# HTML - czym jest HyperText Markup Language?

[Edytuj mnie](https://skydrive.live.com/redir?resid=94901F7FBC72D5B6!118&authkey=!AJ5J6GCzegJa3dA)

Język znaczników hipertekstowych to standard rozwijany przez organizację World Wide Web Consortium (W3C). Organizacja ta wprowadza standardy formatów danych związanych z Internetem.



HTML służy do opisu struktury dokumentu, organizowania jej w semantyczne bloki i umożliwia prezentowanie informacji w taki sposób, aby zewnętrzne programy mogły tę informację przetworzyć. Mogą to być algorytmy analizujące dane (np. Google-bot, robot skanujący Internet), platformy wymiany informacji (np. Facebook), czy w końcu przeglądarki umożliwiające ludziom korzystatnie z zasobów Internetu.

Plik HTML jest plikiem tekstowym, to znaczy, człowiek nawet bez specjalnych narzędzi jest w stanie napisać (bez edytorów) i przeczytać (nie korzystając z przeglądarki) stronę internetową. Taki plik jest uniwersalny, czyli wszystkie programy są go w stanie przeczytać, o ile zachowane są ścisłe reguły, a treść pliku jest zgodna ze standardem. Dokument w HTML składa się ze znaczników i atrybutów, które mają opisane w specyfikacji znaczenie, określające charakter tych elementów.

Dokumentem standaryzującym język jest specyfikacja HTML publikowana przez W3C, aktualna wersja specyfikacji to HTML5.

HTML jest językiem o paradygmacie deklaratywnym, to znaczy że opisuje jaki efekt ma zostać uzyskany, jakie mają być jego własności. HTML nie jest językiem programowania, w odróżnieniu od języków typu C czy PHP, nie posiada struktur kontroli wykonania kodu, to znaczy wyrażeń warunkowych czy pętli.

Specyfikacja HTML5 zawiera opis języka HTML, a także CSS i JavaScript. CSS omówiony zostanie w dalszej części tego podręcznika.

JavaScript to język programowania skryptowego, interpretowanego przez przeglądarkę internetową. Składnia JavaScript jest mocno wzorowana na języku C, wzbogacona została o własności obiektowe (z rozszerzaniem obiektów przez prototypy zamiast tradycyjnego dziedziczenia), jest standaryzowana w oparciu o specyfikację ECMAScript. Rozwój języka JavaScript jest tak silny, że język jest już stosowany poza przeglądarką, jako język skryptowy do programowania logiki programów czy jako serwerowy język skryptowy. Dzięki licznym optymalizacjom, jego wydajność przekracza powszechnie stosowane języki takie jak PHP czy Ruby.

# Rys historyczny

Początki rozwoju języka HTML sięgają lat osiemdziesiątych XX wieku. Człowiekiem którego uważa się za ojca HTML-a jest Tim Burners-Lee. W jego zamyśle HTML miał być ułatwieniem w organizowaniu treści w Internecie który wówczas był siecią akademicką. Chciał on rozwiązać problem braku powiązań pomiędzy dokumentami. Udało się to osiągnąć poprzez wykorzystanie protokołu HTTP (sposobu wymiany danych pomiędzy serwerem i przeglądarką) i wprowadzenie języka zawierającego takie powiązania.

HTML jest rozszerzeniem języka XML, a ten z kolei wywodzi się ze standardu SGML.

Pierwszą zestandaryzowaną wersją HTML-a jest wersja 2.0 i ukazała się jako dokument RFC (request for comments) w roku 1996. Kolejna wersja standardu ukazała się 14 stycznia 1997 i nosiła numer 3.2. Później W3C opublikowało standard HTML4 18 grudnia 1997.

HTML4 składał się z trzech części:

* Transitional - zezwalał na używanie elementow prezentacyjnych
* Strict - ścisłe trzymanie się specyfikacji
* Frameset - rozszerzał Transitional o ramki.

W tamtych czasach rodziła się koncepcja oddzielenia formatowania treści od struktury samej treści, czego wyrazem był CSS. Niestety ówczesne przeglądarki nie radziły sobie z kodem CSS.

Pod koniec roku 2000 powstał standard XHTML1.0 i zawężał HTML4 do zgodności z językiem XML. Wymuszał domykanie wszystkich tagów, nakazywał ścisłe trzymanie się specyfikacji.

W3C porzuciło rozwój języka XHTML więc jego wersja nigdy nie osiągnęła numeru 2.0. Zamiast tego został opracowany standard HTML5, w którym wszystkie tagi mają znaczenie semantyczne. Porzucenie prac nad XHTML2 było spowodowane oporem twórców przeglądarek którzy zbyt wolno implementowali kolejne zmiany i kadra odpowiedzialna za rozwój nowego standardu została przeniesiona do pracy przy specyfikacji HTML5.

**Linki:**

Infografika przedstawiająca rozwój HTML-a:

<http://visual.ly/history-html5-line-popular-culture-events>

# Przeglądarki

Stanowią podstawowe narzędzie w korzystaniu z Internetu. Interpretują one kod HTML tak, by użytkownik zobaczył treść podaną w przystępny sposób.

Pierwszą przeglądarką był program stworzony przez Tima Burnersa-Lee, nosiła nazwę WorldWideWeb. Wkrótce pojawiło się więcej przeglądarek, ale żadna z nich nie uzyskiwała dominacji.

Pierwszą przeglądarką która uzyskała dominację była Mosaic (1993 r.), a rok później oparty o Mosaic, Netscape Navigator. W odpowiedzi na rosnącą popularność przeglądarki Netscape'a była przeglądarka Microsoftu, Internet Explorer.

