

Temat: Grafika i multimedia na stronach www, modele barw, konwersja między formatami, wstawianie na stronach grafiki, animacji, dźwięku, video

Profil: Technik Informatyk (Technikum Informatyczne)

Klasa: 2

Przedmiot: E14.1

Spis treści

1	Dodawanie obrazków	2
1.1	Element <code>img</code>	2
2	Dołączanie rysunków przy użyciu znacznika <code><figure></code> , <code><figcaption></code>	3
3	Mapy odsyłaczy (ang. <i>imagemap</i>).....	5
3.1	Części mapy obrazu	5
4	Kolory	6
4.1	Wpisywanie wartości RGB do arkuszy stylów	9
5	Multimedia. Szkielet strony	10
6	Wstęp.....	10
7	Kontenery wideo	10
8	Kodeki wideo	12
8.1	H.264	12
8.2	Theora	13
8.3	VP8.....	13
9	Kodeki audio	14
9.1	MPEG-1 Audio Layer 3.....	15
9.2	Advanced Audio Coding	15
9.3	Vorbis.....	16
10	Co działa w internecie.....	16
11	AUDIO	17
12	VIDEO	19
13	Serwer i <code><video></code>	21

1 Dodawanie obrazków

Obrazki mogą być na stronach internetowych wykorzystywane na kilka sposobów:

- Jako zwykle ilustracje – obrazek może zostać wykorzystany na stronie w podobny sposób jak w druku jako statyczny obraz dodający informacje, takie jak logo firmowe czy ilustracja tekstu.
- Jako odnośnik – obrazek może zostać wykorzystany jako odnośnik do innego dokumentu dzięki umieszczeniu go w elemencie kotwicy.
- Jako mapa obrazu – mapa obrazu to jeden obrazek, który zawiera kilka odnośników (obszarów) prowadzących do innych dokumentów.

Żeby obrazek mógł zostać wyświetlony w treści strony, musi on być w formacie GIF (.gif), JPG (.jpg, .jpeg) lub PNG (.png).

1.1 Element *img*

`` (XHTML) `` (HTML)

Dodaje obrazek wewnętrzny – wstawiany jest do tekstu w miejscu, w którym chce się, by został wyświetlony.

Atrybuty `src` oraz `alt` są obowiązkowe. Atrybut `src` podaje przeglądarce lokalizację pliku obrazka. Atrybut `alt` zawiera tekst alternatywny, wyświetlany wtedy, gdy obrazek nie jest dostępny.

`width="liczba"` Szerokość obrazka w pikselach
`height="liczba"` Wysokość obrazka w pikselach

Szerokość oraz wysokość każdego obrazka da się podać za pomocą arkuszy stylów.

`<p>Podstawowe cechy wersji 64G to: powrót do starego typu obudowy, przeprojektowania płyta główna – krótsza i przede wszystkim maksymalne cięcie kosztów... jak łatwo się domyślić odbiło się to negatywnie na jakości.</p>`



Podstawowe cechy wersji 64G to: **powrót do star**
to negatywnie na jakości.

2 Dołączanie rysunków przy użyciu znacznika `<figure>`, `<figcaption>`

Oznacza rysunek lub zdjęcie.

`<figcaption>` obejmuje podpis pod rysunkiem

`<figure>` Według specyfikacji HTML5 element definiujący rysunek (ang. *figure*) powinien w działaniu przypominać ilustrację w książce – innymi słowy, obraz nie będący integralną częścią tekstu, który jednak się do niego odnosi. Powinien zawierać znaczniki `<figcaption>`, i element grafiki ``. Stosuje się je do podkreślenia związku pomiędzy obrazem a podpisem

Zwykle tekst powinien opływać rysunek, co oznacza, że można go umieścić w najlepszym dogodnym miejscu wewnątrz treści (zamiast go przypisywać do jakiegoś słowa czy elementu). W pobliżu znacznika rysunku można osadzić podpis.

Stary sposób:

```
<p><span class="">Commodore 64G</span><p>
```

```
<div class="zdjecie">
  
  <p>Po wprowadzeniu odświeżonego stylistycznie komputera Commodore 64C okazało się że fani z nostalgią wspominają stara, obłą obudowę zwana popularnie "bochenkiem".</p>
</div>
```

```
<p>Podstawowe cechy wersji 64G to: powrót do starego typu obudowy, przeprojektowania płyta główna - krótsza i przede wszystkim maksymalne cięcie kosztów... jak łatwo się domyślić odbiło się to negatywnie na jakości.</p>
```

Plik: styl.css

```
...
/* Obstylowuje pływający obszar rysunku. */
.zdjecie
{
float: left;
margin-left: 0px;
margin-top: 0px;
margin-right: 10px;
margin-bottom: 0px;
}

/* Formatuje podpis pod rysunkiem. */
.zdjecie p
{
max-width: 250px;
font-size: small;
font-style: italic;
margin-bottom: 5px;
}
```

Nowy sposób HTML5:

```
<figure class="zdjecie">
  
  <figcaption>Po wprowadzeniu odświeżonego stylistycznie komputera Commodore 64C
okazało się że fani z nostalgią wspominają stara, obłą obudowę zwana popularnie
"bochenkiem".</figcaption>
</figure>
```

Podstawowe cechy wersji 64G to: powrót do starego typu obudowy, przeprojektowania płyta główna - krótsza i przede wszystkim maksymalne cięcie kosztów... jak łatwo się domyślić odbiło się to negatywnie na jakości.

Plik: styl.css

```
/* Obstylowuje pływający obszar rysunku. */
.zdjecie
{
  float: left;
  margin: 0px 10px 0px 0px;    // góra prawo dół lewo
}

/* Formatuje podpis pod rysunkiem. */
.zdjecie figcaption
{
  max-width: 250px;
  font-size: small;
  font-style: italic;
  margin-bottom: 5px;
}
```

W niektórych przypadkach znacznik <figcaption> może zawierać pełen opis obrazka, co sprawia, że atrybut alt staje się zbędny. Wtedy można go usunąć ze znacznika :

```
<figure class="zdjecie">
  
  <figcaption>Po wprowadzeniu ...</figcaption>
</figure>
```

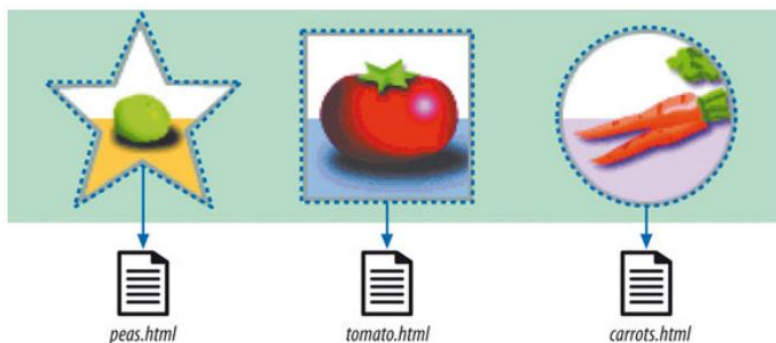
Upewnij się tylko, że nie przypisałeś atrybutowi alt pustego łańcucha znaków, gdyż dla czytnika ekranu oznaczałoby to, że Twoja grafika jest czysto reprezentacyjna i można ją pominąć.

3 Mapy odsyłaczy (ang. *imagemap*)

Pojedynczy obrazek, który zawiera kilka obszarów aktywnych (czy odnośników) wewnątrz.

