

Elementy osadzone (grafika, multimedia, aplikacje)

Obrazek widoczny na stronie WWW od wielu lat nikogo nie dziwi. Dzisiaj standardem jest witryna WWW, na której poza statycznymi elementami pojawiają się animacje, filmy, treści audio i rozbudowane aplikacje. Język HTML5 daje nam znacznie więcej możliwości w zakresie publikacji treści multimedialnych i wiele rozwiązań stosowanych do tej pory znacznie upraszcza (np. publikację wideo i audio).

Publikacja obrazków

Grafika na stronie może być jej uzupełnieniem, ale może także stanowić podstawowy składnik strony. Na stronie WWW możesz umieścić obrazki w następujących formatach: *.jpg*, *.gif* oraz *.png*. Na potrzeby niniejszego podrozdziału założyłem, że posiadasz dowolny obrazek, który sam przygotowałeś lub pobrałeś z jakiejś strony w Internecie. Pamiętaj jednak o prawach autorskich — pożyczonego zdjęcia możesz używać do testów i nauki, ale wyniki Twojej pracy nie powinny wyjść poza Twój komputer. Moim zdaniem własnoręczne przygotowanie elementów graficznych strony daje dużo więcej satysfakcji niż ich "pożyczanie" od innych twórców.

Do umieszczenia elementu graficznego na stronie służy znacznik ``, który posiada atrybuty `src` oraz `alt`. `Src` określa źródło, czyli obrazek, jaki chcemy wstawić, `alt` natomiast definiuje alternatywny tekst dla przeglądarek nieobsługujących grafiki, m.in. przeglądarek tekstowych, np. LYNX, bądź przeglądarek, w których użytkownik wyłączył wyświetlanie grafiki. Przykład użycia znacznika `` podaję poniżej.

```

```

Jeżeli obrazek umieszczony na stronie ma kluczowe znaczenie dla jej zawartości, obowiązkowo musi posiadać atrybut `alt` wraz z dokładnym opisem. W przypadku gdy obrazek nie ma większego znaczenia i jest na przykład ozdobnikiem, należy wykorzystać pusty atrybut `alt`.

```

```

W tym miejscu pojawia się bardzo ważna kwestia, związana z wielkością liter używanych przy wpisywaniu znaczników oraz atrybutów. W przypadku stron korzystających ze starej wersji języka HTML wielkość liter w znacznikach oraz atrybutach nie ma znaczenia, natomiast HTML5 wymaga stosowania wyłącznie małych liter. Od tej zasady istnieje pewne odstępstwo. Chodzi o sytuację, w której znacznik zawiera odwołanie do oddzielnego pliku. W naszym przykładzie atrybut `src` zawiera informacje o pliku obrazka i w takim przypadku sprawą kluczową jest wpisanie nazwy obrazka dokładnie tak samo, jak nazwano plik.

Chodzi o to, że systemy rodziny MS Windows nie rozróżniają wielkości liter i dla nich *plik.gif* oraz *Plik.gif* to jeden i ten sam plik. Natomiast systemy uniksowe (niemal 95% serwerów w Internecie działa w tym systemie) są czułe na wielkość liter i wskazane tu pliki to dwa zupełnie różne pliki.

Pamiętaj o tym, że podczas osadzania obrazka na stronie musisz zdefiniować ścieżkę dostępu do odpowiedniego pliku. Dla uproszczenia założymy, że plik obrazka znajduje się w tym samym miejscu co plik strony HTML.

```
  
  

```



Rysunek 1 Przykład powiększania obrazka za pomocą atrybutów języka HTML

Warto mieć również na uwadze to, że do skalowania rozmiaru obrazków umieszczonych na stronie możesz także wykorzystać kaskadowe arkusze stylów. Rozwiązanie to wydaje się wygodniejsze i bardziej praktyczne.

Skalowanie obrazka za pomocą atrybutów języka HTML lub CSS powoduje pogorszenie jego jakości, a dodatkowo w przypadku zmniejszania oryginału nie zmniejsza wielkości samego pliku. Dlatego zaleca się przygotowanie obrazka o wymiarach zgodnych z potrzebami konkretnej strony.

Ćwiczenie 16

1. Utwórz folder „16 Publikacja obrazów”.

2. Posługując się plikiem z ćwiczenia 2 i dowolnym plikiem graficznym utwórz stronę zawierającą grafikę i jej powiększenie x2 i x3 oraz pomniejszenie. Wzorując się na ćwiczeniu 14 podpisz rysunki.
3. Efekt przetestuj w przeglądarce.

Publikacja filmów

Nowoczesne strony WWW to nie tylko statyczne obrazki w formacie *.jpg*, *.png* czy *.gif*, ale również filmy. Propozycja piątej odsłony języka HTML wprowadza nowy sposób publikowania filmów na stronie, przez co cały proces jest znacznie prostszy, niż miało to miejsce do tej pory. Wystarczy jeden znacznik, kilka atrybutów i po sprawie.

W chwili obecnej obrazy wideo publikowane na stronie powinny być zapisane w jednym z trzech formatów:

- **.mp4** — plik kompresowany za pomocą kodeka wideo **H.264** i kodeka audio **AAC**;
- **.ogg** — plik kompresowany za pomocą kodeka wideo **Theora** i kodeka audio **Vorbis**;
- **.webm** — plik kompresowany za pomocą kodeka wideo **On2 VP8** i kodeka audio **Vorbis**.