Za początek gwałtownego rozwoju przeglądarek można uznać rok 1997, kiedy to Microsoft wydał przeglądarkę Internet Explorer 5, zintegrowaną z systemem operacyjnym Windows. IE5 był przedmiotem sporu sądowego Microsoftu z USA, na tle antymonopolowym.

Netscape nie był w stanie odpowiedzieć na działania konkurenta, kolejne wydanie z numerem 6 było powolne i niestabilne. Skutkiem tego przeglądarka Microsoftu zdobyła dominację rynkową, rozwijała się powolnie, wprowadziła wiele elementów rozszerzających specyfikację (m.in. ActiveX który jest przyczyną wielu błędów związanych z bezpieczeństwem).

Różnice pomiędzy przeglądarkami doprowadziły do tego, że niektórzy pisali dwie wersje strony tak, aby użytkownik w dwóch najważniejszych przeglądarkach widział podobny efekt.

Netscape, nie będąc w stanie konkurować z Microsoftem, zdecydował się udostępnić społeczności kod swojej przeglądarki. Na tej bazie powstała przeglądarka Mozilla Suite, która następnie została przekształcona w Mozilla Firefox. Jej rozwojem do dziś zajmuje się fundacja Mozilli.

Kolejne wydania przeglądarki Microsoftu nie wprowadzały istotnych zmian wobec starszych wersji, chociaż praca ze starszymi przeglądarkami Microsoftu dla wielu webmasterów jest udręgą. Zmiana nastąpiła dopiero z wydaniem IE10 który w dużym stopniu poprawnie interpretuje standard HTML5, ale nadal rozszerza go o swoje własności.

W odpowiedzi na nieudolność Microsoftu do poprawienia swojej przeglądarki powstało wiele stron internetowych skupiających właścicieli stron walczących z IE. W polskim Internecie kampani na rzecz porzucenia IE prowadzi portal browsehappy.pl.

W cieniu największych przeglądarek pozostały mniejsze projekty takie jak Opera Browser rozwijana przez norweską firmę Opera Software. Istnieją też przeglądarki tekstowe takie jak Lynx czy Links z których można korzystać za pomocą konsoli systemowej.

Od roku 1999 był rozwijany silnik KHTML wykorzystany w przeglądarce Konqueror dla środowiska Unix. KHTML jest chwalony za wysoką zgodność ze standardami W3C i niskie zużycie pamięci. Na jego podstawie powstała przeglądarka Safari, wykorzystuje ona pożyczony kod, a silnik został nazwany Webkit. Dzisiaj Webkit jest silnikiem napędzającym przeglądarki Safari i Google Chrome, a także większość przeglądarek na urządzeniach mobilnych (Android i iOS).

## Popularność przeglądarek w Polsce

na podstawie ranking.pl

|  |  |
| --- | --- |
| Firefox 21 | 36% |
| Chrome 27 | 27% |
| IE 8 | 6% |
| Opera 12 | 6% |
| IE 10 | 4% |
| IE 9 | 4% |

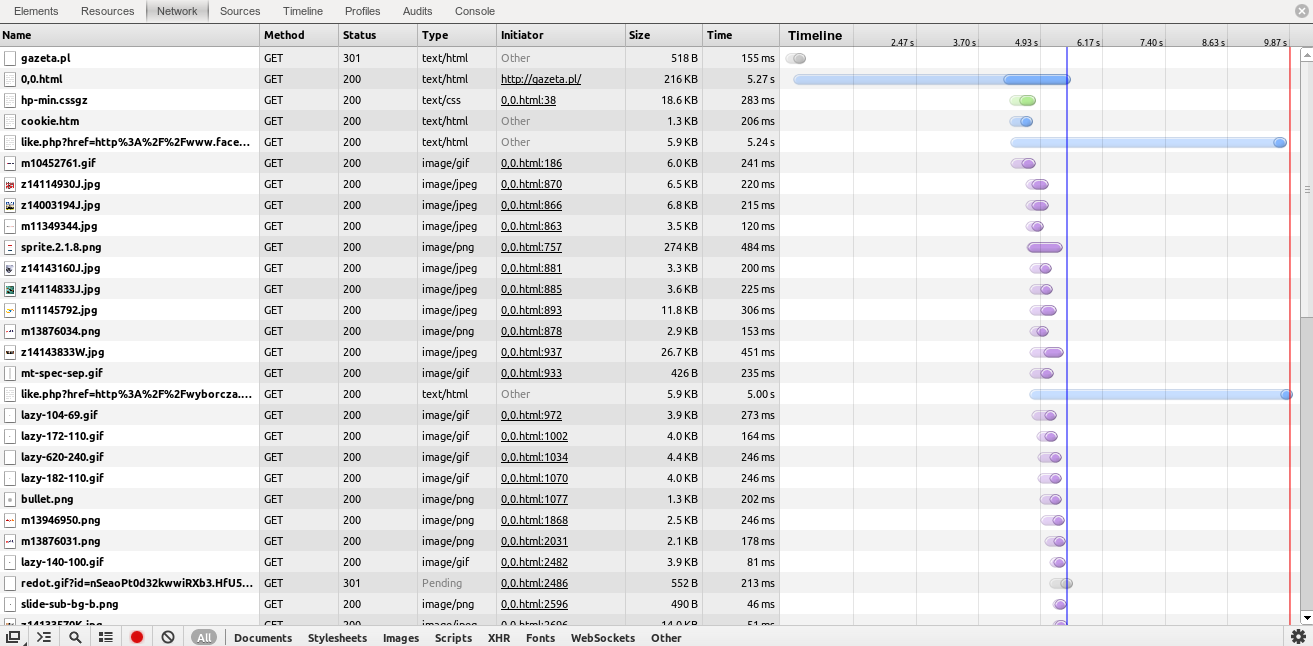
Warto zauważyć, że Firefox i Chrome są aktualizowane na bieżąco (starsze wersje stanowią mniej niż 1%), zaś IE8 nie może zostać zaktualizowany na komputerach z Windowsem XP do wyższej wersji. Dopiero zmiana systemu operacyjnego na Windows 7 pozwala na aktualizację do IE10.