Kiedy użytkownik kliknie gdzieś w obrębie obrazka, przeglądarka przekazuje współrzędne klikniętego miejsca w pikselach, co z kolei powoduje wygenerowanie

odpowiedniego odnośnika. Kiedy wskaźnik myszy znajduje się nad aktywnym polem (ang. *hotspot*), kursor zmienia kształt, by użytkownik wiedział, że ten obszar obrazka jest odnośnikiem.



Ze względu na nowe techniki oraz nurty w projektowaniu stron internetowych mapy obrazu tracą na popularności. Nie działają zbyt dobrze w przeglądarkach tekstowych i tym samym uznawane są za element obniżający dostępność strony.

3.1 Części mapy obrazu

Mapa obrazu po stronie klienta składa się z trzech komponentów:

- Zwykłego pliku obrazka (.gif, .jpg lub .jpeg bądź .png) umieszczonego wewnątrz elementu `img`.
- Atrybutu `usemap` wewnątrz tego elementu `img` określającego, której mapy należy użyć (każda mapa ma swoją nazwę).
- Elementu `map` będącego pojemnikiem dla pewnej liczby elementów `area`. Każdy element `area` odpowiada obszarowi mapy obrazu, który można kliknąć, i zawiera współrzędne w pikselach oraz informacje o adresie URL dla tego obszaru. Mapa obrazu zostanie omówiona szczegółowo za moment.

map	Informacje o mapie dla mapy obrazu.
<code>name="tekst"</code>	Przestarzała metoda nadawania mapie nazwy.
<code>id="tekst"</code>	Aktualna metoda nadawania mapie nazwy.

area	Zawiera informacje o obszarze mapy obrazu, który można kliknąć.
<code>shape="rect circle poly"</code>	Kształt obszaru będącego odnośnikiem.
<code>coords="liczby"</code>	Współrzędne obszaru będącego odnośnikiem – w pikselach.
<code>href="url"</code>	Plik docelowy dla odnośnika.

`<area />` XHTML Obszar aktywny mapy obrazu
`<area>` HTML

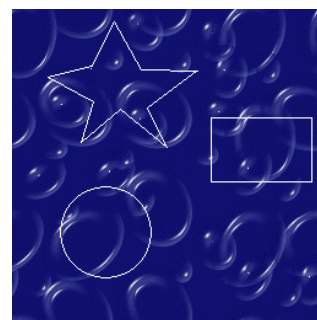
Istnieją narzędzia do generowania map obrazu, dzięki czemu nie trzeba pisać ich ręcznie.

Przykład:

1	<code><map name="obszary" id="obszary"></code>
2	
A	<code><area shape="rect" coords="162,88,244,140" href="prostokat.php" alt="o1" /></code>
B	<code><area shape="circle" coords="76,181,38" href="okrag.php" alt="o2" /></code>
C	<code><area shape="poly" coords="83,10,105,49,151,50,110,80,126,112,88,81,56,108,66,75,29,55,65,46" href="gwiazda.php" alt="o3" /></code>
	<code></map></code>
3	<code></code>

- 1) Początek mapy obrazu. Mapa obrazu ma w tym przypadku nazwę `obszary`. Oba atrybuty (`name` oraz `id`) zostały dodane w celu zachowania zgodności zarówno ze starszymi standardami (`name`), jak i z nowszymi (`id`). Wewnątrz elementu `map` znajdują się elementy `area` reprezentujące każdy z obszarów aktywnych obrazka.
- 2) Każdy element `area` zawiera kilka atrybutów: identyfikator kształtu (`shape`), współrzędne w pikselach (`coords`), adres URL odnośnika (`href`) oraz tekst alternatywny (`alt`). W tej mapie obrazu istnieją trzy obszary odpowiadające prostokątowi, okręgowi oraz wielobokowi, które zostały narysowane na obrazku:

- A) Współrzędne X oraz Y dla prostokąta (`rect`) identyfikują wierzchołki (lewy górny oraz prawy dolny).
- B) Współrzędne dla okręgu (`circle`) identyfikują punkt centralny oraz długość promienia dla obszaru z marchewkami
- C) Lista współrzędnych X oraz Y dla wieloboku (`poly`) identyfikuje każdy z punktów wzdłuż ścieżki kształtu gwiazki zawierającego groszek.



- 3) Element `img` zawiera teraz atrybut `usemap`, który wskazuje przeglądarce, którą mapę należy wykorzystać (`obszary`). W jednym dokumencie HTML można zamieścić kilka map obrazu.

4 Kolory

Istnieją dwa główne sposoby podawania kolorów (w arkuszach stylów):






















































- a) za pomocą zdefiniowanej angielskiej nazwy koloru, np.: `red`; `olive`; `blue`;

Niestety, nie można sobie wymyślić dowolnej nazwy koloru i oczekiwać, że będzie ona działała. Musi to być jedno ze słów kluczowych zdefiniowanych w rekomendacji CSS. W CSS1 oraz CSS2 było szesnaście takich nazw kolorów, które pochodziły ze standardu HTML 4.01. W specyfikacji CSS2.1 dorzucono jeszcze kolor pomarańczowy (`orange`), co razem daje siedemnaście słów kluczowych.



