

**Tematy:** [Budowa strony ramkowej](#) | [Wyjaśnienie znaczników](#) | [Znacznik frameset](#) | [Znacznik frame](#) | [Ramki w kolumnach i wierszach](#) | [Odsyłacze](#) | [Znacznik noframes](#) | [Definicja typu dokumentu](#)

**O rozdziale (aktualizacja 15.01.2013):** Opisujemy tu technikę budowania witryny w oparciu o ramki, czyli specjalne okienka, do których wczytywane są strony. Technika ta jest jednak uznana w HTML 4 i XHTML 1.0 za schyłkową (pozostaje w obrębie wersji Doctype Transitional i Frameset), zaś w XHTML 1.1 została zaniechana, podobnie jak [pływające ramki](#) (w HTML 5 te ostatnie są jednak zredefiniowane i w pełni uprawnione, choć zmienia się zakres atrybutów). Opis techniki umieszczamy tutaj ze względu na fakt, że jest ciągle jeszcze stosowana przez niektórych webmasterów, odradzając jednak jej stosowanie.

W specyfikacji **HTML 5** ramki także nie zostały uwzględnione, zatem w dającej się przewidzieć perspektywie czasowej możemy uznać ramki za technikę historyczną.

Ramki były szczególnie popularne od połowy lat 90., gdy ich obsługa pojawiła się "na dziko" w przeglądarce Netscape Navigator (później też w innych przeglądarkach), a potem zostały uwzględnione w specyfikacji HTML 4. Na przełomie dekad zaczęły być sukcesywnie wypierane przez tzw. portalowe układy stron oparte na tabelach - przypomnij sobie uwagi dotyczące ewolucji *layoutów* stron zawarte w rozdziale [HTML dla początkujących](#).

## Budowa strony ramkowej

Najbardziej typowym przykładem użycia ramek jest podział okna przeglądarki na dwa pionowe okna i umieszczenie w jednym (zwykle lewym, mniejszym) spisu treści, a przeznaczenie drugiego (prawego, większego) na przywoływane dokumenty (odrębne strony HTML), dowiązane do poszczególnych pozycji spisu. Kliknięcie na jakiegokolwiek pozycji spisu w lewym oknie powoduje wczytanie dokumentu do okna z prawej strony.

Podstawową rolę w "ramkowym" dokumencie pełni specjalna strona, na której definiujemy liczbę, wielkość i pozycję ramek oraz zachowanie się innych dokumentów w momencie uaktywnienia łącza hipertekstowego. Jest ona kontenerem dla ramek. Pozostałe strony, wchodzące w skład całego, wielostronicowego układu, są stronami podrzędnymi.

Zauważmy od razu, że na podstawowej stronie nie powinno być żadnych znaczników poza blokiem nagłówkowym i definicją samych ramek. Dopiero znaczniki **noframes** pozwalają umieścić treść, która będzie wyświetlana przez przeglądarki nie interpretujące ramek. Znaczniki **<body></body>** mogą być umieszczane jedynie w obrębie **noframes**, aczkolwiek ich obecność nie jest konieczna.

Poniższy przykład obrazuje znaczniki wprowadzone na fikcyjnej stronie prezentującej kilka edytorów HTML. Za chwilę wyjaśnimy ich znaczenie, [a oto przykład](#). <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">

```
<html>
<head>
<title>Edytory HTML</title>
</head>
<frameset cols="25%,*%">
<frame scrolling="auto" name="lista edytorów" src="spisedyt.htm">
<frame scrolling="yes" name="edytory" src="edyt.htm">
<noframes>
<body>
Zawartość strony dla przeglądarek nie obsługujących ramek
</body>
</noframes>
</frameset>
</html>
```

### Wyjaśnienie znaczników

Kluczowe znaczenie ma para znaczników `<frameset>` `</frameset>`. Stanowią one, z dodatkowymi parametrami, ramy dla całej konstrukcji i między nimi są umieszczane bardziej szczegółowe znaczniki, dotyczące liczby ramek, podziału strony i wielkości ramek.