IE8 traci w kolejnych kwartałach około pół punktu procentowego, zaś popularność IE10 wzrasta o około 0.4 punktu procentowego w tym czasie.

# Sposób działania stron internetowych

Przeglądarki internetowe interpretują kod HTML wysłany na żądanie przez serwer. To znaczy, że przeglądarka wysyła do serwera zapytanie o zasób znajdujący się pod konkretnym adresem URL (Uniform Resource Locator, unikalny adres zasobu). W odpowiedzi dostaje zazwyczaj kod odpowiedzi HTTP i treść dokumentu HTML.

Uzyskana treść jest analizowana, na podstawie kodu HTML tworzone jest drzewo dokumentu. Przeglądarka wysyła do serwera zapytania o kolejne pliki które są dołączone do strony, by w końcu wyświetlić stronę użytkownikowi.

Rys. Diagram typu waterfall przedstawiający kolejne zapytania do serwera przez przeglądarkę Google Chrome. Na przykładzie witryny [http://gazeta.pl](http://gazeta.pl/)

Serwer, gdy znajdzie wskazaną treść, udziela odpowiedzi z kodem HTTP 200, co oznacza że dokument został znaleziony i zostanie przesłana jego treść. Wśród ważniejszych kodów odpowiedzi można wymienić:

* 301 - przekierowanie (oznacza, że dokument zmienił adres),
* 302 - przekierowanie tymczasowe,
* 401 - odmowa dostępu z powodu braku hasła,
* 403 - odmowa dostępu (np. brak uprawnień serwera do odczytania pliku)
* 404 - dokumentu nie znaleziono,
* 500 - błąd wewnętrzny serwera

Pełny spis kodów odpowiedzi wraz z ich opisem można znaleźć w specyfikacji protokołu HTTP i w Wikipedii (<http://pl.wikipedia.org/wiki/Kod_odpowiedzi_HTTP>).

Tworząc serwis www należy mieć świadomość, że każde kliknięcie użytkownika w link powoduje wysłanie nowego zapytania do serwera i w efekcie pobranie nowego dokumentu HTML. Przełamanie tego modelu przyniosła technologia AJAX która pozwala zapytać się serwer o dane, na ich podstawie wygenerować nową treść i podmienić ją na stronie bez przeładowania. Szerokie wykorzystanie JavaScript i AJAX spowodowało powstanie aplikacji internetowych działających w przeglądarce, pracujących bez przeładowania strony (usługa Gmail jest tego przykładem). Powoduje to powstanie nowych wyzwań stojących przed webmasterami.

## Serwer i domena

Strona internetowa, żeby była dostępna w Internecie musi zostać umieszczona na serwerze stron internetowych czyli na hostingu. Serwer hostujący stronę jest dostępny jako adres IP, co jest wygodne dla komputerów ale trudne dla ludzi, dlatego oprócz wykupienia hostingu musimy jeszcze mieć domenę. Wielu hostingodawców umożliwia skorzystanie z bezpłatnej poddomeny, ale profesjonalna strona powinna mieć wykupioną osobną domenę wskazującą na nasz serwer.

# Narzędzia i edytory

Pliki HTML to zwykłe pliki tekstowe, to znaczy systemowy notatnik jest w stanie je poprawnie wyświetlić. Występują jednak pewne niedogodności wynikające z różnic w sposobie zapisu znaków końca linii czy polskich znaków diakrytyczych, które dobry edytor powinien naprawiać. Notatnik nie ma takiej funkcjonalności, a zatem jest złym narzędziem do tworzenia stron www.

Pojedynczy plik może nieraz urosnąć do dużych rozmiarów, przez co pomocne staje się korzystanie z kolorowania składni i podpowiedzi edytora do tworzonego właśnie kodu.

## Edytory

Wśród ważniejszych komercyjnych narzędzi do tworzenia kodu HTML warto wymienić:

* Microsoft Visual Studio - standard biznesowy Microsoftu, zintegrowane środowisko programistyczne o bardzo szerokich możliwościach
* Adobe Dreamweaver – standard biznesowy firmy Adobe, narzędzie często wykorzystywane przez grafików, rozpowszechniane jako oddzielny produkt i jako część pakietu Creative Suite
* PhpStorm - nowatorskie zintegrowane środowisko programistyczne firmy JetBrains
* SublimeText - prosty i elegancki edytor kodu (nie tylko HTML)

## Edytory darmowe i Open Source

W odróżnieniu od komercyjnych rozwiązań, te edytory każdy może ściągnąć taki program z Internetu i używać bez ponoszenia opłat licencyjnych. Ciekawe jest to, że te programy często są bardziej na bieżąco z rozwojem standardów niż narzędzia komercyjne.