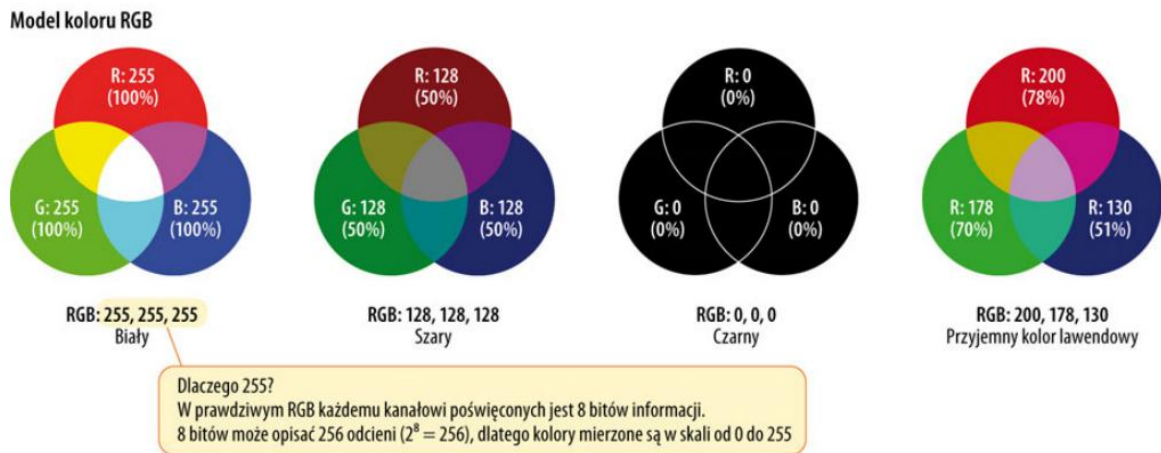
CSS3

Color name	Hex rgb	Decimal	Color name	Hex rgb	Decimal	Color name	Hex rgb	Decimal
<i>aliceblue</i>	#f0f8ff	240, 248, 255	<i>goldenrod</i>	#daa520	218, 165, 32	<i>olivedrab</i>	#6b8e23	107, 142, 35
<i>antiquewhite</i>	#faebd7	250, 235, 215	<i>gray</i>	#808080	128, 128, 128	<i>orange</i>	#ffa500	255, 165, 0
<i>aqua</i>	#00ffff	0, 255, 255	<i>green</i>	#008000	0, 128, 0	<i>orangered</i>	#ff4500	255, 69, 0
<i>aquamarine</i>	#7fffd4	127, 255, 212	<i>greenyellow</i>	#adff2f	173, 255, 47	<i>orchid</i>	#da70d6	218, 112, 214
<i>azure</i>	#f0ffff	240, 255, 255	<i>grey</i>	#808080	128, 128, 128	<i>palegoldenrod</i>	#eee8aa	238, 232, 170
<i>beige</i>	#f5f5dc	245, 245, 220	<i>honeydew</i>	#f0fff0	240, 255, 240	<i>palegreen</i>	#98fb98	152, 251, 152
<i>bisque</i>	#ffe4c4	255, 228, 196	<i>hotpink</i>	#ff69b4	255, 105, 180	<i>paleturquoise</i>	#afeeee	175, 238, 238
<i>black</i>	#000000	0, 0, 0	<i>indianred</i>	#cd5c5c	205, 92, 92	<i>palevioletred</i>	#db7093	219, 112, 147
<i>blanchedalmond</i>	#ffe4cd	255, 235, 205	<i>indigo</i>	#4b0082	75, 0, 130	<i>papayawhip</i>	#ffe4d5	255, 239, 213
<i>blue</i>	#0000ff	0, 0, 255	<i>ivory</i>	#fffff0	255, 255, 240	<i>peachpuff</i>	#ffdab9	255, 218, 185
<i>blueviolet</i>	#8a2be2	138, 43, 226	<i>khaki</i>	#f0e68c	240, 230, 140	<i>peru</i>	#cd853f	205, 133, 63
<i>brown</i>	#a52a2a	165, 42, 42	<i>lavender</i>	#e6e6fa	230, 230, 250	<i>pink</i>	#ffc0cb	255, 192, 203
<i>burlywood</i>	#deb887	222, 184, 135	<i>lavenderblush</i>	#fff0f5	255, 240, 245	<i>plum</i>	#dda0dd	221, 160, 221
<i>cadetblue</i>	#5f9ea0	95, 158, 160	<i>lawngreen</i>	#7cfc00	124, 252, 0	<i>powderblue</i>	#b0e0e6	176, 224, 230
<i>chartreuse</i>	#7fff00	127, 255, 0	<i>lemonchiffon</i>	#ffffac	255, 250, 205	<i>purple</i>	#800080	128, 0, 128
<i>chocolate</i>	#d2691e	210, 105, 30	<i>lightblue</i>	#add8e6	173, 216, 230	<i>red</i>	#ff0000	255, 0, 0
<i>coral</i>	#ff7f50	255, 127, 80	<i>lightcoral</i>	#f08080	240, 128, 128	<i>rosybrown</i>	#bc8f8f	188, 143, 143
<i>cornflowerblue</i>	#6495ed	100, 149, 237	<i>lightcyan</i>	#e0ffff	224, 255, 255	<i>royalblue</i>	#4169e1	65, 105, 225
<i>cornsilk</i>	#fff8dc	255, 248, 220	<i>lightgoldenrodyellow</i>	#fafad2	250, 250, 210	<i>saddlebrown</i>	#8b4513	139, 69, 19
<i>crimson</i>	#dc143c	220, 20, 60	<i>lightgray</i>	#d3d3d3	211, 211, 211	<i>salmon</i>	#fa8072	250, 128, 114
<i>cyan</i>	#00ffff	0, 255, 255	<i>lightgreen</i>	#90ee90	144, 238, 144	<i>sandybrown</i>	#f4a460	244, 164, 96
<i>darkblue</i>	#00008b	0, 0, 139	<i>lightgrey</i>	#d3d3d3	211, 211, 211	<i>seagreen</i>	#2e8b57	46, 139, 87
<i>darkcyan</i>	#008b8b	0, 139, 139	<i>lightpink</i>	#ffb6c1	255, 182, 193	<i>seashell</i>	#fff5ee	255, 245, 238
<i>darkgoldenrod</i>	#b8860b	184, 134, 11	<i>lightsalmon</i>	#ffa07a	255, 160, 122	<i>sienna</i>	#a0522d	160, 82, 45
<i>darkgray</i>	#a9a9a9	169, 169, 169	<i>lightseagreen</i>	#20b2aa	32, 178, 170	<i>silver</i>	#c0c0c0	192, 192, 192
<i>darkgreen</i>	#006400	0, 100, 0	<i>lightskyblue</i>	#87cefa	135, 206, 250	<i>skyblue</i>	#87ceeb	135, 206, 235
<i>darkgrey</i>	#a9a9a9	169, 169, 169	<i>lightslategray</i>	#778899	119, 136, 153	<i>slateblue</i>	#6a5acd	106, 90, 205
<i>darkkhaki</i>	#bdb76b	189, 183, 107	<i>lightslategrey</i>	#778899	119, 136, 153	<i>slategray</i>	#708090	112, 128, 144
<i>darkmagenta</i>	#8b008b	139, 0, 139	<i>lightsteelblue</i>	#b0c4de	176, 196, 222	<i>slategrey</i>	#708090	112, 128, 144
<i>darkolivegreen</i>	#556b2f	85, 107, 47	<i>lightyellow</i>	#ffffe0	255, 255, 224	<i>snow</i>	#ffaafa	255, 250, 250
<i>darkorange</i>	#ff8c00	255, 140, 0	<i>lime</i>	#00ff00	0, 255, 0	<i>springgreen</i>	#00ff7f	0, 255, 127

	<i>darkorchid</i>	#9932cc	153, 50, 204		<i>limegreen</i>	#32cd32	50, 205, 50		<i>steelblue</i>	#4682b4	70, 130, 180
	<i>darkred</i>	#8b0000	139, 0, 0		<i>linen</i>	#faf0e6	250, 240, 230		<i>tan</i>	#d2b48c	210, 180, 140
	<i>darksalmon</i>	#e9967a	233, 150, 122		<i>magenta</i>	#ff00ff	255, 0, 255		<i>teal</i>	#008080	0, 128, 128
	<i>darkseagreen</i>	#8fbc8f	143, 188, 143		<i>maroon</i>	#800000	128, 0, 0		<i>thistle</i>	#d8bfd8	216, 191, 216
	<i>darkslateblue</i>	#483d8b	72, 61, 139		<i>mediumaquamarine</i>	#66cdaa	102, 205, 170		<i>tomato</i>	#ff6347	255, 99, 71
	<i>darkslategray</i>	#2f4f4f	47, 79, 79		<i>mediumblue</i>	#0000cd	0, 0, 205		<i>turquoise</i>	#40e0d0	64, 224, 208
	<i>darkslategrey</i>	#2f4f4f	47, 79, 79		<i>mediumorchid</i>	#ba55d3	186, 85, 211		<i>violet</i>	#ee82ee	238, 130, 238
	<i>darkturquoise</i>	#00ced1	0, 206, 209		<i>mediumpurple</i>	#9370db	147, 112, 219		<i>wheat</i>	#f5deb3	245, 222, 179
	<i>darkviolet</i>	#9400d3	148, 0, 211		<i>mediumseagreen</i>	#3cb371	60, 179, 113		<i>white</i>	ffffff	255, 255, 255
	<i>deeppink</i>	#ff1493	255, 20, 147		<i>mediumslateblue</i>	#7b68ee	123, 104, 238		<i>whitesmoke</i>	#f5f5f5	245, 245, 245
	<i>deepskyblue</i>	#00bfff	0, 191, 255		<i>mediumspringgreen</i>	#00ffa9	0, 250, 154		<i>yellow</i>	ffff00	255, 255, 0
	<i>dimgray</i>	#696969	105, 105, 105		<i>mediumturquoise</i>	#48d1cc	72, 209, 204		<i>yellowgreen</i>	#9acd32	154, 205, 50
	<i>dimgrey</i>	#696969	105, 105, 105		<i>mediumvioletred</i>	#c71585	199, 21, 133				
	<i>dodgerblue</i>	#1e90ff	30, 144, 255		<i>midnightblue</i>	#191970	25, 25, 112				
	<i>firebrick</i>	#b22222	178, 34, 34		<i>mintcream</i>	#f5fffa	245, 255, 250				
	<i>floralwhite</i>	#fffaf0	255, 250, 240		<i>mistyrose</i>	#ffe4e1	255, 228, 225				
	<i>forestgreen</i>	#228b22	34, 139, 34		<i>moccasin</i>	#ffe4b5	255, 228, 181				
	<i>fuchsia</i>	#ff00ff	255, 0, 255		<i>navajowhite</i>	#ffdead	255, 222, 173				
	<i>gainsboro</i>	#dcdcdc	220, 220, 220		<i>navy</i>	#000080	0, 0, 128				
	<i>ghostwhite</i>	#f8f8ff	248, 248, 255		<i>oldlace</i>	#fdf5e6	253, 245, 230				
	<i>gold</i>	#ffd700	255, 215, 0		<i>olive</i>	#808000	128, 128, 0				

