

**Arkadiusz Liber**  
**Emil Dziubecki**

# **Technologie Internetowe Laboratorium**

**Ćwiczenie 3.  
Projektowanie  
i implementacja dokumentów  
hipertekstowych  
zawierających ramki i tabele**

**Legnica 2005**

# 1. Wymagania wstępne

Laboratorium z przedmiotu „Technologie Internetowe” stanowi uzupełnienie i poszerzenie materiału przedstawionego na wykładzie. W ramach zajęć laboratoryjnych wykonywane są kolejno ćwiczenia opisane w dalszej części podręcznika. Obecność na wszystkich zajęciach laboratoryjnych jest obowiązkowa. Ćwiczenia wykonywane są przez studentów indywidualnie. Każdy student musi wykonać łącznie sześć ćwiczeń laboratoryjnych. Prowadzący laboratorium wskazuje zadania do realizacji przez studenta w ramach wykonywanego ćwiczenia.

## 1.1. Sprawozdania z ćwiczeń

Z każdego ćwiczenia student zobowiązany jest wykonać sprawozdanie pisemne. Sprawozdanie z wykonanego ćwiczenia musi być oddane prowadzącemu przed przystąpieniem do ćwiczenia następnego. Sprawozdanie z ćwiczenia powinno być wykonane według ustalonego wzoru.

Pierwszą stroną sprawozdania jest strona tytułowa. Na stronie tytułowej musi się znajdować nazwa uczelni, nazwa wydziału, imię i nazwisko studenta, numer indeksu, numer grupy studenckiej, tytuł realizowanego ćwiczenia, data wykonania ćwiczenia, miejscowość. Po stronie tytułowej następuje opisowa część sprawozdania.

Część opisowa powinna być podzielona na cztery części:

- 1) cel ćwiczenia i zadania do realizacji,
- 2) opis przebiegu ćwiczenia,
- 3) wnioski,
- 4) załączniki do sprawozdania.

W części 1) powinien być zawarty cel ćwiczenia oraz wyszczególnione zadania do wykonania w ramach ćwiczenia. W części tej należy dokładnie wskazać, które zadania zostały wybrane przez prowadzącego a które stanowią jego uzupełnienie. W przypadku, gdy w trakcie realizacji ćwiczenia zostały postawione dodatkowe zadania należy je wyszczególnić.

W części 2) należy dokładnie opisać cały przebieg ćwiczenia. W opisie tym, w szczególności, powinny być zwarte:

- informacje na temat sposobu wykonania ćwiczenia,
- zastosowane rozwiązania,
- informacje na temat użytych narzędzi informatycznych,
- trudności, na które napotkano w trakcie realizacji zadań,
- ilustracje przedstawiające rezultaty otrzymane w wyniku realizacji ćwiczenia,
- inne informacje, związane z wykonaniem ćwiczenia.

W części 3) należy zebrać i podsumować uzyskane w wyniku realizacji ćwiczenia rezultaty. Wnioski powinny być rzeczowe i ściśle związane z tematyką realizowanego ćwiczenia.

Część 4) przeznaczona jest na umieszczenie dodatkowych materiałów związanych z realizacją ćwiczenia. Materiałami takimi mogą być protokoły odbioru na zajęciach wykonanej przez studenta pracy, szkice i ilustracje wykonane odręcznie w trakcie przygotowania się do realizacji i wykonania ćwiczenia, wydruki i kserokopie materiałów dodatkowych mających szczególne znaczenie dla sposobu realizacji ćwiczenia oraz inne materiały uzupełniające. W załącznikach nie należy umieszczać kserokopii ogólnodostępnych książek i podręczników. Na tego typu materiały należy powoływać się za pomocą odnośników literaturowych i przypisów.

Typografia sprawozdań powinna być jednolita w całej grupie laboratoryjnej. Zaleca się stosowanie formatu papieru A4, podstawowej czcionki w części opisowej Times New Roman 12 pkt z pojedynczym odstępem między liniami tekstu, marginesów 2,5 cm, pogrubienia tytułów rozdziałów oraz wyjustowania tekstu pomiędzy marginesami.

## 1.2. Warunki zaliczenia przedmiotu

Warunki zaliczenia przedmiotu są jednakowe dla wszystkich studentów. W przypadku, gdy prowadzący nie poda innych warunków zaliczania ćwiczeń – obowiązują warunki wyszczególnione poniżej.

Obecność na wszystkich ćwiczeniach jest obowiązkowa. W przypadku usprawiedliwionej nieobecności lub niezaliczenia ćwiczenia student wykonuje ćwiczenie w terminie odróbkowym. Jeśli prowadzący nie ustali inaczej na realizację każdego ćwiczenia laboratoryjnego przypadają 4 godziny zajęć.

Do zajęć student powinien być przygotowany. Prowadzący może sprawdzić wiedzę studenta przed przystąpieniem do ćwiczenia lub w trakcie wykonywania ćwiczenia. W przypadku, gdy student nie jest przygotowany do zajęć, prowadzący może przerwać wykonywanie ćwiczenia i nakazać wykonanie ćwiczenia w terminie poprawkowym.

W trakcie realizacji ćwiczenia student sporządza protokół wykonania ćwiczenia lub bezpośrednio sprawozdanie.

Ocenianie każdego ćwiczenia odbywa się w systemie punktowym. Punkty przydzielane są za wykonanie ćwiczenia w trakcie trwania laboratorium oraz za wykonanie sprawozdania. Ocena za pojedyncze ćwiczenie jest sumą uzyskanych punktów za obie te części. Ocena końcowa wynika bezpośrednio z sumy uzyskanych punktów. Jeśli na początku semestru prowadzący nie ustali inaczej obowiązuje następujący przydział punktów za pojedyncze ćwiczenie: od 0 do 10 punktów za część realizowaną w laboratorium i od 0 do 10 punktów za sprawozdanie, co daje łącznie 20 punktów za jedno ćwiczenie. Jeśli prowadzący na początku semestru nie ustali inaczej obowiązują oceny końcowe wynikające z tabeli zamieszczonej poniżej.

Tabela 1. Przykładowa tabela przelicznikowa całkowitej ilości punktów zgromadzonych na laboratorium na ocenę końcową

L.p.	Całkowita ilość punktów	Ocena	Ocena słownie
1	poniżej 60	2	niedostateczny
2	60 – 70	3	dostateczny
3	71 – 80	3,5	dostateczny plus
4	81 – 90	4	dobry
5	91 – 100	4,5	dobry plus
6	powyżej 100	5	bardzo dobry
7	powyżej 118 punktów, pod warunkiem wykazania się przez studenta wiadomościami wykraczającymi poza przewidziane w programie przedmiotu	5,5	celujący

## **2. Projektowanie i implementacja dokumentów hipertekstowych zawierających ramki i tabele**

### **2.1. Cel ćwiczenia**

Celem ćwiczenia jest zapoznanie studentów ze sposobami implementacji ramek, tabel i formularzy w dokumentach hipertekstowych. Do tworzenia dokumentów hipertekstowych w ramach ćwiczenia wykorzystano język znaczników HTML.

### **2.2. Dokumentacja przebiegu ćwiczenia i otrzymanych rezultatów**

Ćwiczenie należy wykonać zgodnie ze wskazówkami prowadzącego. Wykonanie każdego zadania należy zdokumentować (sporządzić protokół, zrzuty ekranów itp.), następnie zgłosić prowadzącemu w celu sprawdzenia poprawności jego wykonania. Przebieg ćwiczenia należy tak dokumentować, aby możliwe było sporządzenie sprawozdania szczegółowo opisującego przebieg ćwiczenia i otrzymane rezultaty. W celu uniknięcia nieporozumień, o do zakresu zrealizowanych na ćwiczeniu zadań należy sporządzić protokół i dać go do podpisu prowadzącemu. Sporządzony protokół należy umieścić w sprawozdaniu w postaci załącznika. Brak podpisanego protokołu może być potraktowany jako świadectwo niewykonania ćwiczenia na zajęciach laboratoryjnych i skutkować niezaliczeniem danego ćwiczenia.

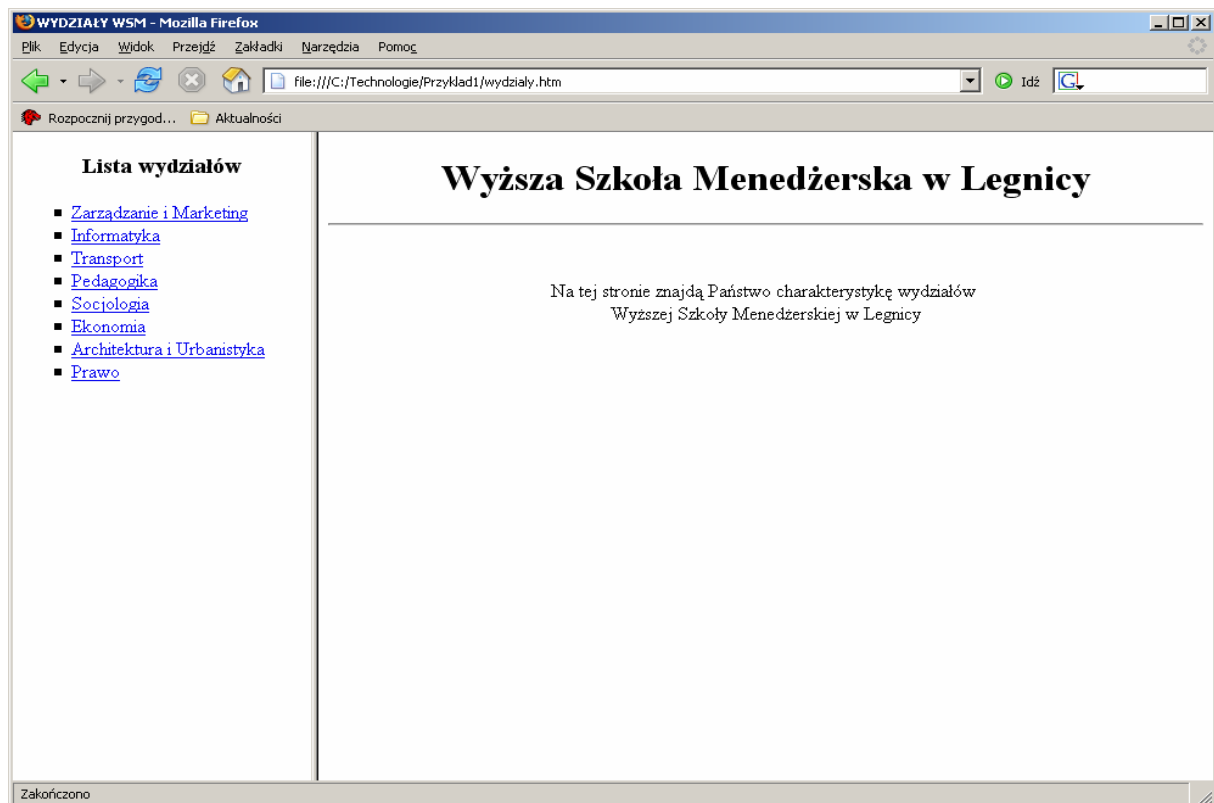
### **2.3. Ramki w dokumentach hipertekstowych**