Za osadzenie obrazu wideo na stronie WWW odpowiada znacznik `<video></video>`, który występuje w parze z kilkoma dodatkowymi atrybutami. Otwierający znacznik `<video>` zawiera obowiązkowy atrybut `src=""`, w którym powinno się znaleźć odwołanie do konkretnego pliku z filmem. Pomiedzy znacznikami `<video></video>` należy zawrzeć tekst informujący o tym, że dana przeglądarka nie wspiera języka HTML5 i nie potrafi wyświetlić filmu.

Opcjonalnie w otwierającym znaczniku `<video>` możemy zdefiniować wymiary okna, w którym odtwarzany będzie film. Za określenie wysokości i szerokości odpowiadają atrybuty `height=""` i `width=""`.

Brak zadeklarowanych wymiarów spowoduje, że przeglądarka domyślnie wyświetli okno filmu w takiej rozdzielczości, w jakiej go zakodowano.

Przykładowa deklaracja odpowiedzialna za wstawienie filmu na stronę WWW może mieć postać:

```
<video src="plik-video.mp4">
Twoja przeglądarka nie obsługuje znacznika video.
</video>
```

Następnym opcjonalnym elementem jest atrybut `controls="controls"`, który odpowiada za wyświetlenie paska postępu, przycisku odtwarzania oraz zmiany głośności. Brak atrybutu spowoduje, że pod oknem, w którym wyświetlany jest film, nie pojawi się belka odtwarzania. Rozwiązanie takie

może się przydać, gdy na stronie umieszczasz na przykład banner reklamy lub gdy film wideo jest integralnym elementem wyglądu strony WWW.

```
<video src="plik-video.mp4"width="480"height="360" controls="controls">
Twoja przeglądarka nie obsługuje znacznika video.
</video>
```

Jeżeli udostępniony film ma być odtwarzany automatycznie po wejściu na stronę, musisz skorzystać z kolejnego atrybutu o nazwie **autoplay="autoplay"**. Dzięki takiemu rozwiązaniu gość odwiedzający Twoją stronę nie musi podejmować żadnych działań, aby uruchomić odtwarzanie filmu.

```
<video src="plik-video.mp4"width="480"height="360"
controls="controls"autoplay="autoplay">
Twoja przeglądarka nie obsługuje znacznika video.
</video>
```

Za pomocą atrybutu **preload="preload"** możesz wymusić, aby odtwarzanie filmu było możliwe dopiero po jego pobraniu na komputer osoby odwiedzającej stronę. Rozwiązanie to pozwoli uniknąć problemów z płynnością w odtwarzaniu obrazu i dźwięku.

```
<video src="plik-video.mp4"width="480"height="360"
controls="controls"preload="preload">
Twoja przeglądarka nie obsługuje znacznika video.
</video>
```

Ostatnim alternatywnym atrybutem jest **loop="loop"**, który odpowiada za wymuszenie ciągłego odtwarzania tego samego filmu. Dokładniej mówiąc, po zakończeniu odtwarzania przeglądarka sama ponownie uruchomi film i czynności będzie powtarzać, dopóki użytkownik strony nie zatrzyma tego procesu lub nie opuści witryny.

```
<video src="plik-video.mp4" width="480" height="360" loop="loop">
Twoja przeglądarka nie obsługuje znacznika video.
</video>
```

Mając na uwadze problemy z odtwarzaniem różnych formatów przez różne przeglądarki, warto zadać sobie trud i ten sam film zakodować w kilku formatach, a następnie podłączyć je pod stronę. Można to zrobić, stosując nieco zmodyfikowany zapis.

```
<video width="480" height="360" controls="controls">
<source src="plik-video.mp4"type="video/mp4"/>
<source src="plik-video.ogv"type="video/ogg"/>
Twoja przeglądarka nie obsługuje znacznika video.
</video>
```

Ćwiczenie 17

1. Utwórz folder „17 Publikowanie filmów”.
2. Posługując się przykładowymi plikami filmowymi utwórz dokument HTML do ich odtwarzania. Wybrany film wideo należy zaprezentować w dwóch formatach, z paskiem odtwarzania i bez, z automatycznym odtwarzaniem oraz z odtwarzaniem ciągłym. Wszystkie sposoby mogą być zaprezentowane w jednym pliku lub rozłożone na kilka plików HTML. Czy wszystkie pliki dadzą się odtworzyć we wszystkich przeglądarkach?
3. Efekt zademonstruj w przeglądarce.

Uwaga: pliki filmowe do ćwiczenia 17 znajdują się pod adresem:

<http://graczyk.pwsz.glogow.pl/motylek/motylek.mp4>

<http://graczyk.pwsz.glogow.pl/motylek/motylek.ogv>

W dokumencie proszę posłużyć się podanymi adresami w taki sposób, aby kopiowanie filmów na dysk lokalny nie było konieczne.

Publikacja plików audio

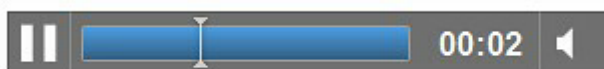
Skoro można na stronie WWW osadzić obrazy wideo, to rzeczą oczywistą jest to, że da się również zrobić to samo w odniesieniu do plików audio. W chwili obecnej pliki audio publikowane na stronie powinny być zapisane w jednym z trzech formatów: *.mp3*, *.ogg* lub *.wave*.