Zwróćmy uwagę na "wyłamującą się" parę znaczników `<noframes>` `</noframes>`, o której będziemy mówić [nieco niżej](#). Służą one do wprowadzenia elementów strony, które będą widoczne dla posiadacza przeglądarki, która nie akceptuje ramek. Jeśli natomiast czytelnik strony posługuje się nowoczesną przeglądarką, wszelkie informacje między tymi znacznikami zostaną zignorowane. W ten sposób autor strony może utworzyć dokument uniwersalny, uniezależniony od konkretnej przeglądarki. Nietrudno się domyślić, że treść między znacznikami `<noframes>` `</noframes>` powinna stanowić swoisty równoważnik strony "ramkowej". Odradzamy raczej stosowany dość często komunikat "Twoja przeglądarka nie obsługuje ramek" - szacunek dla gości strony wymaga obsłużenia także tej wąskiej grupy, która z jakichś powodów posługuje się programami nie akceptującymi ramek. Wygodnym rozwiązaniem może być podanie w ramach elementu `noframes` odsyłacza do strony ze spisem treści, która posłuży takiej osobie jako punkt centralny.

Znacznik `<frame>` służy do wprowadzenia konkretnej ramki. Znacznik jest uzupełniany dodatkowymi parametrami, mówiącymi o zawartości (zagnieżdżone strony), o tym, czy ramka zawiera suwaki, a także definiującymi nazwę ramki.

### Znacznik frameset

Aby konstrukcja mogła działać, konieczne jest podzielenie okna przeglądarki - pionowe lub poziome. Poniższy kod (kontynuujemy nasz przykład) zawiera informację o podzieleniu okna na dwie kolumny. Służy do tego atrybut `cols="x,y"`, umieszczany w otwierającym znaczniku `frameset`. Liczby x i y służą do liczbowego określenia szerokości ramek. Możemy podać obie wartości w pikselach, w procentach szerokości strony, a także określić szerokość jednej kolumny i oznaczyć szerokość drugiej za pomocą wieloznacznika \* (ma to sens przy podziale na większą liczbę kolumn lub wierszy).

Przykład, w którym lewa ramka zajmuje 18% szerokości okna przeglądarki, a prawa ramka - 82%:

```
<frameset cols="18%,82%">
```

[Kliknij w tym miejscu, aby zobaczyć przykład:](#)

Gdybyśmy dzielili stronę na trzy wiersze, moglibyśmy, przez analogię, użyć konstrukcji, w której np. górna ramka ma 200 pikseli, dolna 200 pikseli, a środkowa zajmuje pozostały obszar, zależny od wielkości okna przeglądarki:

```
<frameset rows="200,* ,200">
```

[Kliknij w tym miejscu, aby zobaczyć przykład:](#)

[Informacje o FRAMESET](#)

### Znacznik frame

Sam podział na ramki to nie wszystko. Należy im jeszcze przypisać jakieś dokumenty, zazwyczaj w formacie HTML (choć mógłby to być na przykład plik w formacie PDF, który moglibyśmy obejrzeć za pomocą specjalnej wtyczki (plugin) wspomagającej przeglądarkę.

## Ramki

Dodał Administrator  
niedziela, 13 grudzień 2009 15:24

---

Aby przypisać ramce dokument, musimy podać źródło **src** dokumentu. Należy w tym celu użyć konstrukcji:

```
<frame src="nazwa_pliku.htm" ...>
```

Przywoływany plik może się znajdować w tym samym katalogu lub jakimkolwiek innym - podajemy wtedy nazwę pliku z pełną ścieżką dostępu - może to nawet być dokument w innej witrynie.