Jednym ze sposobów formatowania dokumentu hipertekstowego jest wprowadzenie tzw. ramek. Ramki umożliwiają podział treści na ściśle określone bloki o organizacji zbliżonej do organizacji tabel. Pełnią one następujące funkcje: porządkowanie treści, organizacja rozkładu treści w ramach strony, elastyczne łączenie z niezależnymi dokumentami hipertekstowymi stanowiącymi dokumenty podrzędne w strukturze ramkowej.

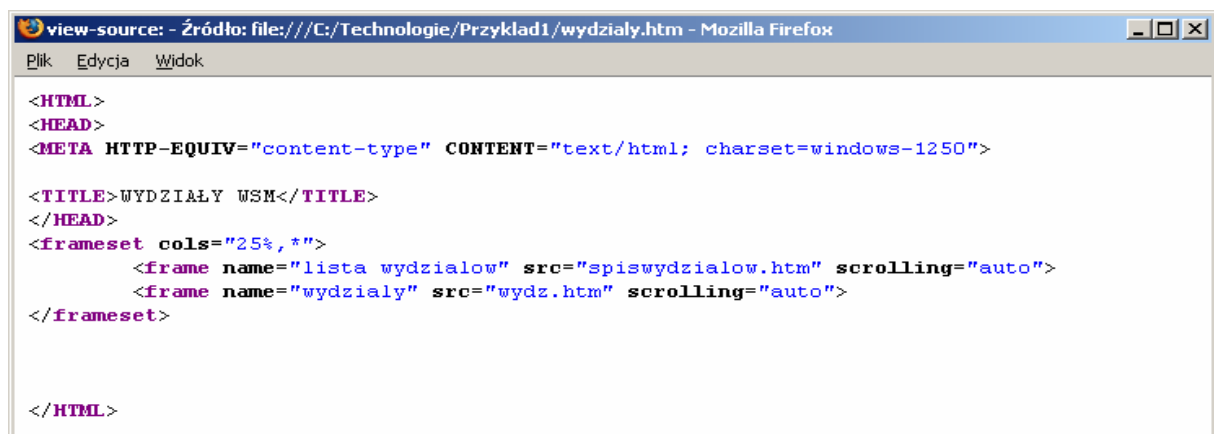
Konstruując stronę zawierającą ramki należy wziąć pod uwagę możliwości maszyny interpretującej hipertekst. W przypadku, gdy maszyna interpretująca nie obsługuje ramek należy poprawnie skonstruować specjalną część dokumentu ograniczoną znacznikami `<NOFRAMES>` i `</NOFRAMES>`. W przypadku, gdy w dokumencie hipertekstowym występuje część definiująca zachowanie dokumentu bez ramek para znaczników `<BODY>` i `</BODY>` może wystąpić tylko w części ograniczonej znacznikami `<NOFRAMES>` i `</NOFRAMES>`.

Na rysunku 1 przedstawiono przykład wykorzystania ramek do podziału treści. Na rysunku 2 przedstawiono kod źródłowy strony, której widok znajduje się na rysunku 1.

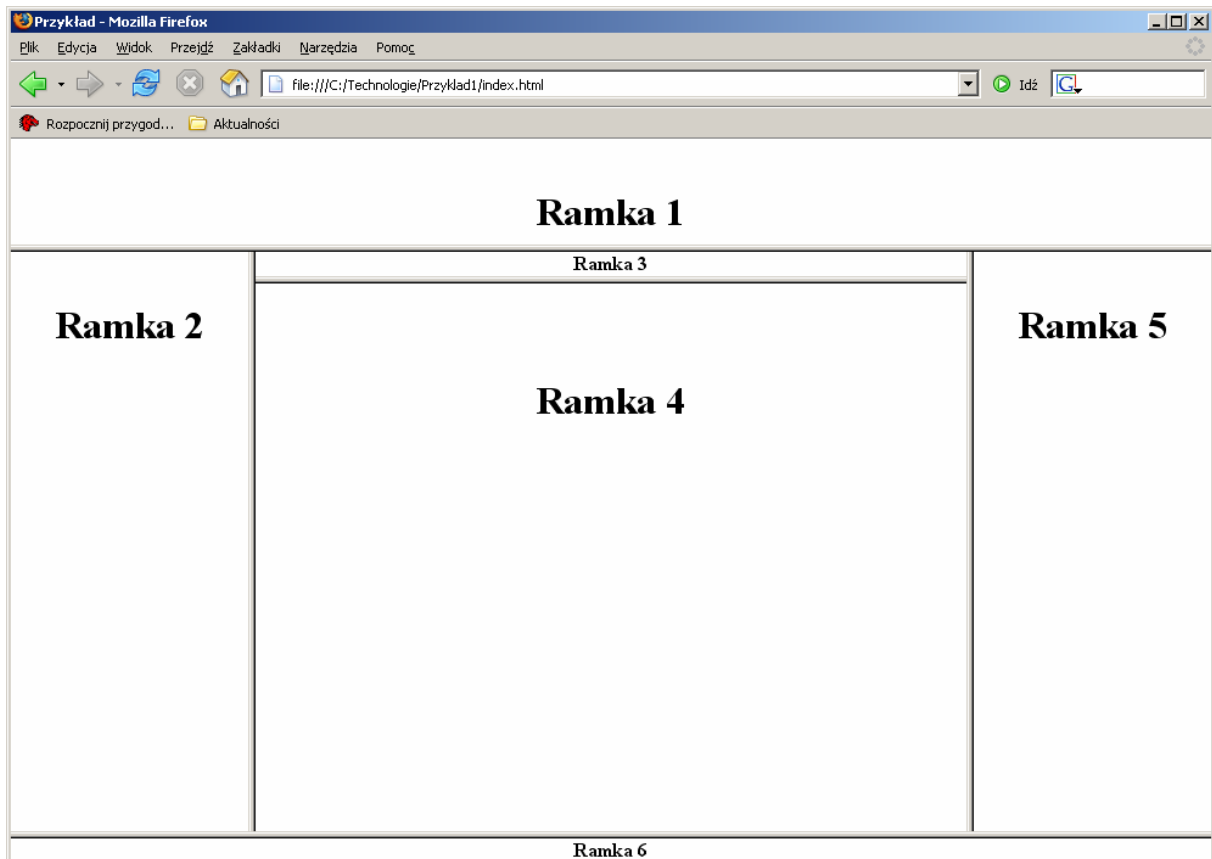
Na rysunku 3 przedstawiono widok dokumentu hipertekstowego, w którym zastosowano podział na 6 części. Do podziału treści zastosowano ramki. Na rysunku 4 przedstawiono kod źródłowy tego dokumentu.



Rysunek 1. Widok dokumentu hipertekstowego sformatowanego przy wykorzystaniu ramek



Rysunek 2. Kod źródłowy strony przedstawionej na rysunku 1



Rysunek 3. Widok przykładowego dokumentu hipertekstowego podzielonego na 6 części przy użyciu ramek

```

view-source: - Źródło: file:///C:/Technologie/Przyklad1/index.html - Mozilla Firefox
Plik  Edycja  Widok

<HTML>
<HEAD>
<META HTTP-EQUIV="content-type" CONTENT="text/html; charset=windows-1250">
<meta http-equiv="author" content="Autor_strony">

<TITLE>Przykład</TITLE>
</HEAD>

<frameset rows=15%,*,3% frameborder=1 framespacing=0>
  <frame marginheight=0 marginwidth=0 name=ramka1 scrolling=no noresize src="ram1.htm">
  <frameset cols="20%,60%,20%" frameborder=1 framespacing=0>
    <frame marginheight=0 marginwidth=0 name=ramka2 scrolling=no src="ram2.htm" >
    <frameset rows=20,* frameborder=1 framespacing=0>
      <frame marginheight=0 marginwidth=0 name=ramka3 scrolling=no noresize src="ram3.htm" >
      <frame marginheight=0 marginwidth=0 name=ramka4 scrolling=yes noresize src="ram4.htm" >
    </frameset>
    <frame marginheight=0 marginwidth=0 name=ramka5 scrolling=no noresize src="ram5.htm" >
  </frameset>
  <frame marginheight=0 marginwidth=0 name=ramka6 scrolling=no noresize src="ram6.htm">
</frameset>

<noframes>
<body>
<font color="#FFFFFF">Twoja przeglądarka nie akceptuje ramek</font>

</body>
</noframes>
</HTML>