Na początku przyjrzyjmy się prostemu przykładowi, który pozwoli nam osadzić plik audio w kodzie strony.

```
<audio src="plik-audio.mp3" controls="controls">
Twoja przeglądarka nie obsługuje znacznika audio.
</audio>
```

Jak widać w załączonym powyżej przykładzie, za osadzenie pliku audio na stronie WWW odpowiada znacznik `<audio></audio>`, który występuje w parze z kilkoma dodatkowymi atrybutami. Otwierający znacznik `<audio>` zawiera obowiązkowy atrybut `src=" "`, w którym powinno się znaleźć odwołanie do konkretnego pliku audio. Pomiędzy znacznikami `<audio></audio>` należy zawrzeć tekst informujący o tym, że dana przeglądarka nie wspiera języka HTML5 i nie potrafi odtworzyć pliku audio.

Opcjonalnym, ale moim zdaniem istotnym elementem jest atrybut `controls="controls"`, który odpowiada za wyświetlenie paska postępu, przycisku odtwarzania oraz zmiany głośności. Brak tego atrybutu spowoduje, że na stronie nie pojawi się belka widoczna na rysunku.



```
<audio src="plik-audio.wav" controls="controls">
```

```
Tvoja przeglądarka nie obsługuje znacznika audio.  
</audio>
```

Jeżeli udostępniony plik audio ma być odtwarzany automatycznie po wejściu na stronę, musisz skorzystać z kolejnego atrybutu o nazwie **autoplay="autoplay"** podobnie jak w przypadku plików wideo.

```
<audio src="plik-audio.wav" controls="controls"autoplay="autoplay">  
Tvoja przeglądarka nie obsługuje znacznika audio.  
</audio>
```

Za pomocą atrybutu **preload="preload"** możesz wymusić, aby odtwarzanie pliku audio było możliwe dopiero po jego pobraniu na komputer osoby odwiedzającej stronę jak w przypadku wideo.

```
<audio src="plik-audio.wav" controls="controls" preload="preload">  
Tvoja przeglądarka nie obsługuje znacznika audio.  
</audio>
```

Ostatnim alternatywnym atrybutem jest **loop="loop"**, który odpowiada za wymuszenie ciągłego odtwarzania tego samego pliku audio.

```
<audio src="plik-audio.wav" controls="controls" loop="loop">  
Tvoja przeglądarka nie obsługuje znacznika audio.  
</audio>
```

Mając na uwadze problemy z odtwarzaniem formatów *.mp3*, *.wave* oraz *.ogg* przez różne przeglądarki, warto zadać sobie trud i ten sam plik zakodować w formatach *.mp3* i *.ogg*, a następnie podłączyć je pod stronę. Można to zrobić, stosując nieco zmodyfikowany zapis.

```
<audio controls="controls">  
<source src="plik-audio.ogg" type="audio/ogg"/>  
<source src="plik-audio.mp3" type="audio/mpeg"/>  
Tvoja przeglądarka nie obsługuje znacznika audio.  
</audio>
```

Dzięki takiemu podejściu przeglądarka sama będzie mogła wybrać format, który wspiera, i poprawnie odtworzy Twój plik audio. Niestety działanie takie wymaga dodatkowej pracy polegającej na zakodowaniu dźwięku w dwóch różnych formatach, ale to jedyny sposób, aby poszerzyć krąg przeglądarek, w których zadziała Twoja strona.

Ćwiczenie 18

1. Utwórz folder „18 Publikacja plików audio”.
2. Utwórz dokument HTML zawierający publikację plików wzorcowych.
3. Przetestuj działanie dokumentów w dostępnych przeglądarkach.

Do ćwiczenia można użyć pliku dostępnego pod adresem:

<http://graczyk.pwsz.glogow.pl/motylek/muzyka.mp3>

Publikacja animacji flash

Wstępna wersja piątej odsłony specyfikacji języka HTML wprowadza nowy znacznik, który znacznie upraszcza osadzanie na stronie animacji wykonanych w technologii Adobe Flash. Mam tutaj na myśli **znacznik <embed />**.

Znacznik <embed /> występuje wraz z atrybutem src=" ", w którym powinno się znaleźć odwołanie do konkretnego pliku z animacją. Dodatkowo omawiany znacznik może być uzupełniony o atrybuty width=" " i height=" ", które odpowiadają za określenie szerokości i wysokości osadzonego elementu. Przykładowy kod HTML odpowiedzialny za osadzenie animacji flash może mieć następującą postać.

```
<div>  
<embed src="przyklad0.swf" width="550" height="400"/>  
</div>
```