b) co jest częściej spotykane – za pomocą wartości liczbowej określającej dany kolor RGB (model koloru z monitorów komputerów), np.: #ff0000;

Angielskie nazwy kolorów są łatwe w użyciu, ale — jak widać — ich liczba jest ograniczona. Z tego powodu najczęściej spotykaną metodą określania koloru jest podanie wartości RGB. Ta opcja umożliwia skorzystanie z milionów dostępnych kolorów.



Komputery tworzą kolory widoczne na monitorze poprzez połączenie trzech kolorów światła: czerwonego, zielonego oraz niebieskiego. Znane jest to pod nazwą modelu koloru RGB, od angielskich nazw tych kolorów — red, green oraz blue. Komputerowi podaje się nasycenie każdego z tych trzech kolorów. Ilość światła w każdym „kanale” jest zazwyczaj opisywana w skali od zera (brak) do dwustu pięćdziesięciu pięciu (pełnia koloru), choć można ją również podać jako wartość procentową. Im bliżej wszystkim trzem wartościom do dwustu pięćdziesięciu pięciu (100%), tym bardziej kolor wynikowy zbliża się do białego.

4.1 Wpisywanie wartości RGB do arkuszy stylów

CSS pozwala na podawanie wartości RGB:

- `rgb (rr,gg,bb)`, np.: `color: rgb(200,178,230)` ;
- wartości procentowe, (rzadziej spotykane): `color: rgb(78%,70%,90%)` ;

- wartość kolorów w systemie szesnastkowym (ang. *hexadecimal*, w skrócie *hex*), które są poprzedzone symbolem #, np.: `color: #C8B2E6` ;



- Jeśli wartość składa się z trzech par takich samych cyfr, np.: `color: #FFCC00` ; lub `color: #993366` ;
można skrócić każdą parę do pojedynczej cyfry: `color: #FC0` ; lub `color: #936` ;

5 Multimedia. Szkielet strony

```
<!DOCTYPE html>
<html lang="pl">

<head>
  <meta charset="UTF-8">
</head>

<body>
</body>

</html>
```

6 Wstęp

Każdy kto kiedykolwiek wszedł na stronę YouTube.com wie, że na stronach internetowych można odtwarzać filmy. Jednak przed pojawieniem się języka HTML5 nie było żadnego dotyczącego tej kwestii standardu. Wszystkie filmy, jakie oglądano na stronach internetowych były odtwarzane dzięki zewnętrznym wtyczkom — np. QuickTime, RealPlayer albo Flash. (W YouTube używany jest Flash.) Wtyczki te tak dobrze integrują się z przeglądarkami, że wielu użytkowników nie wie nawet, że ich używa. Dowiadują się o tym najczęściej dopiero wtedy, gdy chcą obejrzeć film na platformie, która nie obsługuje określonej technologii.

W języku HTML5 zdefiniowano standardową technikę osadzania filmów na stronach przy użyciu elementu HTML5 video. Obsługa tego elementu cały czas się zmienia, chociaż lepiej byłoby chyba powiedzieć, że na razie element `<video>` po prostu nie działa. Przynajmniej nie działa wszędzie. Ale nie panikuj! Jest wiele alternatywnych i awaryjnych rozwiązań.

7 Kontenery wideo

Pliki wideo dla wielu użytkowników to pliki AVI albo MP4. Ale w istocie AVI i MP4 to tylko formaty kontenerów. Podobnie jak plik ZIP może zawierać pliki każdego typu, tak **kontener wideo** służy tylko do przechowywania danych w określony sposób, ale nie determinuje jakiego rodzaju są to dane. (W rzeczywistości jest to bardziej skomplikowane, ponieważ nie wszystkie strumienie wideo są zgodne ze wszystkimi formatami kontenerów, ale na razie to nieważne).

Plik wideo zazwyczaj zawiera kilka ścieżek — ścieżkę wideo (bez dźwięku) oraz przynajmniej jedną ścieżkę dźwiękową (bez wideo). Ścieżki są ze sobą powiązane. Ścieżka dźwiękowa zawiera znaczniki umożliwiające jej synchronizację z filmem. Ścieżki mogą mieć przypisane *metadane*, np. informację o współczynniku kształtu obrazu ścieżki wideo albo o języku ścieżki dźwiękowej. Kontenery również mogą mieć *metadane*, np. tytuł filmu, okładkę, numery odcinków (jeśli jest to np. serial) itd.

Istnieje wiele formatów kontenerów wideo. Do najczęściej używanych należą:

- **MPEG 4** — najczęściej z rozszerzeniem .mp4 lub .m4v. Kontener MPEG 4 powstał na bazie starszego kontenera QuickTime firmy Apple (.mov). Zwiastuny filmów w witrynie Apple wciąż są serwowane w starym kontenerze QuickTime, ale filmy wypożyczane z iTunes są dostarczane w kontenerze MPEG 4.
- **Ogg** — zazwyczaj ma rozszerzenie .ogv. Ogg to otwarty standard przyjazny społeczności open-source, na którym nie ciąży żadne znane patenty. Przeglądarki Firefox 3.5, Chrome 4 oraz Opera 10.5 obsługują bez żadnych dodatków kontener Ogg, format wideo Ogg (o nazwie Theora) oraz format dźwięku Ogg (o nazwie Vorbis). W komputerach osobistych Ogg jest standardowo obsługiwany przez wszystkie najważniejsze wersje Linuksa, a w systemach Mac i Windows można z niego korzystać po zainstalowaniu odpowiednio składników QuickTime lub filtrów DirectShow. Pliki w tym formacie można też odtwarzać przy użyciu znakomitego odtwarzacza VLC na wszystkich platformach.
- **Flash Video** — zazwyczaj z rozszerzeniem .flv. Format Flash Video jest obsługiwany przez wtyczkę Adobe Flash. Przed pojawieniem się wtyczki Flash 9.0.60.184 (znanej też pod nazwą Flash Player 9 Update 3) był to jedyny format kontenera obsługiwany przez Flasha. Obecnie wszystkie najnowsze wersje Flasha obsługują także kontener MPEG 4.
- **WebM** — najczęściej z rozszerzeniem .webm. WebM to bezpłatny i otwarty format kompresji wideo zaprojektowany do użytku z wideo HTML5. Wykorzystuje kodek wideo VP8 i audio Vorbis. Pod względem technicznym jest podobny do formatu o nazwie Matroska. Obsługują go bez żadnych dodatków najnowsze wersje przeglądarek Chromium, Google Chrome, Mozilla Firefox oraz Opera.
- **ASF** — najczęściej z rozszerzeniem .asf. Kontener ASF jest wynalazkiem firmy Microsoft do strumieniowego przesyłania wideo. Zawierał świetne zabezpieczenie DRM, które uniemożliwiało użytkownikom robienie kopii zapasowych legalnie zakupionych licencji. Jeśli ktoś zgubił licencję na jakąś treść, to wg firmy powinien kupić ją jeszcze raz.
- **Audio Video Interleave** — najczęściej z rozszerzeniem .avi. Kontener AVI został wynaleziony przez firmę Microsoft w dawnych czasach, gdy sama informacja, że na komputerach można odtwarzać filmy budziła zdumienie. Oficjalnie format ten nie ma wielu funkcji nowszych kontenerów, choćby nie obsługuje metadanych. Oficjalnie nie obsługuje nawet większości nowoczesnych kodeków wideo i dźwiękowych. Różne firmy rozszerzyły jego funkcjonalność o różne niezgodne ze sobą dodatki i format ten wciąż jest domyślnie stosowany z popularnymi koderami, jak np. MEncoder.