```

Rysunek 4. Kod źródłowy dokumentu hipertekstowego przedstawionego na rysunku 3

## 2.4. Definiowanie i używanie tabel w dokumentach hipertekstowych

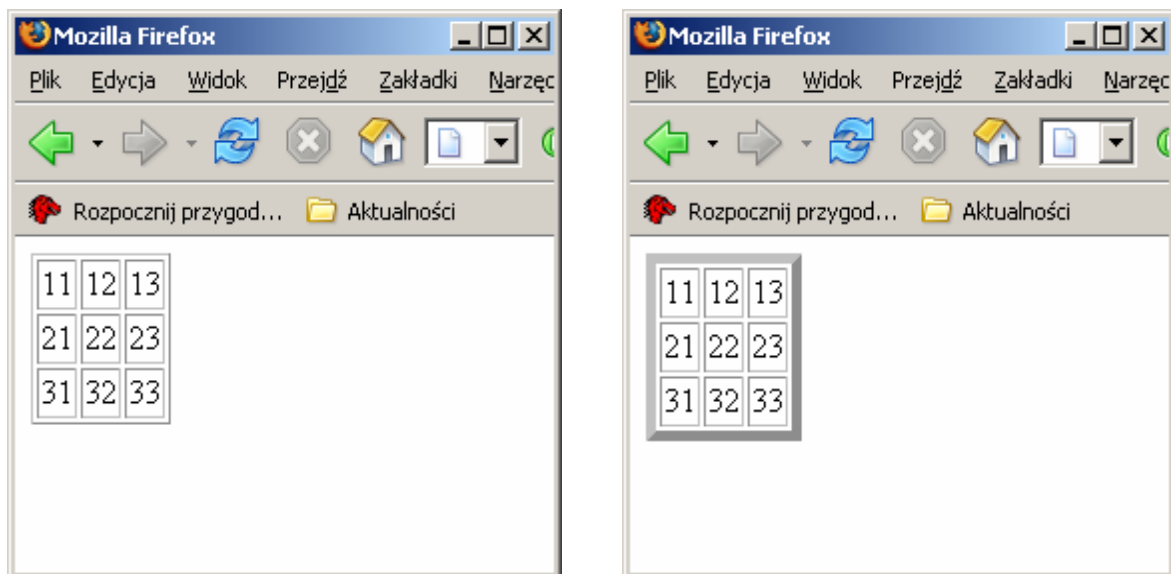
Jedną z najbardziej złożonych struktur w dokumentach hipertekstowych są tabele. Tabele wykorzystywane są nie tylko do prezentacji zestawów danych na stronach, lecz również są ważnym środkiem wyrazu graficznego dokumentów hipertekstowych. W zaawansowanych aplikacjach tabele są łącznikiem, pomiędzy dokumentem hipertekstowym a bazami danych.

Definiowanie tabeli rozpoczyna się od wprowadzenia znaczników: `<TABLE></TABLE>`. Pomiedzy tymi znacznikami umieszcza się między innymi: definicje wierszy, definicje komórek w wierszach, konkretne dane w komórkach, tytuł tabeli i nagłówki wierszy i kolumn. Podstawową częścią tabeli jest wiersz. Pomiedzy znacznikami definiującymi tabelę `<TABLE> </TABLE>` można umieścić wiele kolejnych definicji wierszy `<TR> </TR>`. Aby umieścić komórkę w wierszu należy zastosować parę znaczników `<TD> </TD>`. Między tymi znacznikami można wprowadzać tekst i grafikę. Poniżej przedstawiono przykładowy kod źródłowy definiujący tabelę:

```
<TABLE>
<TR><TD> </TD><TD> </TD><TD> </TD></TR>
<TR><TD> </TD><TD> </TD><TD> </TD></TR>
<TR><TD> </TD><TD> </TD><TD> </TD></TR>
</TABLE>
```

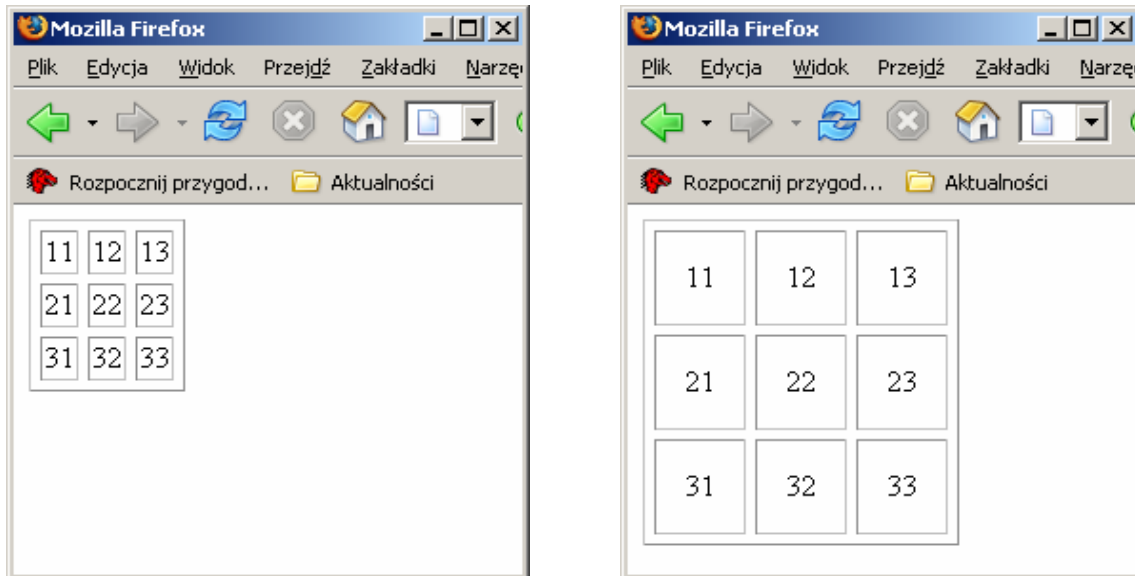
Ze względu na to, iż definicje tabel mogą być bardzo rozbudowane definicje wierszy tabeli powinno umieszczać się jedną pod drugą, natomiast definicje kolejnych komórek obok siebie. Umieszczenie definicji komórek w kolejnych wierszach kodu źródłowego nie wpływa na ich faktyczne położenie wyznaczone przez interpreter na podstawie definicji wierszy.

Tabela może posiadać obramowanie. W celu uzyskania obramowania definicję tabeli wystarczy rozszerzyć o znacznik `BORDER`. W przypadku, gdy nie zostaną podane parametry wewnątrz obszaru definicyjnego znacznika `BORDER` przyjmowana jest domyślny wygląd obramowania tabeli. Na rysunku 5 przedstawiono widok dokumentów hipertekstowych z różnymi definicjami obramowania tabel.



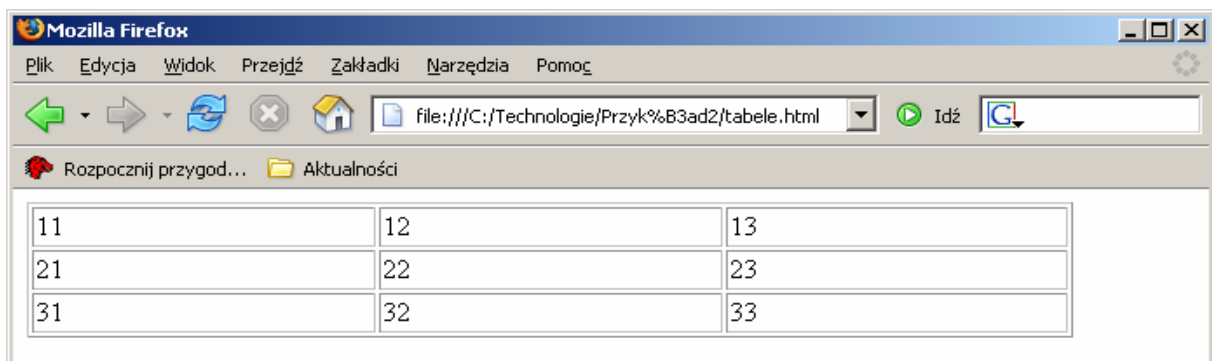
Rysunek 5. Widok tabel z różnym formatowaniem obramowania a) `<TABLE BORDER></TABLE>`, b) `<TABLE BORDER=5></TABLE>`

Dla zwiększenia przejrzystości wyświetlania danych zgromadzonych w tabeli określa się tzw. odległości wewnątrzkomórkowe. Do ustalenia tych odległości wykorzystywane są parametry: CELSPACING oraz CELLPADDING. Na rysunku 6 przedstawiono widok dokumentu hipertekstowego ze zmienionymi wartościami parametrów CELSPACING oraz CELLPADDING.



Rysunek 6. Widok dokumentu hipertekstowego zawierającego tabelę sformatowaną dla różnych wartości parametrów CELSPACING oraz CELLPADDING: a) `<TABLE BORDER CELSPACING=5>` `</TABLE>`, b) `<TABLE BORDER CELSPACING=5 CELLPADDING=15>` `</TABLE>`

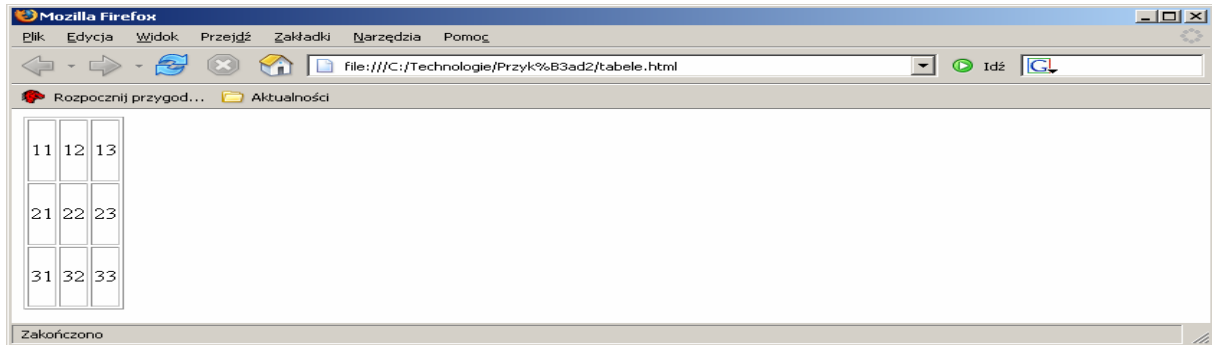
Do sterowania szerokością tabeli służy parametr WIDTH. Do definicji wysokości tabeli służy parametr HEIGHT. Na rysunku 7 przedstawiono widok tabeli z ustawioną szerokością. Szerokość przykładowej tabeli na rysunku 7 ustawiona została w punktach ekranu.