Ćwiczenie 19

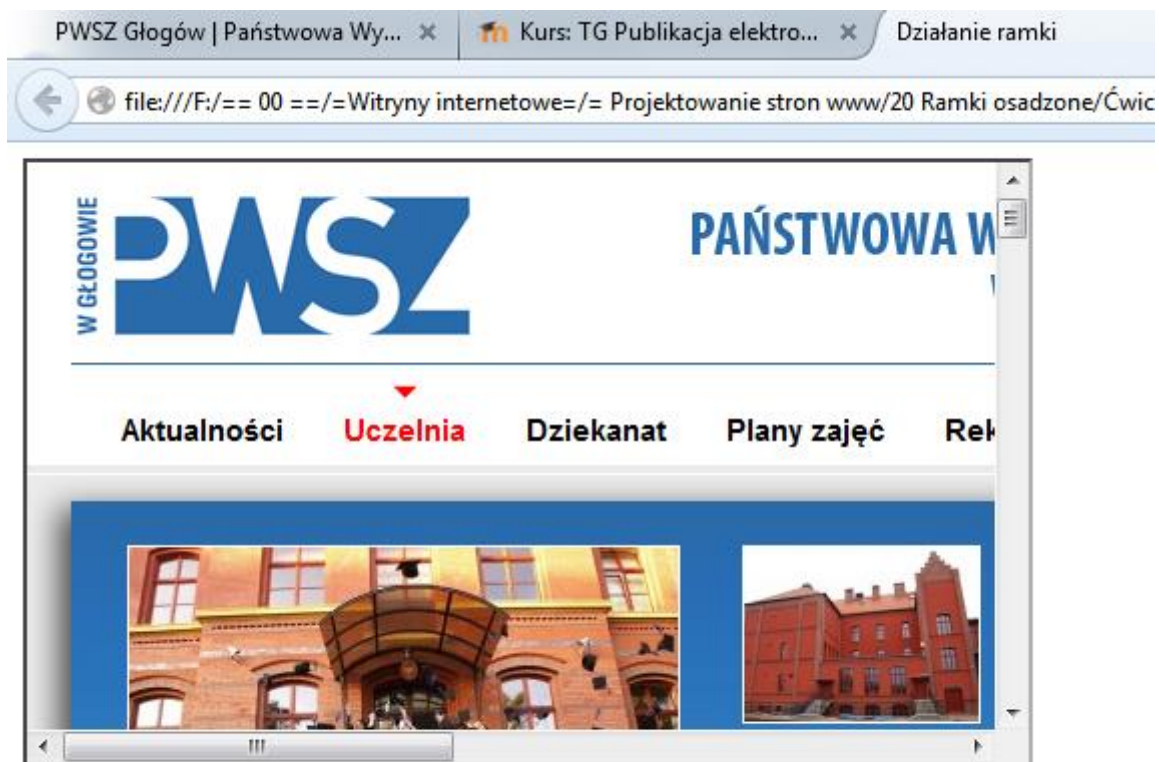
1. Utwórz folder „19 Publikacja animacji flash”.
2. Darmowe animacje do testów nowego znacznika możesz pobrać ze strony <http://www.free-flash-animations.com/>.
3. Utwórz dokument HTML zawierający pobraną animację.
4. Przetestuj dokument w różnych w dostępnych przeglądarkach.

Ramki osadzone

Ramki osadzone (zwane również ramkami pływającymi) to nic innego jak swego rodzaju "okna" na stronie, w których wyświetlana jest zawartość innej strony bądź pliku HTML. Rozwiązanie takie przydaje się w różnych sytuacjach i mimo swoich licznych wad i ograniczeń jest bardzo często stosowane, np. do osadzania bannerów reklamowych.

Za osadzanie ramki pływającej odpowiada znacznik <iframe></iframe>, który jest uzupełniany o kilka dodatkowych atrybutów. Pierwszym i najważniejszym atrybutem jest src=" ", który odpowiada za wskazanie i podłączenie pod ramkę konkretnego dokumentu lub strony. Listing widoczny poniżej prezentuje minimalny zapis, który jest niezbędny do tego, aby na stronie osadzić działającą pływającą ramkę.

```
<iframe src="http://pwsz.glogow.pl">  
  
</iframe>
```



Kolejnym atrybutem, który może występować w parze z otwierającym znacznikiem `<iframe>`, jest `name=" "`, pozwalający przypisać unikalną nazwę do tworzonej ramki. Dzięki takiemu rozwiązaniu możemy adresować odnośniki umieszczone na stronie w taki sposób, aby treści, do których prowadzą, pojawiały się w naszej ramce. Poniżej widać przykład wykorzystania omawianego atrybutu:

```
<iframe src="http://pwsz.glogow.pl" name="ramka">
</iframe>
```

Odnośnik mający za zadanie wyświetlić w konkretnej ramce treść, do której prowadzi, musi zostać nieznacznie zmodyfikowany i uzupełniony o atrybut `target=" "`. W tym atrybucie podana jest nazwa ramki. Kliknięcie wybranego odnośnika spowoduje otwarcie jego zawartości w ramce umieszczonej powyżej.

Ramka osadzona na stronie może mieć określoną wysokość i szerokość. Podobnie jak w innych przypadkach również teraz musimy skorzystać z atrybutów `height=" "` i `width=" "`.

```
<iframe src="http://pwsz.glogow.pl" width="500" height="300">
</iframe>
```

Ćwiczenie 20

1. Utwórz folder o nazwie „20 Ramki osadzone”.
2. Utwórz dokument HTML zawierający osadzoną ramkę do strony uczelni.
3. Dokument przetestuj w dostępnych przeglądarkach.

Hipertłącza

Hipertłącza są podstawą funkcjonowania stron WWW i całej sieci. Bez hipertłaczy przeglądanie stron byłoby bardzo utrudnione i wymagałoby od odwiedzającego znajomości budowy danego serwisu WWW oraz ręcznego wpisywania odpowiednich adresów.