* Eclipse - standardowe narzędzie programistów Javy, posiadające również wsparcie dla HTMLa
* Netbeans - konkurencyjne środowisko programistyczne wobec Eclipse
* Notepad++ - prosty edytor kodu nie tylko HTML, oferuje m.in. kolorowanie składni. Stanowi dobre narzędzie pracy w przypadku mniejszych projektów
* Adobe Brackers - pierwszy poważny projekt Open Source firmy Adobe, lekkie elastyczne i zwinne środowisko dla webmasterów. Projekt jest aktywnie rozwijany, co oznacza że zdarza się, że Brackets się zawiesi. Mimo to Brackets jest chętnie przyjmowany przez społeczność i aktywnie rozwijany. Nowe wersje wychodzą co około 3-4 tygodnie. Ciekawe jest to, że Brackets jest stworzony w HTML5 i JavaScript.
* Środowiska graficzne Gnome i KDE na komputerach typu Unix i GNU/Linux posiadają wbudowane edytory kodu wystarczające w większości przypadków do pracy.

## Klient FTP

Jest to narzędzie dzięki któremu możemy umieścić pliki naszej strony www na serwerze. Nieraz posiadają rozbudowane opcje takie jak szyfrowanie haseł zapisanych w programie, obsługę szyfrowania transmisji danych, wznawianie przerwanej transmisji, synchronizowanie plików lokalnych i zdalnych. Obsługa FTP jest zwykle zintegrowana w środowiskach programistycznych.

Wśród ważniejszych klientów FTP warto wymienić:

* Total Commander - komercyjny klient FTP, zyskał popularność wśród użytkowników starszej daty
* FileZilla - klient na licencji opensource, pozwala szyfrować hasła i transmisję danych, bardzo wygodny do przesyłania dużej liczby plików do/z serwera dzięki wsparciu wznawiania transmisji
* WinSCP - klient na licencji opensource, pozwala szyfrować hasła i transmisję danych, jest wygodny w połączeniu z lekkimi edytorami takimi jak Notepad++ czy Brackets
* Midnight Commander - menedżer plików wzorowany na programie Norton Commander, standard w środowisku konsolowym na serwerach Unix i Linux.
* Środowiska Gnome i KDE posiadają wsparcie dla FTP zintegrowane z menedżerem plików, co oznacza że można na nich korzystać z FTP tak jak z plików przechowywanych lokalnie.

# Format danych

# HTML

Dokumenty HTML to zwykłe pliki tekstowe z rozszerzeniem .htm lub .html który jest częściej spotykany. Na komputerach z Windows może być konieczne włączenie widoczności rozszerzeń plików.

Dokument HTML może być zapisany z różnym sposobem kodowania znaków diakrytycznych (spoza standardu ASCII). Spotykanymi kodowaniami są Windows-1250 I ISO-8859-2. Ten drugi przez lata był standardem zapisu plików z kodowaniem środkowoeuropejskim jednak obecnie następuje zwrot w kierunku uniwersalnego kodowania UTF-8 które pozwala zapisać wszystkie znaki bez względu na rejon.

## Grafika

JPG - pozwala na zapisanie zdjęcia z kompresją stratną dzięki czemu duże zdjęcia ważą mniej.

GIF - zapisuje zdjęcia z utratą jakości poprzez ograniczenie palety barw. Posiada możliwość zapisu przezroczystości binarnej (czyli albo jest grafika albo widać tło, bez efektów przejścia) I zapis animacji kolejnych klatek.

PNG - pozwala na zapis plików w podobny sposób jak GIF czyli z ograniczoną paletą albo jako zdjęcia zapisane bezstratnie z pełną obsługą przezroczystości, dzięki czemu jest formatem uniwersalnym. Jego wadą jest duży rozmiar przesyłanych zdjęć.

SVG - pozwala na zapis obrazu w postaci wektorowej, dzięki czemu obrazy mogą być dowolnie skalowane, bez utraty jakości

## Audio i wideo

Chociaż w HTML można dołączać pliki audio I wideo, wiąże się z tym pewna niedogodność. Różne przeglądarki potrafią obsłużyć takie pliki zakodowane w różny sposób.

Pliki wideo należy zapisać w dwóch wersjach, w formatach:

* WebM/VP8 by przeglądarki Firefox i Chrome potrafiły je wyświetlić
* MP4/h264 by przeglądarki Internet Explorer (9+) I Safari potrafiły je wyświetlić

Większość przeglądarek potrafi prawidłowo wyświetlić filmy zapisane w formacie flv używanym przez wtyczkę Flash.

Pliki audio również należy zapisać na dwa sposoby, analogicznie do wideo:

* Ogg Vorbis dla Firefoksa i Chrome
* MP3 dla IE oraz Safari

Warto zauważyć, że kodeki wspierane w Firefoksie i Chrome są darmowe dostępne jako Open Source zaś te w IE oraz Safari są dostępne komercyjnie.

## Flash

Technologia firmy Adobe jest schodzacym z rynku standardem zapisu obrazów z interaktywnymi animacjami i wideo. Technologia ta jest źle (Android) lub w ogóle nie wspierana (Apple) na urządzeniach mobilnych dlatego zaleca się porzucanie jej.

Mimo to wykorzystanie flasha jest konieczne by dostarczyć treści wideo i audio w starszych wersjach IE.

# Składnia

Dokument HTML jest zwykłym plikiem tekstowym. Przeglądarka przetwarzając go ignoruje wszystkie białe znaki i formatowanie. Wynika z tego, że tekst przekopiowany z Worda i umieszczony w dokumencie HTML wypełni okno przeglądarki od lewej krawędzi do prawej, łamiąc go po słowach, które nie zmieszczą się już w danym wierszu.