Jeśli przywoływany plik jest spory objętościowo, zapewne nie zmieści się w całości w ramce. Możemy w takim razie użyć atrybutu **scrolling**, który wyświetli ramkę razem z suwakami. Stosowny atrybut ma postać:

```
<frame scrolling="yes" ...>
```

Gdy dokument jest niewielki, np. zawiera jakiś tytuł, który ma pozostawać na ekranie, możemy zrezygnować z suwaków i wstawić atrybut **scrolling="no"**, ustawiając zarazem odpowiednio szerokość lub wysokość przeznaczoną dla niego ramki. Gdy użyjemy parametru **scrolling="auto"**, suwaki pojawią się wtedy, gdy dokument będzie większy niż ramka.

Ponieważ stosując ramki używamy jednocześnie często odsyłaczy, powinniśmy jakoś nazwać naszą ramkę - **name**. Nazwa będzie służyć jako punkt orientacyjny dla odsyłaczy.

```
<frame name="nazwa_ramki" ...>
```

Czytelnik dokumentu może regulować wielkość wyświetlonych ramek, przesuując brzegi za pomocą myszki. Gdy jednak użyjemy w definicji atrybutu **noresize**

```
<frame noresize ...>
```

zabezpieczymy ramkę przed skalowaniem.. Domyślnie, wszystkie ramki są skalowalne.

Atrybut **frameborder="1"** lub **frameborder="0"** pozwala ustalić, czy obramowanie danej ramki będzie wyświetlane, czy też nie.

```
<frame frameborder="1" ...>
```

Atrybut **marginheight="x"** wymusza dodatkowy odstęp mierzony w pikselach (margines) u góry i u dołu konkretnej ramki, odsuwając treść od obramowania. Z kolei **marginwidth="x"** wymusza margines lewy i prawy.

```
<frame marginheight="50" ...>
```

Ostatecznie cała definicja ramki może mieć postać:

```
<frame scrolling="auto" name="nazwa_ramki" src="nazwa_pliku.htm">
```

W identyczny sposób określamy pozostałe ramki - tyle, ile kolumn (lub wierszy) zadeklarowaliśmy w definicji frameset. W ten sposób zostaną "na dzień dobry" wczytane pewne strony początkowe.

[Informacje o FRAME](#)

## Ramki w kolumnach i wierszach

Powyżej zostały przedstawione dwa konkretne przykłady, obrazujące podział strony na dwie kolumny lub trzy wiersze. Możliwe jest jednak dość dowolne mieszanie definicji kolumn i wierszy, co pozwala np. na utworzenie strony, w której znajdują się trzy wiersze, wiersz środkowy jest podzielony na trzy, a wiersz ostatni na dwie kolumny (łącznie 6 ramek) Oczywiście zbyt wielka liczba ramek ujemnie wpływa na czytelność strony i wygląda

## Ramki

Dodał Administrator  
niedziela, 13 grudzień 2009 15:24

---

mało estetycznie, [vide](#):

Cała sztuka polega na tym, że najpierw dzielimy (w naszym konkretnym przykładzie) stronę na wiersze, a następnie w drugim i trzecim wierszu deklarujemy dodatkowy podział na kolumny - trzy w drugim i dwie w trzecim. Odpowiedni kod mógłby mieć postać: `<frameset rows="200,*,200">`

```
<frame src="strona6.htm">
<frameset cols="35%,*,35%">
  <frame src="strona7.htm">
  <frame src="strona8.htm">
  <frame src="strona9.htm">
</frameset>
<frameset cols="50%,50%">
  <frame src="strona10.htm">
  <frame src="strona11.htm">
</frameset>
</frameset>
```