Proste hipertłącza

Budowa hipertłącza jest stosunkowo prosta i wygląda tak:

```
<a href="strona.html"> Kliknij tutaj.</a>
```

Jak łatwo wywnioskować, znacznikiem odpowiedzialnym za odsyłacze jest `<a>`, wzbogacony o pewne atrybuty. I tak dla odsyłacza takim obowiązkowym atrybutem jest `href="adres lub nazwa pliku"`, określający, do jakiego dokumentu bądź miejsca ma prowadzić nasz odnośnik. Ma on więc postać:

```
<a href="nazwa_pliku.html">
```

```
<a href="http://start.pl/">
```

```
<a href="http://start.pl/nazwa_pliku.html">
```

To tylko pierwsza część odsyłacza, informująca, dokąd mamy się przenieść. Teraz musimy opisać nasz odsyłacz tak, by był widoczny i dostępny ze strony:

```
<a href="http://www.pwsz.glogow.pl/">Strona PWSZ</a>
```

Oczywiście musimy pamiętać o znaczniku zamykającym ``, gdyż bez niego opis naszego odsyłacza będzie nieskończenie długi.

Przykładowe typy odsyłaczy

dokumenty <i>html</i> , <i>txt</i> , <i>zip</i> itd.	<code>Opis</code>
obrazek	<code>Opis</code>
adresy URL	<code>Opis</code>
adresy FTP	<code>Opis</code>
adresy e-mail	<code>Opis</code>
adresy grup dyskusyjnych	<code>Opis</code>

Jak widać, możemy odnosić się w zasadzie do wszystkiego, co nam przyjdzie do głowy, a przeglądarka pokaże nam wynik bądź zaproponuje zapisanie na dysku danego obiektu, np. *plik.zip lub xls jak w przypadku planu zajęć na stronie uczelni*.

Nieco inaczej wygląda kwestia odsyłacza do adresu e-mail, który umożliwia otwarcie odpowiedniego okna programu pocztowego w celu wysłania poczty. Musisz jednak pamiętać, że jeżeli oglądający Twoją stronę nie ma konta e-mail, to w ten sposób nie będzie mógł do Ciebie napisać.

Odsyłacz do poczty możemy nieznacznie rozbudować, dzięki czemu zyskamy możliwość narzucenia tematu wiadomości. Odpowiednia konstrukcja znajduje się poniżej.

```
<a href="mailto:jan_janowski@wp.pl?Subject=Zamówienie">Zamówienie ze strony www</a>
```

Po kliknięciu tak skromnego odsyłacza zostanie otwarte okno nowej wiadomości, a w polu temat znajdzie się tekst *Zamówienie*.

Ćwiczenie 21

1. Utwórz folder o nazwie „21 Tworzenie hiperłączy”.
2. Otwórz plik przykładowy z Ćwiczenia 02.
3. Pomiędzy znacznikami `<body></body>` umieść swoje odsyłacze. W tym celu skorzystaj ze znacznika `<a>`. Spróbuj umieścić odsyłacz do strony uczelni, kursu elearningowego, swojej strony w weebly oraz swojego adresu e-mail. Poszczególne odsyłacze spróbuj przedstawić w formie listy wypunktowanej (Ćwiczenie 11 Lista).
4. Wynik sprawdź w przeglądarce.

Kotwice

Przedstawione odsyłacze do dokumentów odnoszą się do konkretnego pliku, ale w połączeniu z kotwicą pozwalają odnieść się do konkretnego miejsca w danym dokumencie. Oznacza to, że odniesienie się do konkretnego pliku i do zadeklarowanej kotwicy spowoduje otwarcie przez przeglądarkę wskazanego pliku oraz przewinięcie dokumentu, tak by wiersz zawierający kotwicę był możliwie najbliżej górnej krawędzi okna przeglądarki. Jest to rozwiązanie ogromnie przydatne w przypadku tworzenia dokumentów dłuższych niż jedna lub dwie długości ekranu monitora, gdyż ułatwia nawigację.

Deklaracja kotwicy wygląda tak:

```
<h1><a name="etykieta">Ważny</a> nagłówek</h1>
```

Wraz z wprowadzeniem propozycji piątej specyfikacji HTML5 opisano również nowy sposób dodawania kotwicy. Spróbuj porównać poniższy i powyższy listing ze sobą.

```
<h1 id="etykieta">Ważny nagłówek</h1>
```

Nowy sposób jest dużo wygodniejszy, ale należy pamiętać o tym, iż atrybut `id=""` może być również wykorzystywany przez kaskadowe arkusze stylów. Dlatego należy zadbać o to, aby przypisana wartość była unikalna w skali całego dokumentu.

Dalej przedstawiam sposób odniesienia się do tej kotwicy za pomocą polecenia umieszczonego wewnątrz kodu. Oczywiście odwołanie działa z oboma sposobami definiowania kotwic.

`<p>Jeśli chcesz odwołać się do kotwicy znajdującej się na tej stronie, kliknij`

Jeżeli chcesz się odwołać do kotwicy zamieszczonej na tej samej stronie co odnośnik, to wystarczy następująca deklaracja:

```
<a href="#gora">Do góry</a>
```

#gora to nazwa kotwicy znajdującej się w górnej części strony. Działanie takiego odsyłacza jest bardzo proste i ogranicza się do przewinięcia strony i ustawienia linii zawierającej kotwicę o nazwie #gora równo z górnym brzegiem okna przeglądarki.