Rysunek 7. Widok tabeli z ustaloną szerokością (`<TABLE BORDER WIDTH=600>` `</TABLE>`)

Na rysunku 8 przedstawiono widok dokumentu hipertekstowego zawierającego tabelę z zadaną w kodzie źródłowym wysokością. Wysokość przykładowej tabeli ustawiona została w punktach ekranu.





Rysunek 8. Widok dokumentu hipertekstowego zawierającego tabelę o wysokości ustawionej przy wykorzystaniu parametru HEIGHT (<TABLE BORDER HEIGHT=200> </TABLE>)

Ustalanie szerokości i wysokości tabel w punktach ekranowych jest dość niezręczne, gdyż widok tabeli w znacznym stopniu będzie ulegał zmianie wraz ze zmianą rozdzielczości ekranu oraz parametrów graficznych urządzenia wyświetlającego. Dużo lepszą metodą jest ustalanie wysokości i szerokości tabeli w skali względnej. Jediną zmianą w kodzie źródłowym będzie wtedy wpisanie wartości parametrów zakończonych znakiem %. Przykładowo, aby uzyskać tabelę o szerokości 50% ekranu należy wprowadzić sekwencję znaczników:

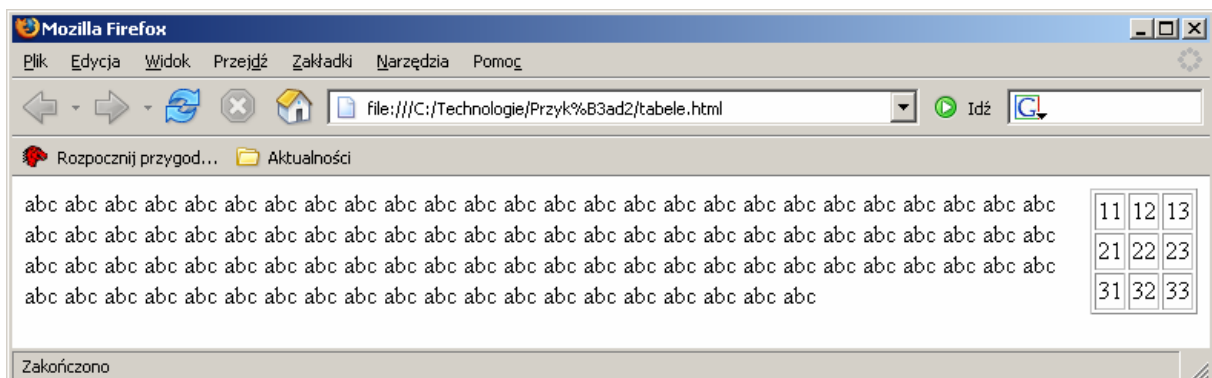
<TABLE BORDER WIDTH=50%> </TABLE>

Aby uzyskać tabelę o wysokości 50% ekranu należy wprowadzić sekwencję znaczników:

<TABLE BORDER HEIGHT=50%> </TABLE>

Możliwe jest zastosowanie wyrównania tabeli względem marginesów dokumentu. W tym celu używa się parametru ALIGN. Na rysunku 9 przedstawiono widok dokumentu hipertekstowego zawierającego tabelę dosuniętą do prawego marginesu. Poniżej przedstawiono przykładową sekwencję znaczników realizujących dosunięcie tabeli do prawego marginesu:

<TABLE BORDER ALIGN=right> </TABLE>

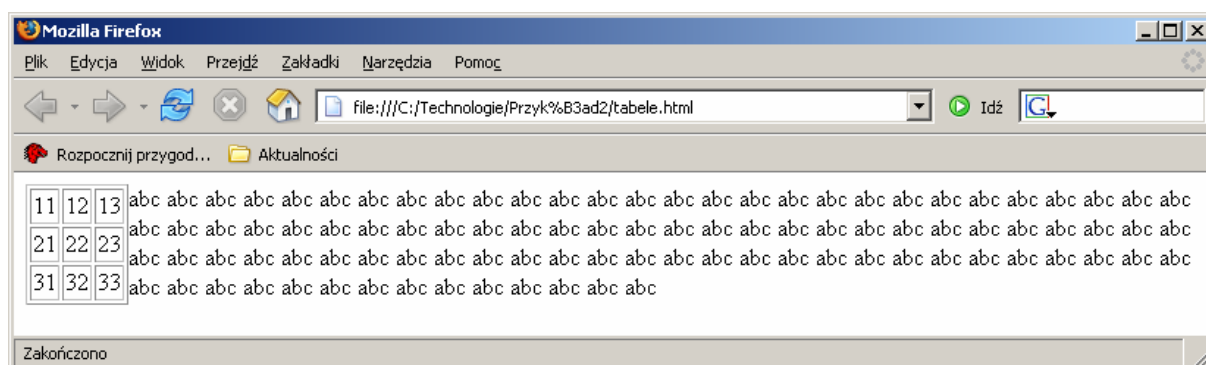


Rysunek 9. Widok tabeli umieszczonej w dokumencie hipertekstowym dosuniętej do prawego marginesu (ALIGN=right)

Na rysunku 10 przedstawiono wynik działania sekwencji:

<TABLE BORDER ALIGN=left> </TABLE>

realizującej dosunięcie tabeli do lewego marginesu.

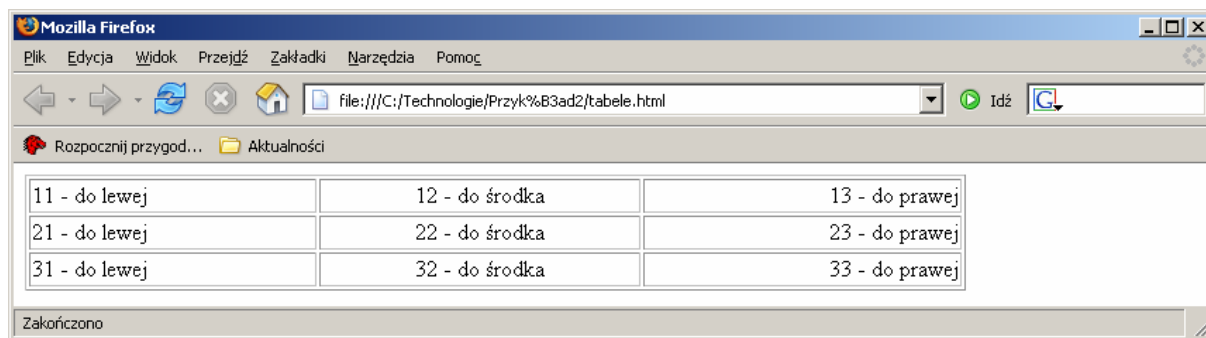


Rysunek10. Widok dokumentu hipertekstowego zawierającego tabelę dosuniętą do lewego marginesu (ALIGN=left)

Tekst w dokumencie hipertekstowym wstawiony za definicją tabeli - otacza ją z prawej lub lewej strony, w zależności od wartości parametru ALIGN. Parametr ALIGN może zostać użyty do poziomego wyrównywania zawartości komórek w tabeli. Poniżej przedstawiono fragment kodu źródłowego realizującego różne wyrównanie zawartości wewnątrz komórek tabeli:

```
<TD ALIGN=center> </TD>  
<TD ALIGN=left> </TD>  
<TD ALIGN=right> </TD>
```

Na rysunku 11 przedstawiono widok dokumentu hipertekstowego zawierającego tabelę z różnym formatowaniem położenia zawartości komórek.