Przeglądarka interpretując dokument HTML respektuje jedynie pojedyncze spacje. Wszelkiego rodzaju wytłuszczenia, podkreślenia i pochylenia czcionki, a także wcięcia i wyrównywanie bloków tekstu nie są przez przeglądarkę interpretowane. Dlatego, aby tekst widoczny na stronie był odpowiednio, wg naszych oczekiwań sformatowany, należy użyć specjalnych oznaczeń ten tekst formatujących. Te oznaczenia w języku HTML nazywa się znacznikami lub tagami. Znacznik to wyrażenie ujęte w nawiasy trójkątne (znak mniejszości i większości) w postaci:

<znacznik>

## Tagi

Tagi tworzą sekcje nadające tekstowi i tagom które zawierają nowe znaczenie. Wszystkie tagi powinny być zamykane, to znaczy każdy tag powinien posiadać swoją domykającą parę, np.:

<h1>Moja strona www</h1>

Wyjątkiem od tej zasady są tagi punktowe, które nie obejmują sekcji. Wtedy powinny posiadać na końcu znacznik domknięcia:

<br />

Od zasady domykania tagów w HTML5 jest pewne odstępstwo - niektóre tagi mogą być niezamknięte, jeśli z kontekstu jasno wynika gdzie one się powinny kończyć. Jednak ze względu na zgodność ze starszym standardem XHTML, warto zawsze domykać tagi.

Nazwy tagów zapisujemy małymi literami, chociaż w czasach standardu HTML4 robiono to wielkimi literami.

NIE WOLNO TAGÓW ZAMYKAĆ NA KRZYŻ  
To znaczy że zanim zamkniemy zewnętrzny tag, najpierw zamkniemy tag wewnętrzny, na przykład:

*Prawidłowy kod:*

*<p>*  
*<a href="http://google.com">Idź do google</a>*  
*</p>*

## Atrybuty

Wszyskie tagi mają swoje znaczenie - rolę do wypełnienia. Jednak jest ich zbyt mało, by odwzorować za ich pomocą kreatywny i interaktywny projekt. Dlatego tagi mogą posiadać atrybuty które dodają tagom nowe własności. Składnia atrybutu jest następująca:

<h1 **id="main"**>tekst</h1>

Atrybut od nazwy tagu musi być oddzielony spacją, dwa kolejne atrybuty również. Kolejność atrybutów jest dowolna. Po nazwie atrybutu następuje znak równości = a następnie wartość która ma zostać przypisana do atrybutu, objęta w cudzysłowy lub apostrofy (częściej spotykane sa cudzysłowy).

Nazwy atrybutów pisane są małymi literami.

## Encje

Encja to specjalny zapis symbolu który w innym przypadku wpłynąłby na resztę kodu. Ponadto encje pozwalają zapsiać znaki niobecne w danym kodowaniu.

Wszystkie encje składają się ze znaku & rozpoczynającego encję i znaku średnika kończącego.

Wśród ważniejszych encji warto wymienić:

* &lt; czemu odpowiada znak <
* &gt; czyli >
* &amp; czyli po prostu &
* $copy; znaczek copyright
* &nbsp; spacja niełamliwa

## Kolory

W HTML kolory zapisywane są przez ich angielskie nazwy, kody szesnastkowe lub kody RGB.

Przykładowe nazwy: red, green, blue, teal, silver, gold, orange

### Zapis szesnastkowy

Ma postać #RRGGBB lub #RGB gdzie symbol R oznacza ilość składowej barwy czerwonej, G zielonej i B niebieskiej (podstawowe barwy w monitorze). Zapis krótszy z trzema cyframi jest równoznaczy zapisowi w którym dana cyfra jest powtórzona dwukrotnie. Cyfry w zapisie szesnastkowym zawierają się od 0 do f (gdzie f to 16, a to 10 i tak dalej) co daje 256 stopni natężenia (lub 16 dla krótszego zapisu).

**Przykładowo**

#ff0000 oznacza kolor czerwony

#fff oznacza kolor biały

### Kody dziesiętne

Mają postać zapisu rgb(x,y,z) gdzie litery x, y, z oznaczają natężenie składowych czerwonej, zielonej i niebieskiej podanej jako liczba z zakresu od 0 do 255.

**Przykładowo:**

rgb(255,0,0) oznacza kolor czerwony

rgb(255,255,255) oznacza kolor biały

Występuje też zapis dziesiętny z przezroczystością: rgba( 0,0,0, **0.5** ) gdzie ostatnia wartość oznacza stopień nieprzezroczystości i jest liczbą z zakresu od 0 (całkowicie przezroczysty) do 1 (całkowicie nieprzezroczysty).

**Do wygenerowania kodów szesnastkowych lub dziesiętnych można wykorzystać narzędzia typu Brackets czy Photoshop. Nie trzeba się ich uczyć na pamięć!**

## Semantyka elementów

Jak zostało omówione wcześniej, elementy w HTML-u mają znaczenie semantyczne. To znaczy, że tagom przypisane jest ich znaczenie. I tak przykładowo:

h1 - nagłówek pierwszego rzędu,   
h2 - nagłówek drugiego rzędu  
a - odnośnik  
em - wyróżnienie (pochylenie)  
strong - wzmocnienie znaczenia (pogrubienie)  
p - akapit tekstu  
aside - notka na boku  
footer - stopka  
article - treść artykułu

Tagi powinny tworzyć logiczną i spójną strukturę treści tak jak przykładowo treść pracy magisterskiej zachowuje spójną strukturę.

# Formatowanie tekstu

HTML udostępnia szereg tagów służących do określenia struktury logicznej treści i nadania jej początkowego formatowania.