8 Kodeki wideo

Stwierdzenie „oglądam film wideo” zazwyczaj zawiera w sobie informację, że używamy kombinacji strumienia dźwiękowego i wideo. Ale żaden film nie składa się z dwóch oddzielnych plików. Jest tylko jeden film. Może to być plik AVI albo MP4. To są tylko formaty kontenerów, które podobnie jak plik ZIP mogą zawierać różne rodzaje plików. Format kontenera określa sposób przechowywania strumieni dźwiękowego i wideo w jednym pliku.

Gdy oglądasz film, odtwarzacz wykonuje przynajmniej trzy czynności na raz:

1. interpretuje format kontenera, aby dowiedzieć się jakie zawiera ścieżki audio i wideo oraz jak są przechowywane, żeby móc znaleźć dane, które ma zdekodować;
2. dekoduje strumień wideo i wyświetla serię obrazów na ekranie;
3. dekoduje strumień audio i wysyła dane dźwiękowe do głośników.

Kodek wideo to algorytm, przy użyciu którego koduje się strumień wideo, czyli informujący, jak wykonać punkt 2 z powyższej listy. (Słowo „kodek” powstało z połączenia słów „koder” i „dekoder”). Odtwarzacz wideo dekoduje strumień wideo wg algorytmu kodeka wideo i wyświetla na ekranie serię obrazów, czyli klatek. W wielu nowoczesnych kodekach stosowane są różne sztuczki pozwalające zmniejszyć ilość danych potrzebnych do wyświetlania kolejnych klatek na ekranie. Przykładowo zamiast zapisywać całe klatki (jak zrzuty ekranu), przechowywane są tylko różnice między kolejnymi klatkami. W większości filmów obraz między poszczególnymi klatkami zmienia się tylko nieznacznie, co pozwala na uzyskanie wysokiego stopnia kompresji i zmniejszenie rozmiaru plików.

Wyróżnia się kodeki wideo z kompresją stratną i bezstratną. Pliki uzyskane przy użyciu kompresji bezstratnej są zbyt duże, aby nadawały się do użytku w internecie, więc skoncentruję się tylko na kompresji stratnej. Kodek wideo z kompresją stratną to taki, który w procesie kompresji powoduje bezpowrotną utratę części danych. Jest to podobne do kopiowania kasyety wideo. Za każdym razem traci się część informacji wideo, co powoduje pogorszenie jakości obrazu. Na kasecie magnetofonowej pojawiają się szумы, a na kasecie wideo obraz staje się zniekształcony, co widać zwłaszcza w dynamicznych scenach. (Może to mieć miejsce też w przypadku kodowania bezpośrednio ze źródła, gdy wybierze się słaby kodek wideo albo użyje się nieprawidłowych parametrów.) Na szczęście stratne kodeki potrafią bardzo dobrze kompresować filmy i ukrywać niedoskonałości obrazu podczas odtwarzania filmu, dzięki czemu są one trudniejsze do zauważenia.

Istnieje bardzo dużo kodeków wideo. Trzy najlepiej nadające się do użytku w internecie to H.264, Theora oraz VP8.

8.1 H.264

Kodek H.264 jest także znany pod nazwami „MPEG-4 part 10”, „MPEG-4 AVC” oraz „MPEG-4 Advanced Video Coding”. Został opracowany przez grupę MPEG, a jego standard powstał w 2003 r. Jest przeznaczony do użytku zarówno w sprzętach o niskiej przepustowości, ze słabym procesorem (telefony komórkowe) jak i urządzeniach o dużej przepustowości i z wydajnym procesorem (jak nowoczesne komputery osobiste) oraz we wszystkim, co mieści się pomiędzy nimi. Aby było możliwe obsłużenie tak różnorodnych urządzeń, kodek H.264 podzielono na profile, z których każdy zawiera opcjonalne funkcje pozwalające zmienić rozmiar pliku kosztem zwiększonej złożoności. Wyższe profile mają więcej opcjonalnych funkcji, dają lepszą jakość obrazu przy mniejszym rozmiarze pliku, wymagają więcej czasu na kodowanie oraz potrzebują więcej mocy procesora, aby dekodować na bieżąco.

W serwisie YouTube aktualnie kodek H.264 to jest wykorzystywany do kodowania filmów HD, które odtwarzane są poprzez Adobe Flash. Ponadto YouTube udostępnia filmy kodowane kodekiem H.264 dla urządzeń przenośnych, wliczając iPhone'a Apple i telefony z systemem operacyjnym Android firmy Google. Ponadto H.264 jest jednym z kodeków wideo zatwierdzonych w specyfikacji Blu-Ray. Na dyskach Blu-Ray zazwyczaj używany jest wysoki profil H.264.

Większość urządzeń innych niż komputery PC podczas odtwarzania filmów H.264 (dotyczy to także iPhone'ów i odtwarzaczy Blu-Ray) wykorzystuje specjalny układ do dekodowania danych, ponieważ ich procesory są za mało wydajne, aby dekodować wideo na bieżąco. Obecnie nawet niższej klasy karty graficzne dekodują dane H.264. Istnieją konkurujące ze sobą kodery H.264, do których zalicza się także otwarta biblioteka x264. **Standard H.264 jest obciążony patentem**, a licencjami zarządza grupa MPEG LA. Wideo w formacie H.264 można osadzić w najpopularniejszych formatach kontenerów, wliczając MP4 (głównie używany w sklepie iTunes Apple) i MKV (używany głównie przez entuzjastów wideo nie działających w celach komercyjnych).

8.2 Theora

Theora to produkt, który wyewoluował z kodeka VP3 i jest rozwijany przez Xiph.org Foundation. Theora to darmowy kodek nieobciążony żadnymi znanymi patentami oprócz oryginalnych patentów VP3, które są udostępniane nieodpłatnie. Mimo że standard pozostaje w stagnacji od 2004 r., opublikowano wersję 1.0 w listopadzie 2008 r. i 1.1 we wrześniu 2009 r projektu Theora (w skład w którego wchodzi otwarty koder i dekodery referencyjne).

Wideo w formacie Theora można osadzić w dowolnym kontenerze, ale najczęściej można je znaleźć w kontenerach Ogg. Wszystkie najważniejsze dystrybucje Linuksa standardowo obsługują Theorę, a Mozilla Firefox 3.5 obsługuje wideo Theora w kontenerze Ogg. Obsługa oznacza w tym przypadku, że „z technologii można korzystać na wszystkich platformach bez potrzeby instalowania żadnych dodatków”. Wideo w formacie Theora można też odtwarzać w systemach Windows i Mac OS X, ale trzeba zainstalować otwarty dekodery Xiph.org.