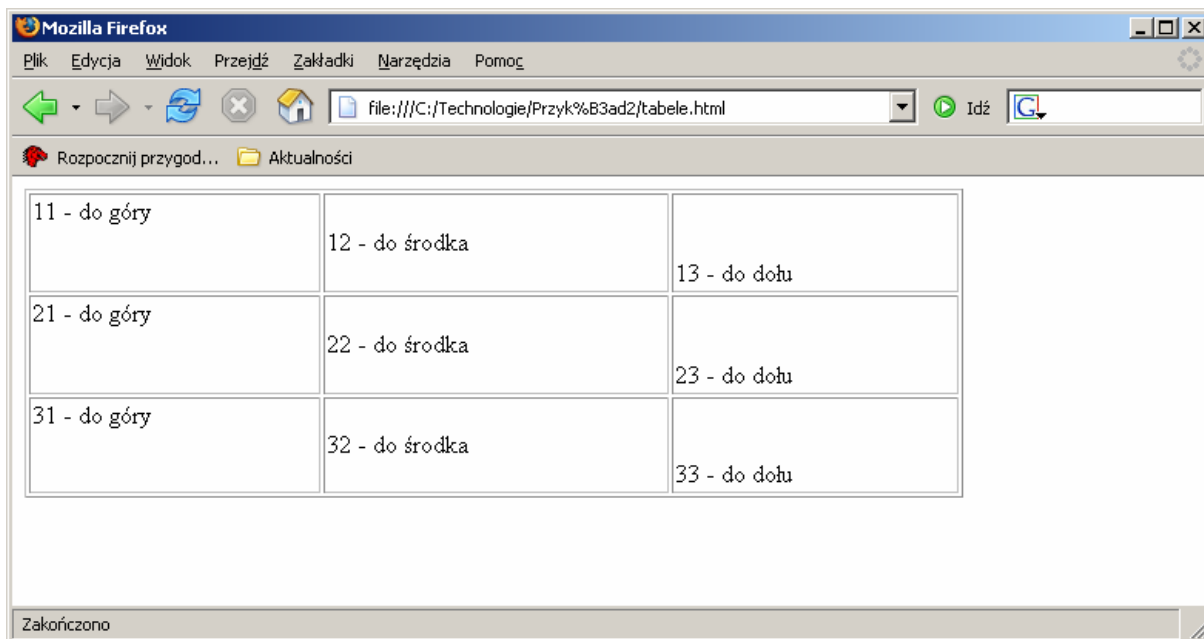


Rysunek 11. Widok dokumentu hipertekstowego zawierającego tabelę z komórkami sformatowanymi z różnymi wartościami parametru ALIGN

Do formatowania pionowego wyrównania zawartości komórek w tabeli służy parametr VALIGN. Możliwe jest wyrównanie do: górnego brzegu, środka, dolnego brzegu. Poniżej przedstawiono fragmenty kodu źródłowego realizujące wyrównanie zawartości komórek w pionie:

```
<TD VALIGN=top> </TD>  
<TD VALIGN=middle> </TD>  
<TD VALIGN=bottom> </TD>
```

Na rysunku 12 przedstawiono widok dokumentu hipertekstowego zawierającego tabelę z różnym formatowaniem zawartości komórek w pionie.

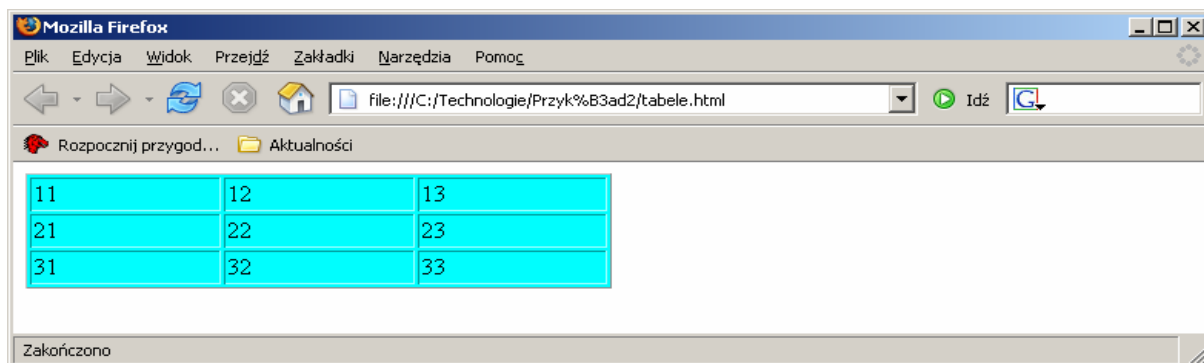


Rysunek 12. Widok dokumentu hipertekstowego zawierającego tabelę z różnym formatowaniem zawartości komórek w pionie (VALIGN=top, VALIGN=middle, VALIGN=bottom)

Do zmiany koloru tła komórek w tabelach służy parametr BGCOLOR. Przykładowo do ustawienia jasnoniebieskiego tła komórek w tabeli można użyć sekwencji:

**<TABLE BORDER HEIGHT=300 BGCOLOR=aqua>**

Na rysunku 13 przedstawiono rezultat zastosowania zmiany tła komórek tabeli w dokumencie hipertekstowym.

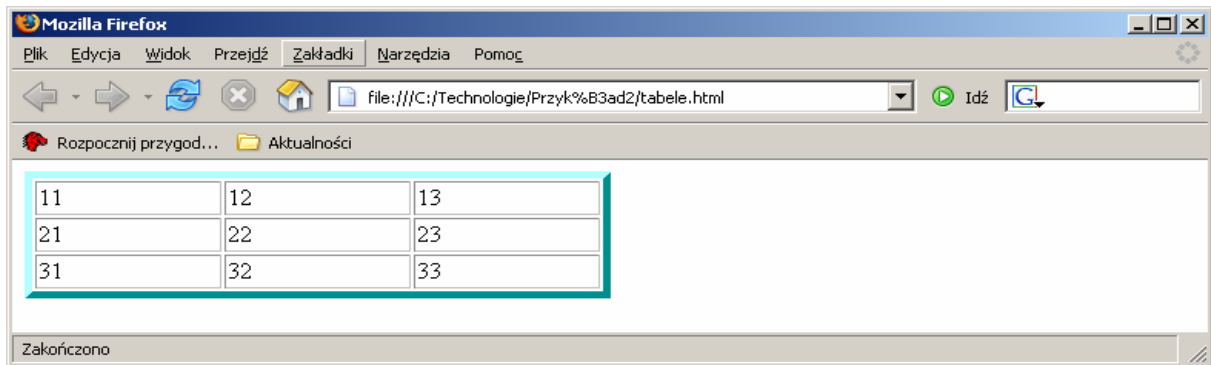


Rysunek 13. Widok dokumentu hipertekstowego zawierającego tabelę ze zmienionym kolorem tła komórek

Jako uzupełnienie formatowania kolorów można wykorzystać formatowanie obramowania tabeli. Do zmiany koloru obramowania tabeli służy parametr BORDERCOLOR. Poniżej przedstawiono fragment kodu źródłowego dokumentu ustwiający kolor obramowania tabeli na niebieski:

**<TABLE BORDER=5 BORDERCOLOR=blue>**

Na rysunku 14 przedstawiono widok dokumentu hipertekstowego zawierającego tabelę z obramowaniem wykonanym w kolorze niebieskim.



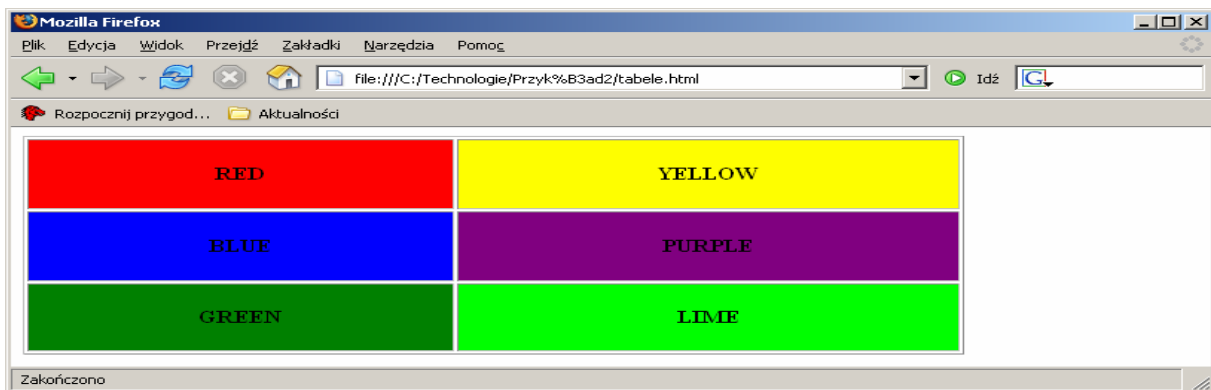
Rysunek 14. Widok dokumentu hipertekstowego zawierającego tabelę z niebieskim obramowaniem ustawionym przy użyciu parametru BORDERCOLOR

W celu ustawienia indywidualnych kolorów komórek tabeli należy parametr BGCOLOR umieścić wewnątrz obszaru definicyjnego komórki ograniczonego znacznikami <TD> oraz </TD>. Definicję koloru komórki wstawia się w postaci sekwencji:

**<TD BGCOLOR="kolor"> </TD>**

gdzie "kolor" jest nazwą koloru lub szesnastkowym oznaczeniem koloru.

Na rysunku 15 przedstawiono widok dokumentu hipertekstowego zawierającego tabelę z komórkami o różnym kolorze tła.

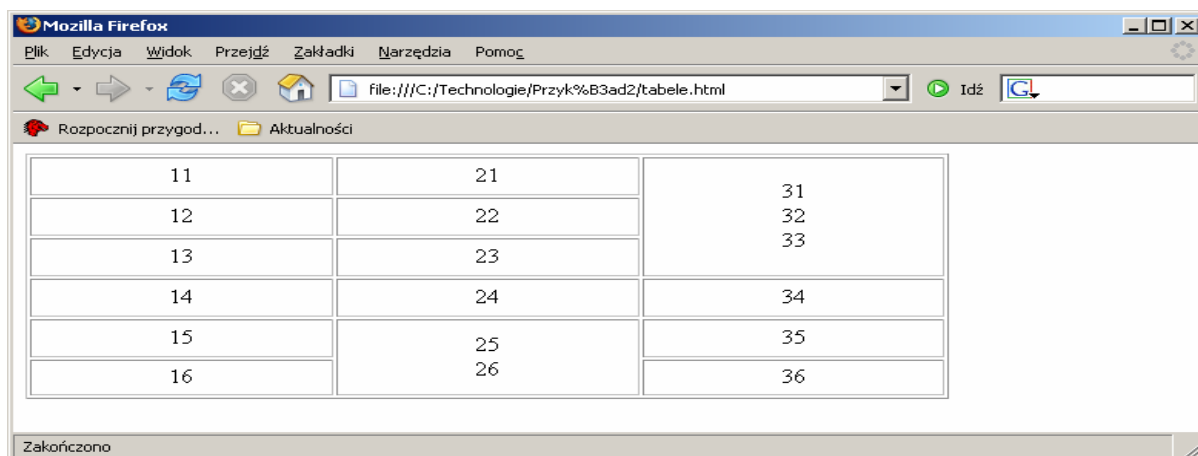


Rysunek 15. Widok dokumentu hipertekstowego zawierającego tabelę z komórkami o różnym kolorze tła