## Blokowe elementy formatowania tekstu

Te tagi charakteryzują się tym, że zajmują całą dostępną szerokość, a po ich zakończeniu, kolejny element zaczyna się od nowego wiersza.

Wśród elementów blokowych warto wymienić:

### Akapit tekstu:

<p> tekst tekst tekst</p>

### Nagłówek pierwszego rzędu

<h1>Nagłówek </h1>

#### Nagłówek drugiego rzędu

<h2>Nagłówek </h2>

i tak dalej

<h3>Nagłówek </h3>

<h4>Nagłówek </h4>

<h5>Nagłówek </h5>

<h6>Nagłówek </h6>

### Sekcja

(blok logiczny)

<section>…</section>

### Warstwa

(blok konstrukcyjny)

<div>...</div>

### Treść artykułu

<article>…</article>

### Nawigacja

<nav>…</nav>

### Tekst preformatowany

to znaczy wyświetlający się w takim układzie, jak w pliku html, z zachowaniem białych znaków

<pre>

tekst tekst tekst

</pre>

### Pozioma linia

<hr />

### Nagłówek i stopka

<header>…</header>

<footer>…</footer>

### Lista nieuporządkowana

nazwa *ul* pochodzi od słów *unordered list*, zaś *li* to *list item*

<ul>  
<li>pierwszy element</li>  
<li>drugi element</li>  
<li>trzeci element</li>  
</ul>

### Lista numerowana

nazwa *ol* pochodzi od słów *ordered list*

<ol>  
<li>pierwszy element</li>  
<li>drugi element</li>  
<li>trzeci element</li>  
</ol>

Do zadań list, jako spisów elementów, należy też grupowanie linków stanowiących menu witryny. Tagi list można zagnieżdżać w sobie, tworząc w ten sposób listy wielopoziomowe.

### Tabela

*td* symbolizuje komórkę tabeli, zaś obejmujący komórki tag *tr* to wiersz w tabeli

<table>  
<tr>  
 <td>1</td> <td>2</td> <td>3</td>  
</tr>  
<tr>  
 <td>2</td> <td>4</td> <td>6</td>  
</tr>  
<tr>  
 <td>3</td> <td>6</td> <td>9</td>  
</tr>  
</table>

Tabele niegdyś pełniły funkcję konstrukcyjną, to znaczy służyły do budowania layoutu strony www. Prowadziło to do wielokrotnego zagnieżdżania tabelek, tworzenia szablonów stron których utrzymanie było trudne, a takie szablony się rozsypywały przy niewłaściwych zmianach.

Obecnie, w duchu kodu semantycznego, zadaniem tabel jest jedynie prezentowanie danych tabelarycznych, zestawień etc. i nie używa się ich do budowania stron. Tę funkcję przejęły znaczniki <div> i <section>.

Komórki tabel mogą być rozepchnięte na sąsiadujący obszar za pomocą atrybutów colspan i rowspan:

<td colspan="3"> oznacza komórkę która łączy 3 kolejne kolumny, tym samym komórki które są po niej a w tym samym wierszy są wypchnięte w prawo

<td rowspan="2"> oznacza komórkę która łączy dwa kolejne wiersze, tym samym komórki z zajętych wierszy są wypchnięte w prawo (a nie w dół)

Opcjonalnym atrybutem dla tagu <table> jest atrybut border nie przyjmujący wartości:

<table border>

który powoduje wyświetlenie się krawędzi tabel (domyślnie wartości rysowane są bez krawędzi).

Jeśli chcemy zwiększyć grubość krawędzi, można temu atrybutowi przypisać wartość, na przykład:

<table border="5">

#### Semantyczna tabela

<table border>  
 <thead>  
 <tr>  
 <th>Month</th>  
 <th>Savings</th>  
 </tr>  
 </thead>  
 <tfoot>  
 <tr>  
 <td>Sum</td>  
 <td>$180</td>  
 </tr>  
 </tfoot>  
 <tbody>  
 <tr>  
 <td>January</td>  
 <td>$100</td>  
 </tr>  
 <tr>  
 <td>February</td>  
 <td>$80</td>  
 </tr>  
 </tbody>  
</table>

Tag *thead*, *tbody*, *tfoot* określają odpowiednio wiersz nagłówkowy, zawartość tabeli i wiersz - stopkę; *th* oznacza komórki nagłówkowe,

Nie są to oczywiście wszystkie tagi dozwolone w HTML5, pełną listę można znaleźć w specyfikacji. Tylko niektóre z przestawionych wyżej wpływają na formatowanie - nagłówki są pogrubione, a akapity mają między sobą odstęp, zaś listy są wcięte.

Za zmiany dotyczące wyglądu strony www w HTML5 odpowiada CSS, opisany w dalszej części tego podręcznika.

## Tagi liniowe

Te tagi, w odróżnieniu od omówionych wcześniej, obejmują swoim zasięgiem wskazany punkt w tekście lub zbiór znaków. Oznacza to, że nie powodują przełamania linii, można umieścić dwa takie tagi obok siebie.

### Podstawowe formatowanie

Pogrubienie <b></b> oraz <strong></strong>, z tym że strong oznacza dodatkowo wzmocnienie (znaczenie semantyczne), a b - nie.

Pochylenie <i></i> oraz <em></em>, z tym że, jak wyżej, em jest znacznikiem semantycznym i oznacza wyróżnienie. Zazwyczaj tekstu pochyłego używa się do wyróżnienia znaczeń technicznych.

Warto też zauważyć, że tekst pochylony na komputerze jest mniej czytelny i dlatego ogranicza się jego występowanie.