Stosowanie kotwic oraz odsyłaczy dających możliwość odwołania się do konkretnego miejsca ma sens zwłaszcza przy bardzo długich stronach. W takim przypadku odsyłacz i kotwica mogą uprościć nawigację na stronie.

Przykład:

```
<body>
```

```
<a href="#etykieta">Do drugiego</a>
```

```
<h1 id=gora>Nagłówek stopnia pierwszego</h1><hr>
```

```
<p> .....Tutaj dużo tekstu....</p>
```

```
<h2 id="etykieta">Nagłówek stopnia drugiego</h2>
```

```
<p> Trochę tekstu</p>
```

```
<a href="#gora">tutaj</a>
```

```
</body>
```

Ćwiczenie 22

1. Utwórz folder o nazwie „22 kotwica”.
2. Wykorzystując tekst, który pojawił się już w Ćwiczeniu 08 Akapity utwórz stronę z dużą ilością tekstu i zastosuj kotwice.
3. Efekt przetestuj w przeglądarkach.

Adresy względne i bezwzględne

Istnieją dwa ich rodzaje odsyłaczy: **względne** i **bezwzględne**. Odsyłacz bezwzględny zawiera dokładny adres URL do danego pliku i wygląda tak:

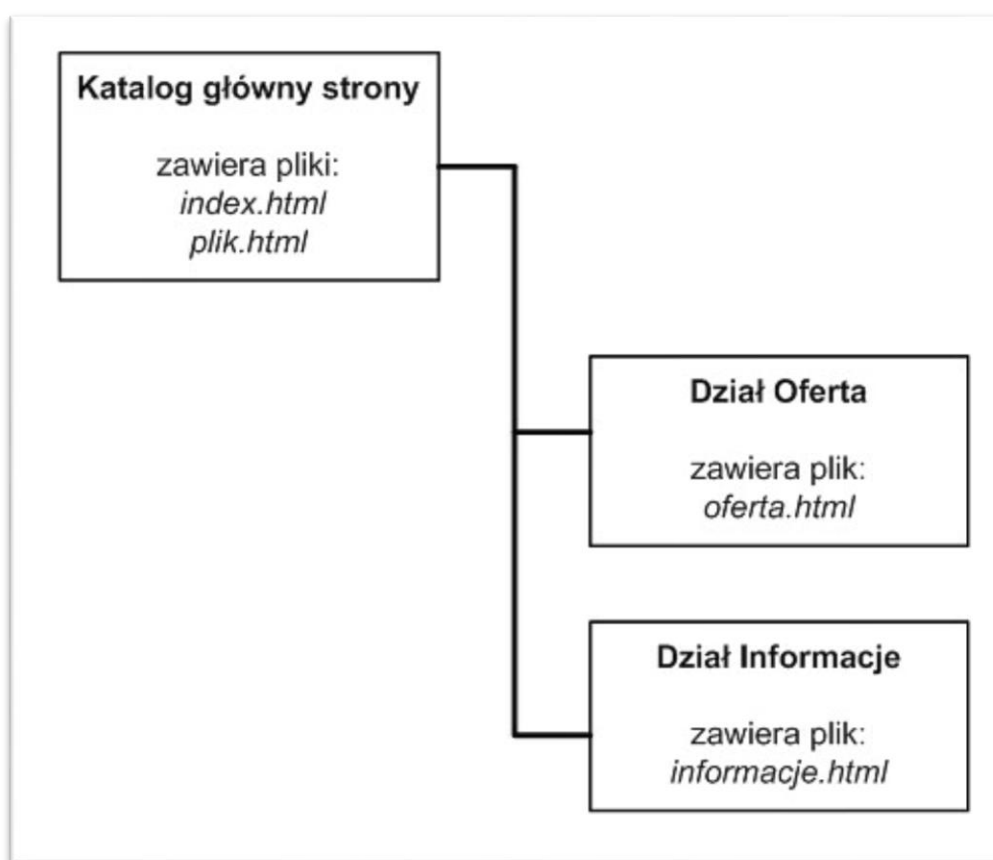
`To jest odnośnik do pliku`

Odnośnik taki ma jedną istotną wadę. W przypadku zmiany serwera, na którym publikujemy witrynę, a co za tym idzie — jego adresu, będziemy musieli poprawić wszystkie odnośniki zapisane w ten sposób. Odnośników bezwzględnych powinno się używać do łączenia zewnętrznych stron, a przy dokumentach wewnątrz jednego serwisu należy stosować odnośniki względne.