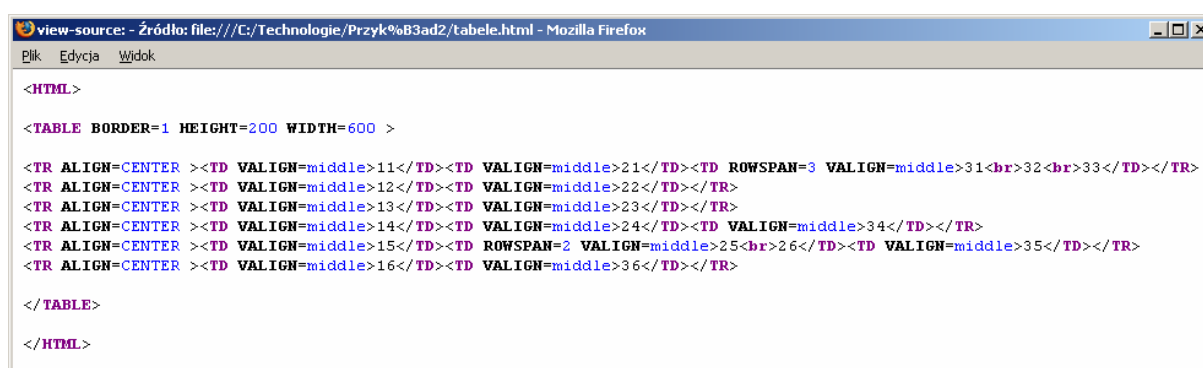
Komórki w tabelach można scalać. Do scalania komórek służą parametry ROWSPAN i COLSPAN. Przykładowo, w celu rozciągnięcia zawartości komórki na  $x$  wierszy należy wprowadzić znacznik TD z parametrem ROWSPAN. Przykładowy kod źródłowy realizujący scalenie dla 3 wierszy przedstawiono poniżej:

**<TD ROWSPAN=3>**

Na rysunku 16 przedstawiono widok dokumentu hipertekstowego zawierającego tabelę ze złączonymi komórkami dla różnych zakresów wierszy. Na rysunku 17 przedstawiono widok kodu źródłowego dokumentu przedstawionego na rysunku 16.



Rysunek 16. Widok dokumentu hipertekstowego zawierającego komórki złączone wierszami

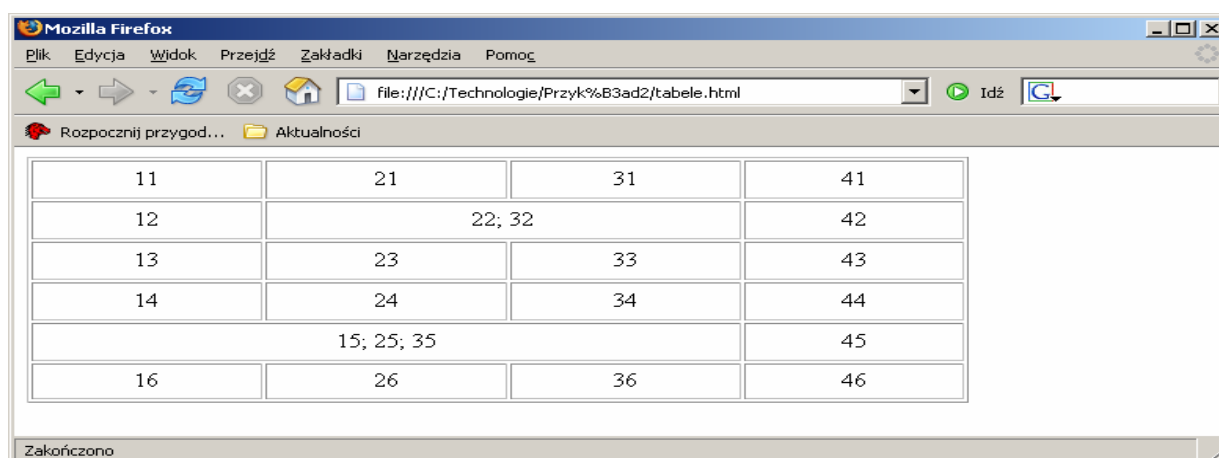


Rysunek 17. Widok kodu źródłowego dokumentu hipertekstowego z rysunku 16

Do złączenia komórek w kolumnach używa się parametru COLSPAN. Przykładowo dla złączenia komórek w trzech kolumnach należy wprowadzić sekwencję źródłową:

**<TD COLSPAN=3>**

Na rysunku 18 przedstawiono wygląd dokumentu hipertekstowego zawierającego tabelę z komórkami złączonymi kolumnowo. Na rysunku 19 przedstawiono kod źródłowy dokumentu z rysunku 18.



Rysunek 18. Wygląd tabeli, w której zastosowano w definicjach wybranych komórek parametr COLSPAN

```

<HTML>
<TABLE BORDER=1 HEIGHT=200 WIDTH=600 >
<TR ALIGN=CENTER ><TD VALIGN=middle>11</TD><TD VALIGN=middle>21</TD><TD VALIGN=middle>31</TD><TD VALIGN=middle>41</TD></TR>
<TR ALIGN=CENTER ><TD VALIGN=middle>12</TD><TD COLSPAN=2 VALIGN=middle>22; 32</TD><TD VALIGN=middle>42</TD></TR>
<TR ALIGN=CENTER ><TD VALIGN=middle>13</TD><TD VALIGN=middle>23</TD><TD VALIGN=middle>33</TD><TD VALIGN=middle>43</TD></TR>
<TR ALIGN=CENTER ><TD VALIGN=middle>14</TD><TD VALIGN=middle>24</TD><TD VALIGN=middle>34</TD><TD VALIGN=middle>44</TD></TR>
<TR ALIGN=CENTER ><TD COLSPAN=3 VALIGN=middle>15; 25; 35</TD><TD VALIGN=middle>45</TD></TR>
<TR ALIGN=CENTER ><TD VALIGN=middle>16</TD><TD VALIGN=middle>26</TD><TD VALIGN=middle>36</TD><TD VALIGN=middle>46</TD></TR>
</TABLE>
</HTML>

```

Rysunek 19. Widok kodu źródłowego dokumentu przedstawionego na rysunku 18

## 2.5. Formularze w dokumentach hipertekstowych

Bardzo ważnym elementem dokumentu hipertekstowego jest tzw. formularz. Formularz służy do komunikacji odbiorcy dokumentu z twórcą dokumentu, firmą udostępniającą serwis itp. Formularze tworzone są w obszarze pomiędzy znacznikami <FORM> oraz </FORM>. Pomędzy tymi znacznikami umieszcza się elementy składowe formularza. Charakterystyczną własnością obszaru formularza jest możliwość zdefiniowania tzw. akcji. Akcja najczęściej związana jest z przekazywaniem danych do odbiorcy. Wartością parametru ACTION jest tzw. cel, czyli adres docelowy wysyłania lub pobierania danych. Z akcją związana jest metoda. Do dyspozycji są dwie metody: GET i POST. Metoda GET służy do pobierania danych z serwera. Metoda POST służy do wysyłania danych. Dane najczęściej wysyłane są przez wskazanie jako źródła adresu pocztowego. Przykładowe odwołania z poziomu formularza przedstawiono poniżej:

```
<FORM ACTION="mailto:webmaster@wsm.edu.pl" METHOD="Post">
```

```
<FORM ACTION="aktualnosci.htm" METHOD="Post">
```

Bardzo ważnym elementem formularza jest możliwość wprowadzania danych. Wprowadzanie danych realizuje się przy wykorzystaniu znacznika INPUT. Znacznik ten ma następującą składnię:

```
<INPUT TYPE="typ" NAME="nazwa" VALUE="wartosc" MAXLENGTH=x
SIZE=x >
```

gdzie:

**TYPE** – określa rodzaj pola (brak-domyślne pole tekstowe),

**NAME** – nazwa pola,

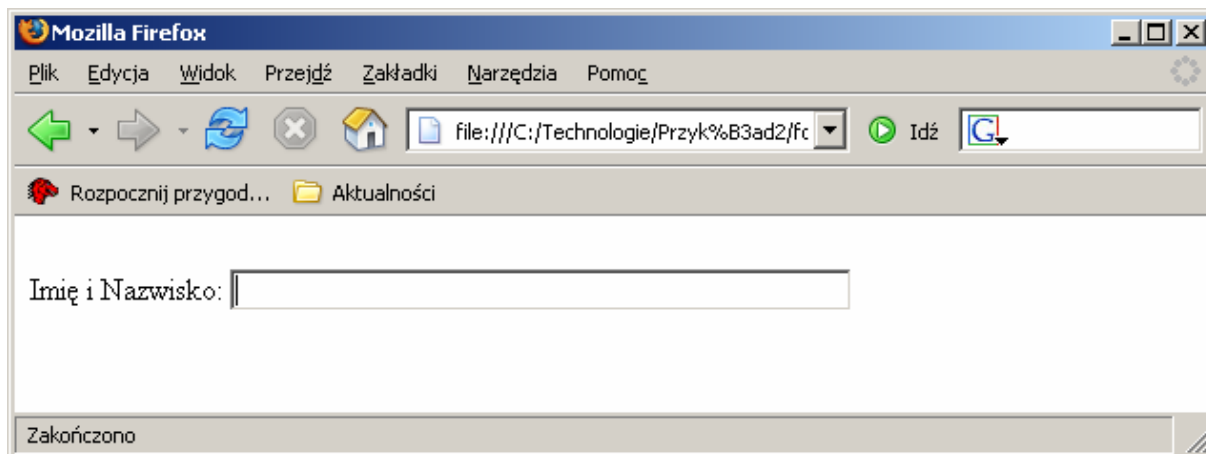
**VALUE** – wartość pola,

**MAXLENGTH=x** - ogranicza długość wpisywanej informacji do n znaków,

**SIZE=x** – definiuje długość pola edycyjnego,

Na rysunku 20 przedstawiono widok dokumentu hipertekstowego zawierającego wprowadzanie danych zdefiniowane przy użyciu znacznika INPUT:

```
Imię i Nazwisko: <INPUT NAME="nazwisko" SIZE=50 >
```