### Objaśnienie:

1. Rozpoczęliśmy konstrukcję globalną **frameset** od zadeklarowania w pierwszej linii liczby wierszy (trzy).
2. W drugiej linii od razu wstawiliśmy ramkę z przypisaną jej stroną, gdyż pierwszy wiersz nie ulega żadnemu podziałowi.
3. W trzeciej linii ponawiamy definicję **frameset** (niższego rzędu), dzieląc drugi wiersz na trzy kolumny.
4. W czwartej, piątej i szóstej linii wstawiamy definicje konkretnych trzech ramek dla drugiego wiersza, przypisując im odpowiednie strony.
5. W siódmej linii musimy zakończyć tę "lokalną" definicję **frameset**, czyli podział drugiego wiersza.
6. W ósmej linii tworzymy kolejną "lokalną" definicję **frameset**, tym razem dla trzeciego wiersza, dzieląc go na dwie kolumny.
7. W dziewiątej i dziesiątej linii wstawiamy definicje dwóch konkretnych ramek dla trzeciego wiersza, przypisując im odpowiednie strony.
8. W jedenastej linii kończymy "lokalną" definicję **frameset**, czyli podział trzeciego wiersza.
9. W dwunastej linii kończymy "globalną" definicję **frameset**.

## Odsyłacze

Teraz powinniśmy się zastanowić, co zrobić, aby kliknięcie na odsyłaczu w **jednej** ramce automatycznie łądowało odpowiednią stronę w **innej** ramce. Jest to kluczowa sprawa, jeśli chcemy wykorzystać ramki w dokumentach. Zauważmy, że w podanym na początku kliknięcie na odsyłaczu w spisie treści w lewej ramce powinno załadować dokument do prawej ramki, a nie do ramki ze spisem treści. W tym miejscu wykorzystujemy nazwy, które nadawaliśmy poszczególnym ramkom.

Powróćmy do przykładu przedstawionego wyżej w punkcie **Ogólna struktura strony**.

Strona ta składa się z dwóch ramek. Lewa zawiera spis treści, natomiast do prawej są łądowane odpowiednie strony z opisami. Lewej nadaliśmy nazwę **lista edytorów**, natomiast prawej **edytory**. `<frameset cols="25%,*">`  
`<frame scrolling="auto" name="lista edytorów" src="spisedyt.htm">`  
`<frame scrolling="auto" name="edytory" src="edyt.htm">`  
`<noframes>`  
`</noframes>`  
`</frameset>`

Nietrudno się domyślić, że należy dokonać pewnych manipulacji na tej stronie, która jest stale widoczna w lewej ramce o nazwie **lista edytorów**. Do ramki tej jest na początku łądowana strona zawarta w pliku o nazwie - spisedyt.htm.

## Ramki

Dodał Administrator  
niedziela, 13 grudzień 2009 15:24

---

Lista tematów zawiera odsyłacze do odpowiednich stron. Należy je tak skonstruować, aby kliknięcie na odsyłaczu przywoływało wskazaną stronę, ale zarazem ładowało ją do prawej ramki o nazwie **edytory**. Można to uczynić za pomocą atrybutu **target**, czyli następującej konstrukcji: `<a href="fp97.htm" target="edytory">FrontPage 97</a>`  
`<a href="hs25.htm" target="edytory">HomeSite 2.5</a>`  
`<a href="paj2000.htm" target="edytory">Pajaczek 2000</a>`

...

itd.

**a href="nazwa\_ładowanej\_strony"** dotyczy nazwy pliku, w którym jest zapisana dana strona

**target="nazwa\_ramki"** dotyczy docelowego miejsca, do którego ma być załadowana strona. Widzimy tutaj rolę, jaką pełnią nazwy ramek.

FrontPage 97, HomeSite 2.5 i Pajaczek 2000 są konkretnymi tematami w ich spisie, pozwalającymi po prostu wybrać interesujące nas zagadnienie.

Zwróćmy jeszcze uwagę na cztery zastrzeżone wartości w przypadku atrybutu **target**.

**target="\_blank"** - w tym przypadku przywoływany dokument zostanie załadowany do nowego, pustego okna (uruchomi nowe okno lub instancję przeglądarki).

**target="\_self"** - w tym przypadku (przypadek domyślny) nowy dokument zamieni w ramce dokument, z którego dokonujemy skoku.

**target="\_parent"** - w tym przypadku przywoływany dokument zamieni dokument nadrzędny w hierarchii dla bieżącego dokumentu.

