

HTML5 - informacje

Źródło

<http://www.w3.org/TR/html5>

Struktura

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Example document</title>
  </head>
  <body>
    <p>Example paragraph</p>
  </body>
</html>
```

Internacjonalizacja

internationalization --> i18n

Localization --> L10n

i18n --> iso-8859-* (a.k.a ISO Latin-2 / środkowoeuropejskie)

i18n --> utf-8 (milion znaków, każdy 8-bit lub 24-bit)

Elementy

<header> - nagłówek strony, obejmuje wewnątrz np. nawigację, **<h1>** (jeśli **h1** jest tytułem strony, a nie tytułem rozdziału), **<p>**, element odnośników do innych podstron **<nav>** - wszystko, co będzie się powtarzać na każdej stronie serwisu;

<footer> - stopka, czyli informacje o prawach autorskich, wersji, data, itp.; może zawierać **<p>** - wszystko, co będzie powtarzać się na każdej innej stronie serwisu;

<main> - oznacza semantyczną sekcję, która jest główną częścią strony, głównym wątkiem, tzn. zawiera elementy, które nie powtarzają się na innych stronach w serwisie (czyli nie obejmuje nagłówek, stopki i menu); na każdej stronie istnieje tylko jedna sekcja **<main>** i nie może być wewnątrz **<article>**, **<aside>**, **<nav>**, **<footer>**, **<header>**.

<article> - obejmuje np. cały artykuł w gazecie; artykuł podzielony jest na rozdziały **<section>**, a wewnątrz rozdziałów mogą być **<h1>**, **<p>**, ****;

<section> - coś w typie rozdziału, zawiera w sobie **<h1>**, **<p>**, itp. Nie stosuje się do samych fotek (wtedy lepiej użyć **<div>** lub bezpośrednio jako **** lub **<figure>**;

<div> - unikaj jeśli możesz; partie tekstu umieszczaj w **<nav>**, **<article>** i **<section>**;

<aside> - notka poboczna, może obejmować np. **<h1>**, **<p>**;

<figure> - obrazek z podpisem lub cytatem z podaniem źródła; może zawierać **<figcaption>**;

<video>, **<audio>** - muszą mieć atrybut **src="plik.mp4"**; może mieć atrybuty: **autoplay**, **controls** (atrybuty te nie mają wartości) oraz **preload="auto"** (materiał video/audio ładowany jest od razu); możliwe formaty kontenery: **mp4**, **webm**, **ogg** (*oga/ogv*), **flac**, **mp3** (*patent uległ przedawnieniu*), **aac**; akceptowane kodery: *Theora*, *VP8*, **VP9** (*Video Processor 8*), *H.264*, **AV1** (*AOMedia Video 1*), *Vorbis*, *Opus*. Uwaga: Kontener ***.mkv** nie jest już akceptowany nawet jeśli zawiera materiał kodowany „wolnym oprogramowaniem”. Jego „wolną” wersją jest „**webm**”.
Dobre połączenia: **webm** (AV1 + Opus), **mp4** (HEVC + AAC), **mp4** (H.264 [x264] + AAC).

**** - ostrzeżenia, wypuklenia (mocniejsze niż ****);

<canvas> - alternatywa dla SVG, pojemnik / miejsce na obiekt z animacją wektorową (np. napisaną w *JavaScript*), gra, może zastąpić flasha; animacja jest w locie przerabiana na

bitmapę; ma atrybuty *width* oraz *height*;

Atrybuty

target - jest na nowo dopuszczony w HTML5, ale powinno się go używać w bardzo rzadkich sytuacjach z dwóch względów: przeglądarki mobilne nie obsługują zakładek, a przeglądarki stacjonarne pozostawiają te ustawienia użytkownikowi (i tak powinno pozostać);

translate="yes/no" - informacja, czy przeglądarka może próbować tłumaczyć tekst na język użytkownika;

SVG

Znacznik `<svg>` jest kontenerem zawierającym figury. Atrybut „*cx*” oraz „*cy*” odnoszą się do położenia koła (`circle`) względem pojemnika `<svg>`, zaś atrybuty „*rx*” / „*ry*” określają zaokrąglenie rogów w prostokącie.