8.3 VP8

VP8 to kolejny kodek wideo firmy On2, w której też powstał kodek VP3 (później zamieniony w Theorę). Jakość obrazu jest porównywalna z profilem wysokim H.264 przy jednoczesnym niskim stopniu złożoności procesu dekodowania w porównaniu z tym kodekiem.

W 2010 r. Google wykupiło On2 i opublikowało w otwartym źródle specyfikację kodeka wideo oraz jego przykładowy koder i dekodery. Dodatkowo firma Google „uwolniła” wszystkie patenty uzyskane przez On2 dla VP8, udostępniając je wszystkim bez opłat licencyjnych. (Z patentami jest tak, że nie można ich po prostu opublikować ani anulować, gdy zostaną już komuś przyznane. Jeśli chcemy, aby nasz patent był otwarty, musimy udzielać na niego darmowych licencji i wówczas każdy kto chce może wykorzystywać naszą technologię bez potrzeby negocjowania ceny). **Od 19 maja 2010 r. VP8 jest darmowym kodekiem nieobciążonym żadnymi znanymi patentami**, innymi niż te, które firma On2 (a teraz Google) udostępnia na darmowej licencji.

9 Kodeki audio

Jeśli nie planujesz pozostać w epoce filmowej sprzed 1927 r., w swoich filmach będziesz też potrzebować ścieżki dźwiękowej. Podobnie jak kodeki wideo, kodeki audio są algorytmami kodowania strumienia dźwiękowego. I podobnie jak w przypadku kodeków wideo istnieją stratne i bezstratne kodeki audio. Ścieżki dźwiękowe tworzone przy użyciu bezstratnych algorytmów są zbyt duże, aby używać ich w internecie. Dlatego skoncentrujemy się tylko na kodekach stratnych.

W istocie algorytmy stratne dzielą się na kilka kategorii. Dźwięku używa się także w sytuacjach, w których obraz nie jest potrzebny (np. w rozmowach telefonicznych) i istnieje cała kategoria kodeków audio zoptymalizowanych do kodowania mowy. Nie należy ich używać np. do zgrzywania płyt CD na komputer, ponieważ śpiew artysty brzmiałby jak głos czterolatka śpiewającego do mikrofonu. Ale używa się ich w PBX Asterisk, ponieważ transfer jest drogi, a te kodeki bardzo dobrze kompresują ludzki głos radykalnie zmniejszając ilość danych, które trzeba przesłać. Ze względu na brak obsługi w przeglądarkach i wtyczkach do przeglądarek algorytmy kompresji mowy nigdy nie znalazły zastosowania w sieci. Dlatego skoncentrujemy się tylko na kodekach audio ogólnego przeznaczenia.

Jak napisałem wcześniej, gdy „oglądasz film wideo”, twój komputer wykonuje przynajmniej trzy czynności na raz:

1. interpretuje format kontenera;
2. dekoduje strumień wideo;
3. dekoduje strumień audio i wysyła dane dźwiękowe do głośników.

Kodek audio umożliwia wykonanie trzeciego punktu z tej listy — dekodowanie strumienia audio i zamienienie go na cyfrowe fale zamieniane następnie w dźwięk przez głośniki. Podobnie jak w przypadku kodeków wideo, projektanci stosują rozmaite sztuczki, aby zminimalizować ilość danych w strumieniu dźwiękowym. A ponieważ mowa jest o stratnych kodekach audio, w trakcie procesu nagrywania → kodowanie → dekodowanie → słuchanie następuje strata jakości. Różne kodeki usuwają różne informacje, ale wszystkie działają w tym samym celu: zmniejszyć ilość danych jednocześnie ukrywając ich ubytki tak, aby ludzkie ucho nie mogło ich wychwycić.

Jednym z pojęć, które nie dotyczą wideo są kanały. Dźwięk wysyłamy do głośników, prawda? A ile masz u siebie głośników? Jeśli siedzisz przy komputerze, to możliwe że masz tylko dwa głośniki: jeden po lewej i jeden po prawej stronie. Ja mam trzy: lewy, prawy i jeden na podłodze. Tak zwane systemy surround mogą mieć nawet sześć i więcej głośników odpowiednio rozmieszczonych w pomieszczeniu. Do każdego głośnika przekazywany jest określony kanał nagrania. Ogólnie rzecz biorąc chodzi o to, że słuchacz jest otoczony z sześciu stron głośnikami, z których każdy obsługuje inny kanał dźwiękowy, co daje wrażenie jakby się znajdowało w centrum słuchanych wydarzeń. Czy to działa? Rynek ten jest wart miliardy dolarów, a więc chyba tak.

Większość ogólnych kodeków audio obsługuje dwa kanały dźwiękowe. Podczas nagrywania dźwięk jest rozdzielany na lewy i prawy kanał. Podczas kodowania oba kanały są zapisywane w jednym strumieniu dźwiękowym, a w czasie dekodowania każdy z nich zostaje odkodowany i wysłany do odpowiedniego głośnika. Niektóre kodeki audio obsługują więcej kanałów i zapamiętują który kanał jest który, dzięki czemu odtwarzacz może je kierować do właściwych głośników.

Istnieje bardzo dużo kodeków audio. Czy wspominałem, że kodeków wideo też jest dużo? Zapomnij o tym. Kodeków audio jest cała masa, ale w sieci liczą się tak naprawdę tylko trzy: MP3, AAC oraz Vorbis.

9.1 MPEG-1 Audio Layer 3

MPEG-1 Audio Layer 3 to popularny format MP3.

Pliki MP3 mogą zawierać maksymalnie 2 kanały dźwiękowe. Mogą mieć różną szybkość transmisji bitów: 64 kbps, 128 kbps, 192 kbps i wiele innych od 32 do 320. Wyższa szybkość transmisji bitów oznacza większy rozmiar pliku i lepszą jakość dźwięku, chociaż parametry te nie są ze sobą powiązane liniowo. (128 kbps brzmi lepiej niż dwa razy 64 kbps, a 256 kbps nie brzmi tak dobrze, jak dwa razy 128 kbps.) Ponadto w formacie MP3 możliwe jest kodowanie ze zmienną szybkością transmisji bitów, co oznacza, że niektóre części strumienia są skompresowane bardziej, a inne mniej. Przykładowo okresy ciszy można zakodować w gorszej jakości, a gdy rozpocznie się część skomplikowanej formy muzycznej jakość tę można polepszyć. W formacie MP3 możliwe jest też kodowanie ze stałą szybkością transmisji bitów, co oczywiście nazywa się kodowaniem ze stałą szybkością transmisji bitów.

Standard MP3 nie określa szczegółowo sposobu kodowania dźwięku (ale za to zawiera dokładne instrukcje, jak go dekodować). Dlatego w koderach używane są różne psychoakustyczne modele pozwalające uzyskać bardzo różne wyniki, ale wszystkie można odkodować w tym samym odtwarzaczu. Najlepszym darmowym koderem jest otwarty projekt LAME. Jest on prawdopodobnie najlepszym koderem w ogóle do wszystkich zastosowań oprócz najniższych szybkości transmisji bitów.

Format MP3 (ustandaryzowany w 1991 r.) jest obciążony patentami, co wyjaśnia dlaczego nie można odtwarzać plików MP3 w Linuksie, chyba że zainstaluje się dodatkowe oprogramowanie. Praktycznie każdy przenośny odtwarzacz muzyki obsługuje pliki MP3, a ponadto format ten może być też wykorzystywany w kontenerach wideo. Adobe Flash odtwarza zarówno pliki MP3 jak i strumienie audio w formacie MP3 w kontenerze wideo MP4.

