracji wyglądu kontenera wiersz odpowiedzialny za czerwone tło tego bloku. Nie będzie nam już ono potrzebne.

24. Przejdźmy teraz do zdefiniowania wyglądu pierwszego bloku znajdującego się wewnątrz bloku kontenera. Jest to blok o identyfikatorze „`logo`” (`<div id=”logo”></div>`). Na początek nadajmy mu czarne tło.

```
#logo{
    background-color: black;
}
```

25. Odświeżmy wygląd strony w przeglądarce www (Ryc. 7).

Ryc. 7. Wygląd strony www po zmianie koloru tła bloku `logo`

26. Czarne tło bloku o identyfikatorze logo przykryło czarny tekst z nagłówka `<h1></h1>` pliku html i jest on teraz niewidoczny. Zmieńmy zatem jego kolor na biały.

```
#logo{
    background-color: black;
    color: white;
}
```

27. Wyśrodkujmy tekst nagłówka znajdującego się wewnątrz elementu blokowego o identyfikatorze logo.

```
#logo{
    background-color: black;
    color: white;
    text-align: center;
}
```

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

Ryc. 8. Wygląd strony www po zmianie koloru czcionki i wyśrodkowaniu tekstu bloku `logo`

29. Tekst nagłówka bloku logo przylega bezpośrednio do krawędzi bloku. Nie wygląda to dobrze dlatego odsuńmy zawartość bloku od jego krawędzi za pomocą deklaracji szerokości marginesu wewnętrznego `padding`.

```
#logo{
    background-color: black;
    color: white;
    text-align: center;
    padding: 15px;
}
```

30. Odświeżmy wygląd strony w przeglądarce www (Ryc. 9).

Ryc. 9. Wygląd strony www po zmianie szerokości marginesu wewnętrznego bloku `logo`

31. Zajmiemy się teraz elementami blokowymi o identyfikatorach `nav`, `content` i `ad`. Musimy ułożyć je obocznie względem siebie. Na początek dodajmy odpowiednie nagłówki deklaracji do pliku `style.css`.

```
#nav{

}
#content{

}
#ad{

}
```

32. Aby odwrócić domyślne układanie bloków przez przeglądarki www jednego pod drugim, do deklaracji każdego bloku układanego obocznie należy dodać deklarację `float: left;`. Aby efekt był widoczny należy zdefiniować szerokości bloków aby mogły się pomieścić wewnątrz bloku nadrzędnego (w naszym przypadku to blok kontenera o szerokości 1000px). Zacznijmy jednak od nadpisania domyślnego ustawiania bloków przez przeglądarkę jednego pod

> CSS

drugim.

```
#nav{
    float: left;
}
#content{
    float: left;
}
#ad{
    float: left;
}
```

33. Zdefiniujmy teraz szerokości bloków o identyfikatorach `nav`, `content` i `ad` zgodnie z Ryc. 2.

```
#nav{
    float: left;
    width: 120px;
}
#content{
    float: left;
    width: 720px;
}
#ad{
    float: left;
    width: 160px;
}
```

34. Odświeżmy wygląd strony w przeglądarce www (Ryc. 10).

Ryc. 10. Wygląd strony www po zmianach ustawienia i szerokości bloków `nav`, `content` i `ad`

Jak widać tak jak tego chcieliśmy wszystkie zdefiniowane bloki ułożyły się obok siebie. Strona zaczyna wyglądać tak jak tego oczekujemy.

35. Zmieńmy tła bloków na `nav` i `ad` na jasnoszare.

```
#nav{
    float: left;
    width: 120px;
    background-color: lightgray;
}
#content{
    float: left;
    width: 720px;
}
#ad{
    float: left;
    width: 160px;
    background-color: lightgray;
}
```

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

Ryc. 11. Wygląd strony www po zmianach ustawień tła bloków `nav` i `ad`

Jak widać powstaje tu problem polegający na tym, że wysokość bloków „dopasowuje" się do wysokości treści bloków. Jeżeli chcemy aby bloki miały większą wysokość powinniśmy zdefiniować minimalną wysokość tych bloków.