```
<svg width="300px" height="100px">
<circle r="50px" cx="50px" cy="50px" fill="green">
<ellipse cx="80px" cy="200px" rx="100px" ry="50px" style="fill: red; stroke: black;">
<rect width="300px" height="200px" y="30px" x="30px" style="stroke-width:5px;
stroke:#000000; fill:#ff0000; fill-opacity:0.7;">
<line x1="0" y1="0" x2="260px" y2="340px" stroke="red" stroke-width="3px">
<text x="30px" y="50px" style="font-family:DejaVu; font-size:30px;">Jakiś tekst</text>
</svg>
```

```

```

Info: SVG korzysta z XML, zaś Canvas korzysta z JavaScript.

Powiększanie, przenikanie i transformacje

Uwaga: Przenikanie nie działa w przypadku przejścia z „*display: none*” do „*display: block*” i odwrotnie, ani w przypadku każdej innej wartości „*display*” (blok albo jest, albo nie ma go... nic pośredniego). Rozwiązaniem tego problemu jest (zamiast „*display: none;*”) zdefiniowanie bloku o parametrach: *width: 0px; height: 0px*. Można także zamiast tego wykorzystać „*visibility: collapse*”.

Po najechnaniu myszką na element `<div>` powiększy się od w ciągu 2 sekund do rozmiaru 100px i zmieni kolor na zielony w ciągu 4 sekund:

```
div
{width: 50px;
background-color: red;
transition: width 2s, background-color 4s;}
```

```
div:hover
{width: 100px;
background-color: green;}
```

Do obracania elementów służy atrybut „*transform*”:

```
div
{transform: rotate(7deg);}
```

Aby uzyskać płynny efekt animacji:

```
div
{animation: wlasnaNazwaEfektu 10s infinite linear;}
```

Uwaga: Zamiast wartości „*infinite*” możemy dać „*normal*” (wykona się tylko raz) lub dowolną liczbę powtórzeń, np. „3”. Poniższa wartość procentowa dotyczy części tych 10 sekund:

```
@keyframes wlasnaNazwaEfektu {
0% {transform: rotate(0deg);}
10% {transform: rotate(-25deg);}
```

```
20% {transform: rotate(47deg);}
30% {transform: rotate(-125deg);}
40% {transform: rotate(-25deg);}
50% {transform: rotate(25deg);}
60% {transform: rotate(165deg);}
70% {transform: rotate(42deg);}
80% {transform: rotate(180deg);}
90% {transform: rotate(-300deg);}
100%{transform: rotate(360deg);}
}
```

Gradientsy:

div

```
{background-image: linear-gradient(red, yellow);}
```

lub

```
{background-image: linear-gradient(to right, red, yellow);}
```

lub

```
{background-image: radial-gradient(red 5%, yellow 15%, green 60%);}
```

lub z przezroczystością:

```
{background-image: linear-gradient(to right, rgba(255,0,0,0), rgba(255,0,0,1));}
```

Uwaga: można podać więcej kolorów niż 2.

Czcionki

Zamiast wielkości określonej w pikselach, należy używać wartości „small”, „medium”, „large”, itp. W pliku CSS można importować czcionki z serwisu Google: <https://www.google.com/fonts>, np.:

```
@import url(https://fonts.googleapis.com/css?family=Josefin+Sans:400,100,700,100italic,400italic,700italic&subset=latin,latin-ext);
```

Podpisy pod fotkami

Zamiast tego:

```

```

```
<p>Fotka z konferencji.</p>
```

używaj tego:

```
<figure>
```

```

```

```
<figcaption>Fotka z konferencji.</figcaption>
```

```
</figure>
```

Odwołanie do stylów

Zamiast tego:

```
<link rel="stylesheet" href="glowny.css" type="text/css" />
```

używaj tego (bez „type”):

```
<link rel="stylesheet" href="glowny.css">
```

Należy unikać importowania stylów (@import), gdyż spowalnia to renderowanie całej strony.

Odwołanie do skryptów

Zamiast tego:

```
<script type="text/javascript" src="moj-skrypt.js"></script>
```

używaj tego (bez „type”):

```
<script src="moj-skrypt.js"></script>
```

Formularz z e-mail

Dzięki zastosowaniu atrybutu „type=email” - przeglądarka sprawdza, czy wpisany tekst ma postać e-maila. Zastosowanie atrybutu „placeholder” powoduje wyświetlanie w polu formularza

domyślnego adresu lub tekstu w kolorze szarym, który znika po kliknięciu w pole. Atrybut „required” powoduje zaś, iż pole zostanie otoczone czerwoną ramką, gdy przy wysłaniu pozostanie puste (jako błąd). Atrybut „pattern” określa warunki jakie musi spełnić wpis (wielkość liter, długość ciągu):