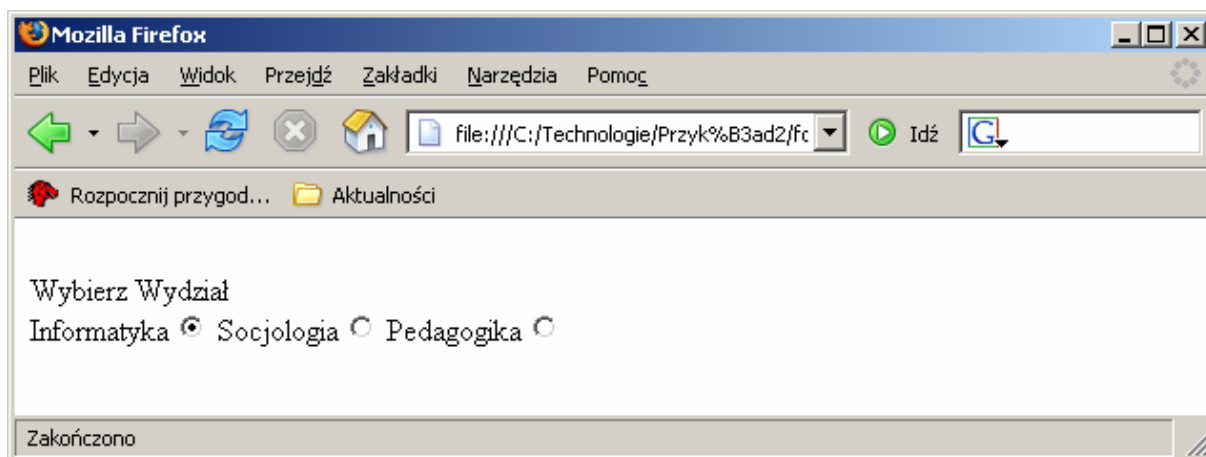


Rysunek 20. Widok dokumentu hipertekstowego zawierającego pole wprowadzania danych zorganizowane przy wykorzystaniu znacznika INPUT

Pole wprowadzania danych może mieć różną postać. Jedną z postaci jest pole wyboru typu: „radio”. Pole typu „radio” (TYPE=„radio”) umożliwia wprowadzanie danych przez wybór jednej z podanych możliwości. Parametrem związanym z polem „radio” jest VALUE. Parametr ten może przybierać wartość „checked” wskazującą domyślnie zaznaczony wybór. Poniżej przedstawiono fragment kodu źródłowego dokumentu zawierającego definicję obszaru wyboru danych:

```
<INPUT type=radio value=informatyka name=wydzial checked>
<INPUT type=radio value=socjologia name=wydzial>
<INPUT type=radio value=pedagogika name=wydzial>
```

Na rysunku 21 przedstawiono widok dokumentu tekstowego zawierającego pola wyboru typu „radio”.



Rysunek 21. Widok dokumentu hipertekstowego zawierającego pola wyboru typu „radio”

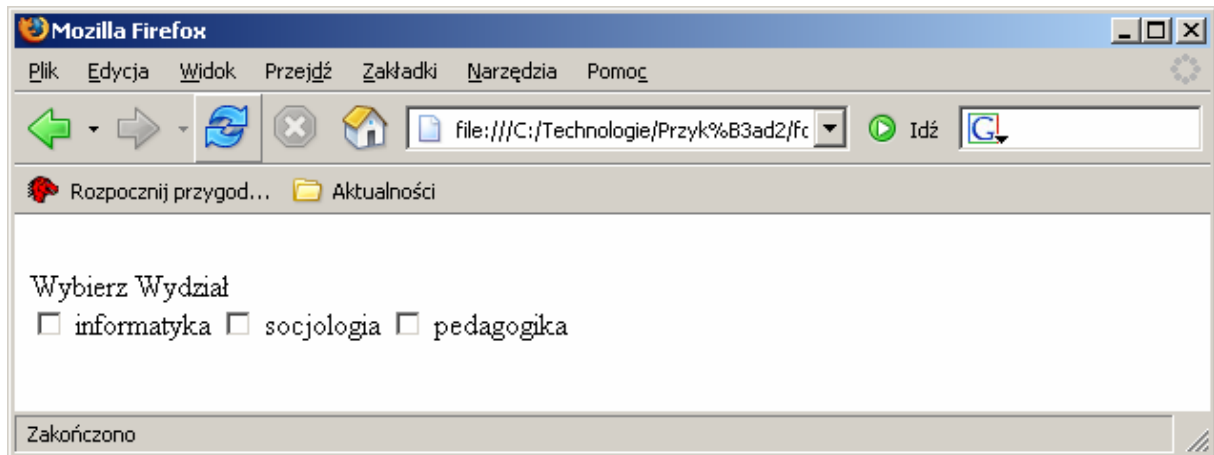
Oprócz pól wyboru typu „radio” na formularzu możliwe jest wstawienie pól wyboru typu „checkbox”. Definicję tego typu pola poprzedza się wpisaniem odpowiednio wartości parametru: TYPE=„checkbox”. Pole wyboru typu „checkbox” funkcjonuje odmiennie od pola typu „radio” (umożliwia dokonanie wyboru wielokrotnego). Poniżej przedstawiono przykładowy kod źródłowy definiujący obszar trzech przełączników typu „checkbox”:

```

<INPUT type=checkbox value=informatyka name=wydzial>
informatyka
<INPUT type=checkbox value=socjologia name=wydzial> socjologia
<INPUT type=checkbox value=pedagogika name=wydzial> pedagogika

```

Na rysunku 22 przedstawiono widok dokumentu hipertekstowego zawierającego obszar formularza z polami typu „checkbox”.



Rysunek 22. Ilustracja wykorzystania pola TYPE="checkbox"

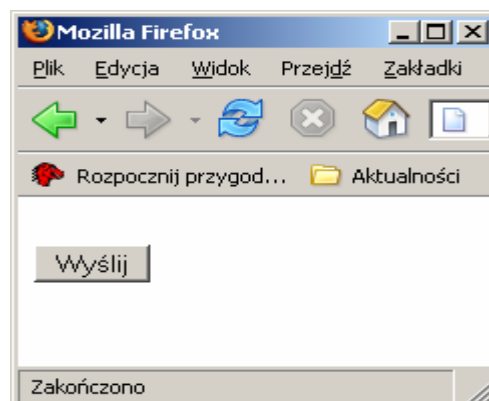
Do wysłania zawartości formularza pod adres wskazany w jego części definicyjnej służy przycisk wysłania. Przycisk wysłania tworzy się przez użycie znacznika INPUT z parametrem TYPE ustawionym na wartość: „submit”. Poniżej przedstawiono przykładowy kod źródłowy tworzący na formularzu przycisk typu „submit” z tekstem „Wyślij” wpisanym wewnątrz przycisku:

```

<INPUT TYPE="submit" VALUE="Wyślij">

```

Na rysunku 23 przedstawiono widok dokumentu hipertekstowego zawierającego pole wejściowe typu „submit”.



Rysunek 23. Widok dokumentu hipertekstowego zawierającego pole wejściowe typu „submit”

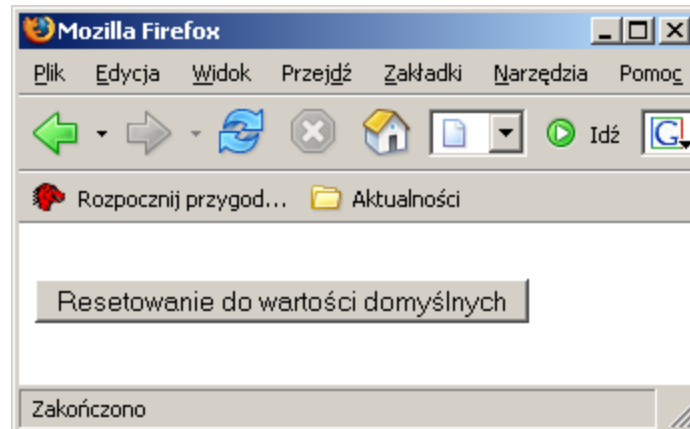
W obszarze definicji formularza możliwe jest przywrócenie wartości domyślnych (w tym min. wykasowanie danych wprowadzonych przez użytkownika”. Do realizacji tego celu służy znacznik INPUT z typem pola ustawionym na „reset” (TYPE="reset"). Poniżej



przedstawiono przykładowy kod źródłowy wstawiany do obszaru formularza w celu przywrócenia danych domyślnych:

```
<INPUT TYPE="reset" VALUE="Resetownie do wartości domyślnych">
```

Na rysunku 24 przedstawiono widok przycisku wstawianego do dokumentu hipertekstowego w wyniku wykonania powyższego kodu.