37. Dodajmy minimalną wysokość do deklaracji wyglądu bloków `nav`, `content` i `ad`.

```
#nav{
    float: left;
    width: 120px;
    background-color: lightgray;
    min-height: 420px;
}
#content{
    float: left;
    width: 720px;
    min-height: 420px;
}
#ad{
    float: left;
    width: 160px;
    background-color: lightgray;
    min-height: 420px;
}
```

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

Ryc. 12. Wygląd strony www po zmianach minimalnej wysokości bloków `nav`, `content` i `ad`

39. Treści zamieszczone wewnątrz bloków `nav`, `content` i `ad` brzydko przylegają do krawędzi bloków. Dodajmy odpowiednie marginesy wewnętrzne, które pozwolą odsunąć je od granic. Dla bocznych bloków dodajmy `padding` 10-pikselowy, zaś dla bloku `content` – 20-pikselowy.

```
#nav{
    float: left;
    width: 120px;
    background-color: lightgray;
    min-height: 420px;
    padding: 10px;
}
#content{
    float: left;
    width: 720px;
    min-height: 420px;
    padding: 20px;
}
#ad{
    float: left;
    width: 160px;
```

CSS

```
    background-color: lightgray;
    min-height: 420px;
    padding: 10px;
}
```

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

W efekcie dodania szerokości marginesów wewnętrznych do szerokości bloków:

10___120___10__20_____720_____20__10___160___10

140 + 760 + 180 = 1080

1080 > 1000!

okazało się, że całkowita szerokość obiektów przekroczyła szerokość bloku kontenera, w którym te bloki się znajdują. To spowodowało konieczność przesunięcia bloku ad poniżej bloku `content`. Aby to skorygować musimy odjąć nadwyżkę najlepiej od najszerszego bloku `content`.

> CSS

41. Zmodyfikujmy szerokość bloku `content` aby zmieścić wszystkie szerokości w 1000 pikseli. Aby wszystkie bloki wraz z marginesami zmieściły się w bloku `container` od szerokości bloku `content` musimy odjąć szerokość wszystkich marginesów (łącznie 80 pikseli). Nowa szerokość bloku `content` będzie więc wynosiła 720 – 80 = 640 pikseli.

```
#nav{
    float: left;
    width: 120px;
    background-color: lightgray;
    min-height: 420px;
    padding: 10px;
}
#content{
    float: left;
    width: 640px;
    min-height: 420px;
    padding: 20px;
}
#ad{
    float: left;
    width: 160px;
    background-color: lightgray;
    min-height: 420px;
    padding: 10px;
}
```

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

Ryc. 13. Wygląd strony www po korekcie szerokości bloku `content`

43. Na koniec pozostało wystylizowanie bloku stopki `footer`. Jak widać na Ryc. 13 blok ten częściowo został dołączony do bloku `ad`. Stało się tak z powodu braku odwołania deklaracji `float: left;` zmieniającej domyślne ustawianie bloków pionowo jednego na drugim oraz z powodu różnic w wysokościach bloków `nav` i `ad` (`padding` dolny i górny wynoszą tu po 10 pikseli), a wysokością bloku `content` (`padding` dolny i górny wynosi tutaj po 20 pikseli). Na początek zmniejszmy minimalną wysokość bloku `content` o 20 pikseli.

```
#content{
    float: left;
    width: 640px;
    min-height: 400px;
    padding: 20px;
}
```

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

Ryc. 14. Wygląd strony www po korekcie minimalnej wysokości bloku `content`

45. Dodajmy do pliku styli `style.css` deklarację wyglądu bloku `footer`.

```
#footer{


}
```

46. Odwołajmy teraz deklarację `float: left;` aby przeglądarki wiedziały, że od tego momentu maja interpretować ułożenie bloków w sposób domyślny, a więc jeden na drugim. Służy do tego deklaracja `clear: both;`.

```
#footer{
    clear: both;
}
```

47. Dodajmy do deklaracji bloku o identyfikatorze footer czarne tło, białą czcionkę, wyśrodkowanie oraz margines wewnętrzny.

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

48. Odświeżmy wygląd strony w przeglądarce www (Ryc. 15).

Ryc. 15. Wygląd strony www po dodaniu właściwości bloku `footer`

Tym sposobem zakończyliśmy projekt szablonu prostej strony www opartej o model blokowy. 😊