```
<input name="email" type="email" placeholder="domyslny@email.com" required="required" "autofocus=autofocus" pattern="^[a-zA-Z0-9\.\_\-]+@[a-zA-Z0-9\.\_\-]+\.[a-z]{2,4}$">
```

Legenda:

^ - zaczyna się na (element początkowy, początek wiersza);

\$ - kończy się na (element końcowy, koniec wiersza);

+ - conajmniej jedno wystąpienie poprzedzającego znaku (znaków);

? - najwyżej jedno wystąpienie poprzedzającego znaku;

. - kropka oznacza dowolny znak;

**** - symbol, który następuje po tym ukośniku, nie jest symbolem wieloznacznym - czyli należy odczytywać go dosłownie.

Audio

Przeglądarka Firefox odtwarza nieopatentowane formaty: **WAV** (PCM 8/16 bps), **OGG** (.ogg, .oga, .ogv, .ogx, .spx, .opus), **WebM**., **MP4** (niektóre, np. kodowane jako AAC).

Odtworzony zostanie plik.ogg, a jeśli przeglądarka nie rozpoznaje tego formatu, to plik.mp3:

```
<audio autoplay="autoplay" controls="controls">
  <source src="plik.ogg">
  <source src="plik.mp3">
  <a href="plik.ogg">Pobierz muzykę.</a>
</audio>
```

Uwaga: Znacznik <audio> nie potrafi odtwarzać playlisty, np. *.m3u.

Video

Przeglądarka Firefox odtwarza nieopatentowane formaty **OGG** (kodek video *Theora*, kodek audio *Vorbis*), **WebM** (kodek video *VP8*, audio *Vorbis*), a także **MP4** (częściowo, niektóre kodery, w tym video *H.264*, oraz audio AAC).

Przeglądarka spróbuje odtworzyć plik „film.ogv”. Jeśli nie będzie znała tego formatu, spróbuje zastosować wersję drugą, potem trzecią. Atrybut „preload” ładuje wstępnie plik do pamięci:

```
<video controls preload="auto">
  <source src="film.ogv">
  <source src="film.mp4">
  <p>Twoja przeglądarka jest przestarzała i nie potrafi odtworzyć tego pliku. <a
  href="film.mp4">Możesz go jednak pobrać na dysk.</a> </p>
</video>
```

Menu

```
<nav>
  <span>Link 1</span><span>Link 2</span>
</nav>
```

lub

```
<nav class="menu">
<ul>
  <li><a href="#">Link</a></li>
  <li><a href="#">Link</a></li>
  <li><a href="#">Link</a></li>
</ul>
</nav>
```

a w części CSS:

```
.menu ul{
text-align: center;
margin-bottom: 10px;
float: left;
width: 100%;
list-style-type: none; }
```

W menu opartym na można tworzyć podmenu (wewnątrz tworzymy). Aby automatycznie rozwijało się i najechaniu myszką i zwiijało, należy w CSS:

```
ul { // Wewnętrzne <ul>
visibility: collapse;}
```

Inne ciekawe właściwości

Block - elementy typu „*block*” (np. *div*) domyślnie wypełniają całą przestrzeń horyzontalnie, tzn. nie tolerują elementów przed sobą ani za sobą (stawiają za sobą znak łamania wiersza). Można zdefiniować ich wymiary (szerokość i wysokość).

Inline - elementy zachowują się podobnie do liter w zdaniu (np. *img*, *span*), mogą występować w jednej linii i tylko w tej sytuacji działa właściwość „*vertical-align*”. Wyrównanie poziome tych elementów (*text-align*) nie działa na każdego z osobna (można wyrównać poziomo tylko całą linię, czyli wszystkie elementy razem).

Inline-block - elementy blokowe, które tkwią obok siebie w jednej linii i zachowują się jak wyrazy w zdaniu.

List-item - elementy blokowe, które zachowują się jak elementy .

Float - powoduje, że element staje się typu „*block*”, jest siłą przesunięty (wyrównany) w lewy lub prawy bok i jest opływany (otaczany) przez pozostałe elementy tej warstwy. Jeśli nie chcemy, aby z jakiejś strony był otaczany innymi elementami, używamy np. atrybutu „*clear: left*”. Element typu „*float*” może swoimi gabarytami wychodzić poza element nadrzędny; jeśli nie chcemy, aby tak było, możemy dać elementowi nadrzédnemu parametr „*overflow: auto*”.

Position - przesuwa blok lub linię umieszczając je automatycznie w nowej warstwie (nawet jeśli nie zadeklarujemy „*z-index*”):

- nadaje im nową pozycję względem miejsca, gdzie normalnie jest ich miejsce, ale w zależności także od otaczających je elementów (**relative** - może być typu „*block*” lub „*inline*”),
- względem całego dokumentu html (**absolute** - tylko typ „*block*”; element odrywany jest od warstwy elementu macierzystego, nie wpływa na położenie innych elementów i inne elementy nie wpływają na jego położenie; jeśli element jest wewnątrz elementu z „*position: relative*”, wtedy będzie pozycjonowany względem tego nadrzédnego elementu, a nie względem strony html)
- lub względem przeglądarki (**fixed** - element utrzymuje swoją pozycję nawet wtedy, gdy dokument html jest przewijany; jest odrywany od warstwy całego dokumentu html).

Z każdą z powyższych wartości „*position*” związana jest także wartość „*z-index*” (i tylko z nią). Im większa wartość „*z-index*” tym warstwa znajduje się wyżej, ale tylko pod warunkiem, że znajduje się w tej samej przestrzeni „*position*” (warstwa posiadająca „*z-index: 1*” i „*position: absolute*” będzie wyżej niż warstwa z „*z-index: 50*” i mająca „*position: relative*” - bo została ona oderwana z macierzystego elementu). Warstwy mogą mieć też wartości ujemne, wtedy przesuwa się pod spód standardowej domyślnej warstwy.

Wartość „**static**” nie zmienia domyślnego zachowania bloku, bo jest domyślna nawet jeśli jej nie zadeklarujemy (element zachowuje się jak zwykły blok, a wartości „*top*”, „*left*”, oraz „*z-index*”, itp. - nie mają w tym przypadku zastosowania).

Co wybrać: „*float*” czy „*position: absolute*”? Jeśli element ma być opływany przez inne elementy, wybierz „*float*”. Tak samo jak chcemy wyrównać do prawej dwa bloki (np. kolumny zrobione z *div*-ów) - użyjemy „*float*” (wzajemnie dopasują swoje położenie obok siebie). Jeśli element ma nie wchodzić w interakcje (związane z położeniem) z innymi elementami, wybierz „*position*”.

<a> - Specyfikacja HTML5 pozwala na otaczanie elementów blokowych linkami, nawet całych

paragrafów i zawartości mieszanej. Ten sposób jest przydatny dla małych ekranów smartfonów. Jednakże Google odradza taką filozofię, gdyż jej maszyny mają w tym przypadku problem z indeksowaniem stron w wyszukiwarce i analizą ich zawartości.

Walidacja

Najkrótsze uniwersalne odnośniki do walidacji to (jedna długa linia):

```
<p onclick="document.location='http://validator.w3.org/check?uri='+document.location.href">  
[ Walidacja HTML ]  
</p>
```

```
<p onclick="document.location.href='https://jigsaw.w3.org/css-validator/validator?uri=https://'+document.location.host">  
[ Walidacja CSS ]  
</p>
```

Info: Kod działa zarówno w przypadku załadowania strony za pomocą protokołu *http*, jak i za pomocą protokołu *https*.

Te same odnośniki w wersji rozbudowanej (symulacja przycisku):

```
<footer style="margin-top: 80px; text-align: center;">  
<p>  
<span style="background-color: lightgray; padding-left: 10px; padding-right: 10px; padding-top: 2px; padding-bottom: 2px; cursor: pointer; border-style: solid; border-width: 3px; box-shadow: 3px 3px gray; font-weight: bold; margin-right: 10px;"  
onclick="document.location='http://validator.w3.org/check?uri='+document.location.href"  
onmouseover="this.style.backgroundColor='lightpink'"  
onmouseout="this.style.backgroundColor='lightgray'"> Walidacja HTML </span>  
  
<span style="background-color: lightgray; padding-left: 10px; padding-right: 10px; padding-top: 2px; padding-bottom: 2px; cursor: pointer; border-style: solid; border-width: 3px; box-shadow: 3px 3px gray; font-weight: bold;"  
onclick="document.location.href='https://jigsaw.w3.org/css-validator/validator?uri=https://'+document.location.host"  
onmouseover="this.style.backgroundColor='lightpink'"  
onmouseout="this.style.backgroundColor='lightgray'"> Walidacja CSS </span>  
</p>  
</footer>
```

Ostatnia aktualizacja: 3 marca 2024.