**target="\_top"** - w ostatnim przypadku przywoływany dokument zostanie wstawiony na miejsce dokumentu pierwszego w hierarchii naszych skoków, a więc na samą "górze".

Szczególnie często stosowana jest ta ostatnia wartość. Użycie **"\_top"** pozwoli uniknąć dość kłopotliwej sytuacji, gdy do ramki jest ładowany dokument, który sam składa się z ramek. **"\_top"** usuwa więc wszelkie istniejące ramki i wyświetla dokument "na czysto". Naturalnie moglibyśmy tutaj z powodzeniem użyć **"\_blank"**, ale pozbawiamy się w ten sposób "historii" skoków.

## Znacznik noframes

Atrybut **noframes** jest ukłonem w stronę użytkowników, którzy nie mogą oglądać ramek w swojej przeglądarce, np. w prostej przeglądarce tekstowej (inna sprawa, że są one już niezwykle rzadko stosowane).

Posiadacz takiej przeglądarki zobaczyłby pustą stronę, gdyby między **<noframes>** **</noframes>** nie było żadnej informacji. Dlatego musimy w tym miejscu powtórzyć informację zawartą w jednej z ramek. W naszym przykładzie podstawową stroną jest strona o nazwie Edytory HTML (plik `edytory.htm`). Strona ta zawiera definicję dwóch ramek: do lewej jest na stałe przypisana strona `spisedyt.htm`, do prawej natomiast są wlewane inne strony, zależnie od wskazanego tematu.

Z kolei między znacznikami **<noframes>** **</noframes>** znalazło się powtórzenie treści strony `spisedyt.htm` (lista edytorów), jako strony wyjściowej dla osób korzystających z prostej przeglądarki. Tym samym osoby te, wywołując stronę, zobaczą stronę Edytory HTML (`edytory.htm`), ale w taki sposób, jakby wywołały stronę Lista edytorów (`spisedyt.htm`). Z poziomu tej strony mogą sięgać do konkretnych tematów, które będą oczywiście zajmować cały ekran, a nie ramkę, jak w nowoczesnej przeglądarce. Naturalnie powinny się tutaj znaleźć stosowne odsyłacze hipertekstowe, które pozwolą nawigować w całym, wielostronicowym zestawie. Przypomnijmy, że gdy używamy nowoczesnych programów, wszelkie informacje między znacznikami **noframes** zostaną zignorowane.

W naszym przykładzie będzie to następujący blok: **<noframes>**  
**<body>**

```
<h1>Edytory HTML</h1>
<ul>
<li><a href="corelweb.htm">Corel WebGraphics Suite</a>
<li><a href="webmast.htm">Corel WebMaster Suite</a>
<li><a href="fp97.htm">FrontPage 97</a>
<li><a href="hedit11.htm">Hedit 1.1</a>
<li><a href="homesite.htm">HomeSite 2.5</a>
<li><a href="hotdog3.htm">HotDog 3</a>
<li><a href="composer.htm">Netscape Composer</a>
<li><a href="paj2000.htm">Pajęczek 2000</a>
</ul>
</body>
</noframes>
```

**Uwaga:** Specyfikacja HTML 4 przewiduje, że znaczniki **noframes** mają być umieszczone w ramach **frameset** "najwyższego rzędu".

[Informacje o NOFRAMES](#)

## Definicja typu dokumentu

Jak wspomnieliśmy we wstępie, ramki nie są już obecne w specyfikacji HTML 4.01 Strict, a jedynie w HTML 4.01 Transitional i Frameset. Jeśli stosujesz w swojej witrynie ramki, strona zawierająca definicję ramek powinna mieć podaną następującą deklarację typu dokumentu:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">
```

Również ramka, w której znajdują się odsyłacze wczytujące dokumenty do innej ramki nie może mieć deklaracji Strict, gdyż nie przewiduje ona atrybutu **target**. Dlatego konieczne jest użycie deklaracji Transitional:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

Za to strony przywoływane do głównej ramki mogą mieć deklarację DTD odpowiadającą wersji Strict.