9.2 Advanced Audio Coding

Kodek Advanced Audio Coding jest szerzej znany pod nazwą AAC. Standard tego formatu opracowano w 1997 r., ale czasy świetności zaczęły się, gdy firma Apple zaczęła go stosować jako domyślny format w sklepie iTunes. Pierwotnie wszystkie pliki AAC „kupione” w iTunes były zaszyfrowane przy użyciu algorytmu DRM Apple o nazwie FairPlay. Obecnie niektóre utwory w tym sklepie są dostępne bez zabezpieczeń w plikach AAC. Apple nazywa je „iTunes Plus”, bo to brzmi o wiele lepiej niż nazywanie wszystkiego „iTunes Minus”. **Format AAC jest obciążony patentami**, a ceny licencji można znaleźć w internecie.

Format AAC pozwala uzyskać lepszą jakość dźwięku od MP3 przy takiej samej szybkości transmisji bitów oraz można go używać do kodowania z dowolną szybkością transmisji bitów. (W MP3 liczba możliwych szybkości jest ograniczona, a górną granicę stanowi 320 kbps.) W AAC można zakodować do 48 kanałów dźwiękowych, chociaż nikt tego nie robi. Kolejną cechą odróżniającą AAC od MP3 są profile, które są czymś podobnym do profili w H.264 i zostały utworzone w tym samym celu. Profil o niskim poziomie złożoności jest przeznaczony do odtwarzania dźwięku na bieżąco na urządzeniach ze słabszymi procesorami, natomiast wyższe profile dają dźwięk lepszej jakości przy takiej samej szybkości transmisji bitów, ale bardziej pracochłonnym kodowaniu i dekodowaniu.

Aktualnie wszystkie produkty formy Apple, wliczając iPody, AppleTV oraz QuickTime obsługują wybrane profile AAC zarówno w samodzielnych plikach jak i strumieniach audio w kontenerze wideo MP4. Adobe Flash obsługuje wszystkie profile AAC w MP4, podobnie jak otwarte odtwarzacze wideo MPlayer i VLC. Do kodowania można używać otwartej biblioteki FAAC. Obsługę jej można włączyć za pomocą opcji kompilacji w mencoder i ffmpeg.

9.3 Vorbis

Vorbis jest często nieprawidłowo nazywany „Ogg Vorbis”. (Ogg to tylko format kontenera, a strumienie audio Vorbis można osadzać także w innych kontenerach.) **Vorbis nie jest obciążony żadnymi znanymi patentami**, dzięki czemu standardowo obsługują go wszystkie najważniejsze wersje Linuksa i urządzenia przenośne z otwartym oprogramowaniem układowym Rockbox. Przeglądarka Mozilla Firefox 3.5 obsługuje pliki dźwiękowe Vorbis w kontenerze Ogg oraz wideo Ogg ze ścieżką dźwiękową w formacie Vorbis. Także telefony komórkowe z systemem Android odtwarzają pliki dźwiękowe w formacie Vorbis. Strumienie dźwiękowe Vorbis są najczęściej osadzane w kontenerach Ogg i WebM, ale można ich używać także w kontenerach MP4 i MKV(a po zastosowaniu pewnych sztuczek także w AVI). Vorbis obsługuje dowolną liczbę kanałów dźwiękowych.

Istnieją otwarte kodery i dekodery Vorbis, np. OggConvert (koder), ffmpeg (dekodek), aoTuV (koder) oraz libvorbis (dekoder). Dostępne są też składniki QuickTime dla systemu Mac OS X i filtry DirectShow dla systemu Windows.

10 Co działa w internecie

W języku HTML5 zdefiniowano element `<video>` służący do osadzania filmów na stronach internetowych. W swoich produkcjach możesz używać dowolnych kodeków audio i wideo oraz jakiego chcesz kontenera. Jeden element `<video>` może być powiązany z wieloma plikami wideo i wówczas przeglądarka wybierze pierwszy, który potrafi odtworzyć. Ty musisz wiedzieć, jakie kodeki i kontenery obsługuje każda przeglądarka.

Aby dotrzeć do jak największej liczby odbiorców, należy:

1. utworzyć wersję filmu w WebM (VP8 + Vorbis).
2. utworzyć wersję filmu przy użyciu podstawowego kodeka H.264 i AAC (niski profil) w kontenerze MP4.
3. utworzyć wersję przy użyciu kodeków Theora i Vorbis w kontenerze Ogg.*
4. zdefiniować łącza do wszystkich trzech plików wideo w jednym elemencie `<video>` i dodać awaryjnie możliwość użycia odtwarzacza opartego na Flashu.

* WebM i H.264 są wystarczająco obsługiwane. Jeśli więc nie interesują cię już przeglądarki Firefox 3.5 i Opera 10.5, możesz pominąć wersję z kodekiem Theora.

11 AUDIO

W języku HTML5 znacznik `<audio>` służy do odtwarzania plików muzycznych bez użycia dodatkowych wtyczek (np: Adobe Flash Player czy Microsoft Silverlight).

Obecnie jest obsługiwane przez każdą nowoczesną przeglądarkę, jednakże każda z nich wymaga innego formatu pliku:

Google Chrome – .mp3 .wav .ogg .acc .webm

Firefox – .wav .ogg .webm .opus

Opera – .wav .ogg .webm

Internet Explorer – .mp3 .acc

Muzykę na stronach można umieszczać na dwa sposoby. W obu przypadkach używa się elementu `<audio>`. Jeśli mamy tylko jeden plik audio, to możemy po prostu wpisać ścieżkę do niego w atrybucie `src`. W taki sam sposób wstawia się na strony obrazy przy użyciu elementu ``.

Sam znacznik `<audio>` może posiadać atrybuty:

`autoplay` – Automatyczne uruchomienie odtwarzania muzyki.

`loop` – Zapętlenie odtwarzania.

`controls` – Wyświetlanie kontrolki sterowania odtwarzaczem.

Odtwarzacze trzech różnych przeglądarek:

Internet Explorera (u góry),

Google Chrome (środek)

Firefoksa (u dołu).



`autobuffer` – Buforowanie pliku audio

`preload="tryb"` – Jak załadować nagranie.

Tryb:

`none` – nie ładuje automatycznie w tle muzyki.

`metadata` – ściąga tylko informacje o pliku (czas trwania itp.).

`auto` – ładuje automatycznie w tle wszystkie dane oraz muzykę, zaraz po załadowaniu strony. Można użyć samo `preload` jako atrybut.

`muted` – Domyślne wyciszenie odtwarzanej muzyki

Przykład 1:

```
<!-- Jeden plik -->
<audio src="swvx.mp3" controls>
Twoja przeglądarka nie obsługuje tagu audio.
</audio>
```

W praktyce pojawi się jednak problem, związany z implementacją różnych wersji kodeków w przeglądarkach (np. Vorbis, Advanced Audio Coding, MPEG-1 Audio Layer 3). Z tego powodu, trzeba aktualnie definiować dodatkowe zasoby w tagu `source`. Obsługuje on atrybuty:

- `media` – zdefiniowanie typu zasobu. Domyślnie odpowiada wszystkim typom mediów,
- `src` – adres URL do pierwotnego, ładowanego źródła,
- `type` – definiuje typ zasobu, pozwalając na określenie przeglądarce, czy może odtworzyć go, bez konieczności pobierania. Musi być pod postacią prawidłowego typu MIME. Opcjonalnie, wewnątrz niego można zdefiniować parametr `codecs`, określający jak zakodowany jest zasób.