Fragment kodu - <code></code>, często występuje w parze z <pre> dzięki czemu łamanie wierszy i wcięcia są poprawnie odwzorowane.

### Odnośniki

Stanowią jeden z najważniejszych elementów języka HTML. Obejmują swym zasięgiem tekst, który po kliknięciu przeniesie użytkownika do innej podstrony.

Najważniejszym atrybutem tagu <a> jest atrybut *href*, od słów *hypertext reference* i to właśnie on mówi o tym, jaki URL ma zostać otwarty.

Można spotkać atrybut target ustawiony na wartość \_blank, na przykład:

<a href="http://google.com" target="\_blank">Idź do google'a</a>

co powoduje otworzenie linku w nowej karcie przeglądarki. Ten atrybut jest jednak uznany za przestarzały w HTML5 i odradza się jego stosowanie. W założeniu ktoś, kto chce otworzyć stronę w nowej karcie, kliknie ją przy pomocy kółka myszy lub prawym guzikiem i wybierze samodzielnie właściwą opcję. Takie zachowanie jest również bardziej wskazane w stronach www otwieranych na urządzeniach mobilnych.

#### Linki wewnętrzne:

Przykładowy link ma postać:   
<a href="Adres">Opis odnośnika</a>

gdzie Adres oznacza nazwę pliku który ma zostać otwarty, na przykład:

<a href="kontakt.html">Jak się z nami skontaktować?</a>

Plik może się znajdować w innym katalogu, na przykład:

<a href="galeria/index.html">Idź do galerii</a>

Lub w katalogu wyżej:

<a href="../index.html">Wróć do początku</a>

Można też podawać adresy plików w sposób bezwzględny, to jest od adresu domeny:

<a href="/index.html">Opis odnośnika</a>

<a href="/">Opis odnośnika</a>

Co przeniesie nas do pliku inedx.html w katalogu najwyższego rzędu.

#### Linki zewnętrzne

Linki mogą kierować również do zasobów znajdujących się na obcym serwerze, przykładowo:

<a href="http://google.com">Idź do google</a>

Charakterystyczne tutaj jest podanie pełnego adresu, razem z protokołem w ramach atrybutu href.

#### Specjalne linki

Funkcjonują też odnośniki mailowe czy telefoniczne, których zadaniem jest wywołanie programu pocztowego lub ekranu wywołania połączenia telefonicznego (na komórce lub gdy korzystamy ze Skype'a):

<a href="[mailto:adres](https://mail.google.com/mail/?view=cm&fs=1&tf=1&to=adres)@example.com">Napisz do mnie</a>

<a href="tel:22123123">Zadzwoń do mnie</a>

#### Link - kotwica

Ma postać:

<a href="#"> napis</a>

I wtedy kliknięcie linku nie powoduje przeładowania strony.Ewentualnie po znaku # można podać nazwę kotwicy - nazwanego elementu do którego użytkownik zostanie przewinięty, wtedy potrzebujemy też zadeklarować w dokumencie taką kotwicę:<a name="kotwica"></a>

### Span

Fragment tekstu. Tag nie nadaje żadnego znaczenia, po prostu oznacza fragment tekstu.

Lorem <span>ipsum dolor</span> sit amet

### Br

<br /> Znak łamania wiersza

### Media

#### obrazki

Umieszczenie grafiki jako obiektu wiąże się z zastosowaniem znacznika img i należących do niego atrybutów. Najprostsze zastosowanie tego znacznika to:

<img src=”../images/foto1.jpg” alt="Opis alternatywny">

Atrybut src to inaczej źródło (ang. source), z którego pobierana jest dana grafika do dokumentu html. W miejscu kodu HTML gdzie zostanie umieszczony powyższy znacznik, pojawi się zdjęcie w oryginalnych wymiarach. Zasady podawania adresu URL w atrybucie src są identyczne jak w przypadku linków.

Atrybut alt, w XHTML I HTML5 obowiązkowy, opisuje tekst alternatywny czyli opis zdjęcia. Ten opis zostanie wyświetlony komuś kto korzysta z przeglądarki tekstowej, z syntezatora mowy (np osoba niewidząca) a także robotowi wyszukiwarki.

Jeśli zachodzi potrzeba, można ograniczyć wielkość prezentowanego obrazka poprzez dodanie atrybutów width I height:

<img src=”image/foto1.jpg” width=”400” height=”280”>

Obrazek wtedy zostanie przeskalowany przez przeglądarkę do pożądanych rozmiarów. Pominięcie jednego z tych wymiarów oznacza przeskalowanie proporcjonalne.

#### audio

Aby dołączyć plik dźwiękowy do strony www możemy wykorzystać tag <audio> wprowadzony przez HTML5. Jego składnia jest następująca:

<audio controls>  
 <source src="song.ogg" type="audio/ogg">  
 <source src="song.mp3" type="audio/mpeg">  
Twoja przeglądarka nie wspiera audio w html5  
</audio>

Atrybut controls powoduje wyświetlenie się odtwarzacza, tagi source opisują źródło multimedialne do odtworzenia. Należy podać dwa źródła, to znaczy tę samą ścieżkę audio zapisaną w różnych formatach ze względu na niewspieranie niektórych kodeków w niektórych przeglądarkach.