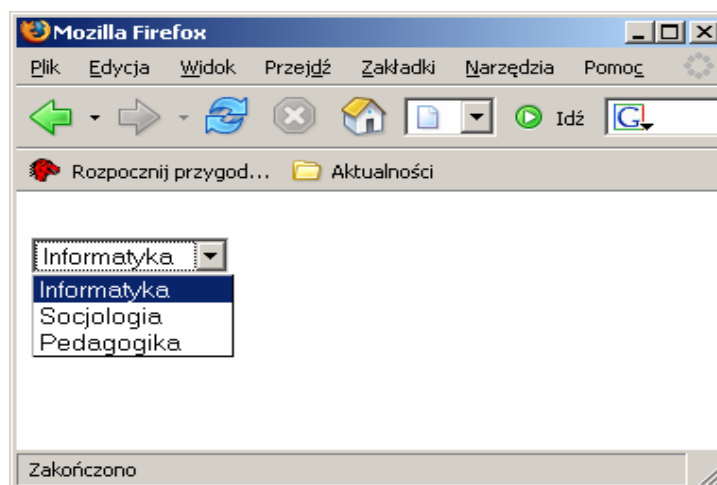


Rysunek 24. Widok przycisku resetującego dane na formularzu wstawionego przy wykorzystaniu znacznika INPUT

Kolejnym rodzajem pól formularza są pola wyświetlane w postaci rozwijalnych list. Do zdefiniowania listy rozwijalnej w obszarze formularza używa się znacznika <SELECT>. Poniżej przedstawiono przykładowy kod źródłowy definiujący trzypozycyjną listę rozwijalną:

```
<SELECT NAME="nazwa_listy">  
<OPTION> pierwsza pozycja  
<OPTION> druga pozycja  
<OPTION> trzecia pozycja  
</SELECT>
```

Na rysunku 25 przedstawiono widok przykładowej listy rozwijalnej umieszczonej w obszarze formularza dokumentu hipertekstowego.



Rysunek 25. Widok przykładowej listy rozwijalnej umieszczonej w obszarze formularza dokumentu hipertekstowego

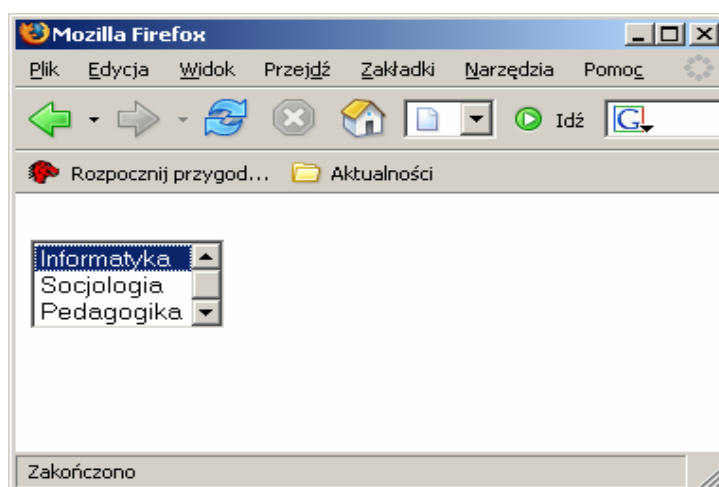
Możliwy jest wielokrotny wybór elementów z listy rozwijalnej. W celu umożliwienia wyboru wielokrotnego w części definicyjnej listy należy wstawić parametr MULTIPLE np.:

```
<SELECT NAME="nazwa_listy" MULTIPLE>
```

W obszarze listy rozwijalnej możliwe jest wskazanie wartości domyślnej. Wartość domyślną wskazuje się przez zastosowanie parametru SELECTED (<OPTION SELECTED>). Do zmiany długości listy, używa się parametru SIZE np.:

```
<SELECT SIZE=10>
```

Na rysunku 26 przedstawiono widok listy rozwijalnej z możliwością wielokrotnego wyboru.



Rysunek 26. Widok dokumentu hipertekstowego z listą rozwijalną umożliwiającą wybór wielokrotny

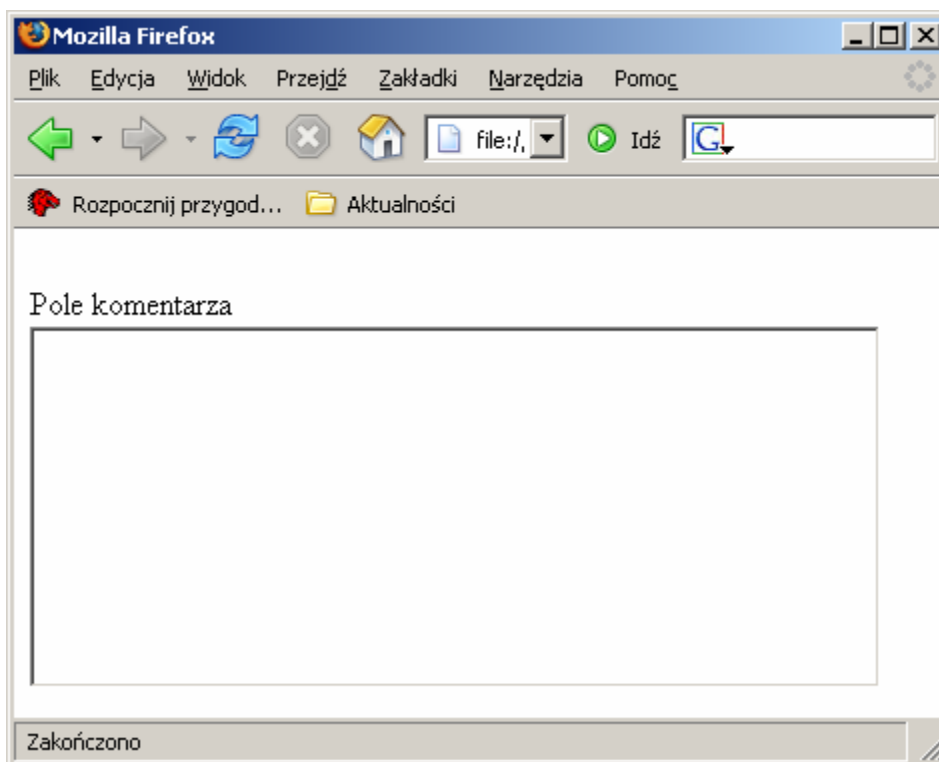
Jednym z ważniejszych pól formularza jest pole tekstowe umożliwiające wprowadzanie dowolnych tekstów. Obszar pola tekstowego definiowany jest przy użyciu znacznika TEXTAREA np.:

```
<TEXTAREA NAME=nazwa> </TEXTAREA>
```

Wielkość pola tekstowego dostępnego dla użytkownika definiuje się przez podanie wartości parametrów ROWS i COLS n.:

```
<TEXTAREA NAME="Pole komentarza" ROWS=10 COLS=50> </TEXTAREA>
```

Na rysunku 27 przedstawiono widok pola tekstowego umieszczonego w obszarze definicyjnym formularza dokumentu hipertekstowego.



Rysunek 27. Widok dokumentu hipertekstowego zawierającego obszar wyznaczony przez pole tekstowe

### 3. Zadania do wykonania

Zadania laboratoryjne należy wykonywać przy użyciu prostego edytora tekstu (notatnik, vi itp.) oraz programu interpretującego wskazanego przez prowadzącego zajęcia.

#### Zadanie 1

W dowolnym edytorze tekstowym (np. Notatnik) stworzyć dokument hipertekstowy o nazwie index.html z podziałem na dwie ramki.

#### Zadanie 2

Zmodyfikować strukturę utworzoną w zadaniu 1 tak, aby zawierała:

- a) 2 ramki poziome i 1 pionową
- b) 2 ramki poziome i 2 pionowe
- c) 3 ramki poziome i 2 pionowe
- d) 3 ramki poziome i 3 pionowe

#### Zadanie 3

Wykonać dokument HTML zawierający tabelę składającą się z 4 kolumn i 6 wierszy. Dane do umieszczenia w poszczególnych komórkach oraz parametry wysokości i szerokości tabeli zostaną podane przez prowadzącego.

#### Zadanie 4

Zmodyfikować tabelę wykonaną w zadaniu 3 poprzez zwiększenie grubości obramowania komórek oraz odstępu między zawartością komórki a jej obramowaniem. Wartości liczbowe pogrubień zostaną wskazane przez prowadzącego zajęcia laboratoryjne.

#### Zadanie 5

Wykonaną tabelę w zadaniu 4 zmodyfikować tak, aby zawartość komórek była wyrównana:

- a) do środka w pionie i poziomie
- b) do środka w pionie i do lewej krawędzi w poziomie
- c) do górnej krawędzi w pionie i do środka w poziomie
- d) do dolnej krawędzi w pionie i do prawej w poziomie

#### Zadanie 6

Zmienić kolor obramowania tabeli oraz pokolorować poszczególne komórki w taki sposób, żeby każdy wiersz tabeli był w innym kolorze.

#### Zadanie 7

W utworzonej, w zadaniu 3 tabeli (4k x 6w) połączyć następujące komórki:

- a) w wierszu 3 – kolumny 2 i 3
- b) w wierszu 4 – kolumny 3 i 4
- c) w kolumnie 2 – wiersze 1 i 3
- d) w kolumnie 4 – wiersze 1 i 2

#### Zadanie 8

Wykonać formularz umożliwiający wprowadzenie i wysłanie na swój adres e-mail danych osobowych np. kandydata na studia w WSM. W formularzu należy użyć wszystkich możliwych pól opisanych w dokumentacji ćwiczenia.

#### Zadanie 9

Wykonać obraz szachownicy w postaci tabeli 9 x 9 pól w kształcie kwadratów, zawierających blok właściwy szachownicy (8 x 8) oraz blok opisów (literowych dla kolumn, liczbowych dla wierszy). W dowolnym edytorze graficznym wykonać bitmapy figur szachowych. Bitmapy figur rozmieścić na utworzonej szachownicy. \*\*\*Co należało by zrobić aby bitmapy figur nie przesłaniały kolorów pól.