Odnośnik względny ma następującą postać:

`To jest odnośnik względny`

Przykładowe hiperłącze odnosi się do pliku umieszczonego w tym samym katalogu. Rozpatrzmy teraz przypadek, gdy nasz serwis zawiera kilka działów, a każdy z nich ma zadeklarowany oddzielny katalog. Struktura katalogów została ogólnie przedstawiona na rysunku



Nasze pliki znajdują się wewnątrz katalogów. Zaczniemy od zadeklarowania hipotetycznego łącza pomiędzy plikami *index.html* a *plik.html*:

`Odnosnik do plik.html`

Teraz *plik.html* połączymy z plikiem *oferta.html* znajdującym się w podkatalogu *Oferta*:

`Odnosnik do pliku oferta.html`

W następujący sposób połączymy plik *informacje.html* z plikiem *index.html*:

```
<a href="../../index.html">Odnosnik do index.html</a>
```

Natomiast łączenie pliku *informacje.html* z plikiem *oferta.html* ma taką postać:

```
<a href="../../oferta/oferta.html">Odnosnik do oferta.html</a>
```

Dwie kropki (..) sugerują, by przeglądarka w poszukiwaniu pliku wróciła o jedno "piętro" wyżej. Deklaracja katalogu bezpośrednio po dwóch kropkach (..) mówi, by przeglądarka zeszła o jedno "piętro" niżej w drzewie katalogów — czyli do katalogu sąsiedniego — w poszukiwaniu danego pliku. Oczywiście ciąg ../../ może być dowolnie długi w zależności od potrzeb.

Ćwiczenie 23

1. Utwórz folder „23 Odsyłacze względne i bezwzględne”.
2. W folderze utwórz dokument HTML zawierający w formie listy wypunktowane odnośniki do plików z ćwiczeń 16, 17, 18, 19. Czy występowanie polskich znaków w nazwach plików i katalogów ma znaczenie?
3. Efekt przetestuj w przeglądarce.

Graficzne odsyłacze

Możemy za pomocą znacznika `<a>` sprawić, by nasz obrazek był hiperłączem do innej strony. Budowa takiego graficznego odnośnika jest bajecznie prosta. Odnosnik graficzny składa się z deklaracji odsyłacza oraz znacznika wstawiającego obrazek do strony. Znacznik `<a>` i znacznik `` już znasz! Teraz pozostaje umiejętnie połączyć je ze sobą.

```
<a href="http://nazwa.pl/">

</a>
```

Jak widzisz, część pomiędzy znacznikiem `<a>` zamiast tekstu będącego odnośnikiem zawiera jedynie deklarację umieszczającą w tym miejscu obrazek.

Graficzny odsyłacz domyślnie ma ramkę, która go otacza i szpeci projekt strony. Aby zgodnie ze specyfikacją pozbyć się tego upiększenia, należy skorzystać z kaskadowych arkuszy stylów. Zanim jednak tym powiemy w poniższym listingu zamieściłem gotowy styl usuwający obramowanie z graficznego odsyłacza.

```
<a href="http://nazwa.pl"><imgsrc="obrazek.gif" alt="Do mojej strony" style="border:0;" /></a>
```

Ćwiczenie 24

1. Utwórz folder o nazwie „24 Odsyłacze Graficzne”.
2. Utwórz dokument HTML zawierający dwa odsyłacze graficzne:
 - a. Do Twojej strony na weebly.com,
 - b. Do Twojego adresu e-mail.Pliki graficzne służące do wykonania odsyłaczy skopiuj z sieci lub wykorzystaj własne.
3. Efekt sprawdź w przeglądarce.

Tabele

Wprowadzenie obsługi tabel w języku HTML spowodowało prawdziwą rewolucję. To właśnie dzięki tabelom strony nabrały ciekawego wyglądu i stały się bardziej przestrzenne niż dotychczas. W tym miejscu zaryzykuję stwierdzenie, że tabele stały się tak popularne jako narzędzie wykorzystywane do budowania struktury strony, że w pewnym momencie się "przejadły". Wraz z popularyzacją XHTML i CSS zaprzestano korzystania z tabel i zastąpiono je pozycjonowaniem elementów oraz warstwami. Takie podejście spowodowało znaczące odchudzenie kodu strony, poprawę szybkości wczytywania oraz umożliwiło realizację bardziej skomplikowanych projektów.

Proste tabele

Każda tabela składa się z odpowiedniej kombinacji znaczników `<table></table>`, `<tr></tr>` oraz `<td></td>`. Parametr `border` określa grubość linii tabeli. Oto przykład prostej symetrycznej tabeli.

```
<table border="5">
<tr>
<td>Pierwsza komórka tabeli</td>
<td>Druga komórka tabeli</td>
</tr>
<tr>
<td>Trzecia komórka tabeli</td>
<td>Czwarta komórka tabeli</td>
</tr>
</table>
```

Konstrukcja tabeli jest stosunkowo łatwa do opanowania, jednak wymaga, by każdy znacznik wprowadzać bardzo uważnie. Mała pomyłka, np. brak zamknięcia znacznika, może spowodować błędne wyświetlenie całości.

Znacznik `<table></table>` określa ogólne ramy tabeli, natomiast za pomocą `<tr></tr>` wstawiamy jej wiersze. Każdy wiersz to jedna para znaczników `<tr></tr>`. Przykładowy listing zawiera dwie pary tych znaczników, co oznacza, że tabela ma dwa wiersze. Wewnątrz wiersza znajdują się komórki, które są tworzone za pomocą znacznika `<td></td>`. Jedna para znaczników odpowiada jednej komórce tabeli. Przykładowa tabela zawiera po dwie komórki w każdym wierszu, co oznacza, że ma ona dwa wiersze, a w każdym z nich dwie komórki.