W języku HTML5 można połączyć wszystkie formaty muzyki z jednym elementem: służy do tego element `<source>`. Każdy element `<audio>` może zawierać dowolną liczbę elementów `<source>`. Przeglądarka przejrzy tę listę źródeł i wybierze pierwsze, które potrafi odtworzyć.

Nasuwa się kolejne pytanie: skąd przeglądarka wie, którą muzykę będzie w stanie odtworzyć? W najgorszym przypadku wczyta każdy z nich i spróbuje je odtworzyć po kolei. Ale to byłoby wielkim marnotrawstwem transferu. Dużo zyskasz, jeśli z góry dostarczysz przeglądarce odpowiednich informacji. Służy do tego atrybut `type` elementu `<source>` wg. wzorca:

`"audio/format"`

np: `type="audio/mpeg"` dla Mp3.

- Dla `.mp3` w miejsce format wpisujemy `mpeg`
- Dla `.wav` w miejsce format wpisujemy `wav`
- Dla `.ogg` w miejsce format wpisujemy `ogg`
- Dla `.acc` w miejsce format wpisujemy `aac`
- Dla `.webm` w miejsce format wpisujemy `webm`
- Dla `.opus` w miejsce format wpisujemy `opus`

Przykład 2:

```
<!-- Alternatywne formaty AUDIO -->
<audio controls>
  <source src="swvx.mp3" type="audio/mpeg" />
  <source src="swvx.ogg" type="audio/ogg" />
  Twoja przeglądarka nie obsługuje tagu audio.
</audio>
```

Źródło: <http://kurshtml5.edu.pl/audio/>

12 VIDEO

W języku HTML5 znacznik `<video>` służy do odtwarzania filmów bez użycia dodatkowych wtyczek (np: Adobe Flash Player czy Microsoft Silverlight).

Obecnie jest obsługiwane przez każdą nowoczesną przeglądarkę, jednakże każda z nich wymaga innego formatu pliku:

- Google Chrome – .webm .ogv
- Firefox – .webm .ogv
- Opera – .webm .ogv
- Internet Explorer – .mp4

W języku HTML5 filmy na stronach można umieszczać na dwa sposoby. W obu przypadkach używa się elementu `<video>`. Jeśli mamy tylko jeden plik wideo, to możemy po prostu wpisać ścieżkę do niego w atrybucie `src`. W taki sam sposób wstawia się na strony obrazy przy użyciu elementu ``.

Sam znacznik `<video>` może posiadać atrybuty:

- `src` ścieżka do pliku
- `width i height` szerokość i wysokość filmu, niepodanie tych parametrów powoduje przyjęcie przez przeglądarkę rozmiarów filmu dla odtwarzacza.
- `autoplay` – Automatyczne uruchomienie odtwarzania filmu.
- `loop` – Zapętlenie odtwarzania.
- `controls` – Wyświetlanie kontrolki sterowania odtwarzaczem.
- `autobuffer` – Buforowanie pliku audio

`preload="tryb"` – Jak załadować nagranie.

Tryb:

- `none` – nie ładuje automatycznie w tle filmu.
- `metadata` – ściąga tylko informacje o pliku (czas trwania itp.).
- `auto` – ładuje automatycznie w tle wszystkie dane oraz film, zaraz po załadowaniu strony.
Można użyć samo `preload` jako atrybut.

- `muted` – Domyślne wyciszenie odtwarzanego filmu
- `poster` – Pozwala na umieszczenie obrazka który pojawi się w oknie odtwarzacza zanim film zostanie uruchomiony.

Przykład 1:

```
<!-- Jeden plik -->
<video src="canvas.mp4" controls>
Twoja przeglądarka nie obsługuje tagu video.
</video>
```

Tak jak w przyp. audio i tu w praktyce pojawi się jednak problem, związany z implementacją różnych wersji kodeków w przeglądarkach (np. Ogg Theora, H.264, VP8[WebM]). Z tego powodu, trzeba aktualnie definiować dodatkowe zasoby w tagu `source`. Obsługuje on atrybuty:

- `media` – zdefiniowanie typu zasobu. Domyślnie odpowiada wszystkim typom mediów,
- `src` – adres URL do pierwotnego, ładowanego źródła,
- `type` – definiuje typ zasobu, pozwalając na określenie przeglądarce, czy może odtworzyć go, bez konieczności pobierania. Musi być pod postacią prawidłowego typu MIME. Opcjonalnie, wewnątrz niego można zdefiniować parametr `codecs`, określający jak zakodowany jest zasób.

W języku HTML5 można połączyć wszystkie formaty video z jednym elementem: służy do tego element `<source>`. Każdy element `<video>` może zawierać dowolną liczbę elementów `<source>`. Przeglądarka przejrzy tę listę źródeł i wybierze pierwsze, które potrafi odtworzyć.

Nasuwa się kolejne pytanie: skąd przeglądarka wie, który film będzie w stanie odtworzyć? W najgorszym przypadku wczyta każdy z nich i spróbuje je odtworzyć po kolei. Ale to byłoby wielkim marnotrawstwem transferu. Dużo zyskasz, jeśli z góry dostarczysz przeglądarce odpowiednich informacji. Służy do tego atrybut `type` elementu `<source>` wg. wzorca:

"video/**format**"

np: `type="video/webm"` dla WebM.

- Dla `.webm` w miejsce format wpisujemy `webm`
- Dla `.ogv` w miejsce format wpisujemy `ogg`
- Dla `.mp4` w miejsce format wpisujemy `mp4`

Uwaga! Format pliku `.webm` musi być podany jako pierwszy. Nie podanie ścieżki do formatu `.webm` może powodować błędy w odtwarzaniu filmu.

Przykład 2:

```
<!-- Alternatywne formaty VIDEO -->
<video controls>
  <source src="canvas.webm" type="video/webm" />
  <source src="canvas.ogv" type=" video/ogg" />
  <source src="canvas.mp4" type=" video/mp4" />
  Twoja przeglądarka nie obsługuje tagu video.
</video>
```

Źródło: <http://kurshtml5.edu.pl/video/>

Do audio i video można dodać również jako alternatywę:

- Obsługę wtyczki flash
- Sterowanie odtwarzaczem za pomocą javascript

13 Serwer i <video>

Jednak właściwe ustawienie w atrybucie `type` w kodzie HTML to za mało. Dodatkowo trzeba się upewnić, że serwer dodaje odpowiedni typ MIME w nagłówku HTTP Content-Type.

Jeśli używasz serwera Apache albo jakiegoś pochodnego, to możesz w ogólnym pliku `httpd.conf` albo w pliku `.htaccess` znajdującym się w folderze z filmami użyć dyrektywy `AddType`. (Jeżeli używasz innego serwera, poszukaj w dokumentacji informacji jak ustawić nagłówek HTTP Content-Type dla wybranych typów plików.)

Zawartość pliku `.htaccess`:

```
AddType video/ogg .ogv
AddType video/mp4 .mp4
AddType video/webm .webm
```

Pierwszy z powyższych wierszy kodu dotyczy plików wideo w kontenerze Ogg.

Drugi dotyczy plików wideo w kontenerze MPEG-4.

Trzeci dotyczy WebM.

Ustawiamy raz i zapominamy. Jeśli tego nie zrobisz, twoje filmy w niektórych przeglądarkach nie będą dały się odtworzyć, nawet mimo ustawienia typu MIME w atrybucie `type` w elemencie HTML.

Źródło: <http://www.bt4.pl/kursy/html5-k/html5-rzeczowo/>