Bardzo często przy tworzeniu tabel zdarza się, że jeden wiersz zawiera opis zawartości — nagłówek tabeli. Zazwyczaj chcemy, by ten element wyróżniał się na tle reszty, np. poprzez pogrubioną czcionkę i wyśrodkowany tekst. Na stronie WWW efekt ten można uzyskać, stosując zamiast znacznika `<td></td>` inny znacznik: `<th></th>`. Jedyna różnica polega na tym, że zawartość takiej komórki zostanie przedstawiona za pomocą pogrubionej czcionki i wyśrodkowanego tekstu. Stosowanie tego znacznika ma również znaczenie, gdy chcesz w przyszłości upiększyć tabelę przy użyciu kaskadowych arkuszy stylów.

Ćwiczenie 25

1. Utwórz folder „25 Prosta tabela”.
2. Korzystając z pliku wzorcowego (Ćwiczenie 2). Utwórz tabelę jak na rysunku.
Odszukaj informację jak wyróżnić wiersz i kolumnę w tabeli.

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

3. Dokument sprawdź w przeglądarce.

Wymiarowanie tabel

Każda tabela publikowana za pomocą znaczników języka HTML domyślnie dopasowuje się do swojej zawartości. Mówiąc prościej, wymiary całej tabeli i poszczególnych komórek dopasowują się do tego, co w nich zamieściliśmy. Doskonale widać to na rysunku

Pierwsza komórka tabeli	Druga komórka tabeli
Trzecia komórka tabeli	Czwarta komórka tabeli

1	2
3	4

W przeszłości za pomocą atrybutów **width=" "** i **height=" "** mogliśmy kontrolować szerokość oraz wysokość tabeli. W piątej specyfikacji języka HTML uznano, że wspomniane atrybuty są już zbędne, a do wymiarowania tabeli należy stosować kaskadowe arkusze stylów.

Wstępna wersja piątej specyfikacji HTML pozbawia nas również możliwości kontrolowania odstępów pomiędzy komórkami oraz wewnętrznych marginesów we wnętrzu samej komórki. W przeszłości mogliśmy to robić za pomocą atrybutów **cellspacing=" "** oraz **cellpadding=" "**, a teraz używamy do tego celu kaskadowych arkuszy stylów.

Tabele niesymetryczne

Dotychczas opisywane tabele były symetryczne i zawsze zawierały tyle samo komórek w każdym wierszu. Teraz przyszła pora, by zająć się tworzeniem tabel o nieregularnych kształtach — niesymetrycznych. Zacznę od przykładu tabeli niesymetrycznej, byś dokładnie wiedział, o czym piszę. Takie tabele znajdują się na rysunku:

Przykład pierwszy

1	2
3	

Przykład drugi

1	2
	3

Przedstawione działanie na tabelach jest możliwe dzięki dwóm atrybutom języka HTML używanym w znaczniku `<td>`. Mam tutaj na myśli **`rowspan=" "`** i **`colspan=" "`**. Pierwszy z nich rozciąga komórkę na sąsiednie wiersze, natomiast drugi na kolumny. Poniżej zamieszczam przykładowy listing dla tabel z rysunku.

```
<body>
<!--Pierwszy przykład-->
<table border="5">
<tr>
  <td>1</td>
  <td>2</td>
</tr>
<tr>
  <td colspan="2">3</td>
</tr>
</table>
</body>

<body>
<!--Drugi przykład-->
<table border="5">
<tr>
  <td rowspan="2">1</td>
  <td>2</td>
</tr>
<tr>
  <td>3</td>
</tr>
</table>
</body>
```


Przy atrybutach **rowspan=" "** i **colspan=" "** występuje wartość liczbowa, która określa, na ile wierszy lub kolumn ma zostać rozciągnięta formatowana komórka.

Oczywiście tabele mogą przybierać różne wymyślne kształty, a wszystko zależy od pomysłowości webmastera oraz jego znajomości znaczników języka HTML. Pamiętaj, że do tworzenia tabel nie potrzeba Ci już niczego więcej.

Ćwiczenie 26

1. Utwórz folder „25 Tabele niesymetryczne”.
2. Bazując na przykładach utwórz własny przykład tabeli niesymetrycznej.
3. Efekt sprawdź w przeglądarce.

Podpis tabeli

Każda tabela może zawierać podpis, który możemy stworzyć na kilka sposobów, wykorzystując do tego różne znaczniki dostępne we wstępnej wersji piątej specyfikacji języka HTML. Najprostszym sposobem uzyskania podpisu jest element `<caption></caption>`, który powinien znaleźć się pomiędzy znacznikami `<table></table>`.

```
<table>
<caption>Tabela 1.1. Każda tabela może zawierać podpis</caption>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td colspan="2">3</td>
</tr>
</table>
```

Drugim sposobem uzyskania podpisu tabeli jest wykorzystanie omawianego już znacznika `<figure></figure>` oraz `<figcaption></figcaption>`. Przykład takiego działania znajduje się poniżej. Natomiast na rysunku 3.28 zamieściłem efekt działania obu metod.

```
<figure>
<figcaption>Tabela 1.1. Każda tabela może zawierać podpis</figcaption>
<table>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td colspan="2">3</td>
</tr>
</table>
</figure>
```

Ćwiczenie 27

1. Utwórz folder „26 Podpis tabeli”.
2. Bazując na ćwiczeniu 25 uzupełnij stworzoną tabelę o podpis.
3. Efekt sprawdź w przeglądarce.