#### wideo

Standard HTML5 wprowadził możliwość umieszczenia materiału wideo bezpośrednio jako tagu, bez półśrodków takich jak wykorzystanie pluginów flash. Składnia jest następująca:

<video poster="movie.jpg" controls>   
<source src="movie.webm" type='video/webm; codecs="vp8.0, vorbis"'/>   
<source src="movie.ogg" type='video/ogg; codecs="theora, vorbis"'/>   
<source src="movie.mp4" type='video/mp4; codecs="avc1.4D401E, mp4a.40.2"'/>   
<p>Filmik którego Twoja przeglądarka nie wyświetli :( </p> </video>

Atrybut poster wyświetla obrazek zanim film zacznie być odtwarzany. Controls powoduje wyświetlenie się guziczków do odtwarzania filmu.

Tagi source wskazują na pliki wideo w różnych kodowaniach tak, by wszystkie przeglądarki rozumiejące tag <video> mogły poprawnie wyświetlić film.

Niestety ze względu na decyzje twórców przeglądarek, przeglądarki komercyjne (IE i Safari) nie wspierają wolnych kodeków, zaś przeglądarki otwarte (Firefox i Chrome) nie wspierają kodeków komercyjnych (o ile system operacyjny użytkownika nie zapewnia wsparcia dla nich). Ma to znaczenie dla użytkowników mobilnych, którzy często mają ograniczony transfer i powinni dostać mniejszy (lepiej skompresowany) film, a jest to trudne.

### Mapy linków

Oryginalnym rodzajem odnośników są aktywne obszary grafiki, które działają jako odsyłacze. Obszary te mogą mieć kształt okręgu (circle), prostokąta (rectangle) i wieloboku (polygon). Dobrym przykładem mapy graficznej może być mapa Polski z naniesionymi na niej siedzibami filii jakiejś dużej firmy, lub budynek szkoły, którego okna są odsyłaczami do opisów różnych sal i pracowni. Najechanie myszą na obszar aktywny zmienia znacznik jej kursora na „łapkę”, a kliknięcie spowoduje przekierowanie do określonego dokumentu lub miejsca w dokumencie. Aby jednak grafika mogła działać jako zespół odnośników należy w znaczniku umieszczającym ją na stronie zastosować atrybut usemap z odpowiednią nazwą.

<img src=”image/mapa\_polski.gif” width=”400” height=”400” usemap=”#mapa\_pol” border=”0”>

Aby zdefiniować obszary aktywne w tej grafice należy gdziekolwiek w tym samym dokumencie html oczywiście w sekcji body zastosować blok <map> definiujący te obszary.

<map name=”mapa\_pol”>   
<area shape=”circle” coords=”50,43,25” href=”strona1.html” alt=”jakaś informacja”>   
<area shape=”rect” coords=”45,26,89,97” href=”strona2.html” alt=”jakaś informacja”>   
<area shape=”poly” coords=”50,43,75,89,150,120,90,38” href=”strona3.html” alt=”jakaś informacja”>

Atrybut shape definiuje kształt obszaru. Wartości można wpisywać w skrócie, tak jak w podanym wyżej

Coords to są współrzędne, które definiują położenie i wielkość obszaru aktywnego w grafice. I tak:

• circle (okrąg) zdefiniowany jest przez współrzędne środka x (50 px), y (43 px) i przez długość promienia r (25 px).   
• rect (prostokąt) wyznaczają dwie pary współrzędnych leżących po przekątnej x1 (45px), y1 (26px), x2 (89px), y2 (97 px). To wystarczy, ponieważ jak powszechnie wiadomo prostokąt ma kąty proste.   
• poly (wielobok) wymaga najbardziej pracochłonnej definicji. Tu do wyznaczenia obszaru służą współrzędne wieloboku x1,y1,x2,y2,x3,y3 …xn,yn. Im bardziej precyzyjne określenie obszaru tym lista współrzędnych jest dłuższa.

Atrybut href to taki sam odsyłacz jak w przypadku prostych linków. Można również zastosować alt, w tym celu, aby po najechaniu myszą na dany obszar uzyskać jakąś informację, która tego obszaru dotyczy w pojawiającym się po sekundzie etykietce („dymku”).

W znaczniku <img> grafiki która została zdefiniowana mapą graficzną został również zastosowany atrybut border z wartością 0 w tym celu aby nie została ona otoczona domyślną ramką o grubości 3px, stanowiącą domyślne oznaczenie linku.

### Komentarz

Jest to fragment kodu źródłowego, który nie ma żadnego wpływu na wygląd i funkcjonowanie dokumentu. Używany w celu czysto informacyjnym (dla webmastera), jest przez przeglądarkę pomijany.

<!-- fragment który się nie wyświetli -->

### Dlaczego tagów jest tak dużo?

Większość znaczników wygląda tak samo, jak przy użyciu każdego, innego częściej używanego. Rzecz w tym że przytłaczająca właśnie większość tych to znaczniki semantyczne. Używanie ich ułatwia innym życie:

* Z tych znaczników korzystają syntezatory mowy w przeglądarce (np. Opera). Gdy ich brak przy wartościowym tekście (np. definicja , akronim lub skrót) syntezator nie będzie ich mógł prawidłowo odczytać.
* Roboty wyszukiwarek internetowych - roboty nie myślą, więc nie domyślą się że ten fragment tekstu taki i taki. Jeśli im pomożemy przy prawidłowym zindeksowaniu twej strony, może to zaowocować to wyższą pozycję podczas wyszukiwania.
* Znaczenie - Tekst obejmowany takim znacznikiem musi być w jakiś sposób wartościowy (np. skrót i akronim, i jego rozwinięcie) czyli nie może być wykorzystywany do ustawiania wyglądu tekstu.

W HTML5 oddzielono strukturę dokumentu od jego wyglądu - innymi słowy, HTML zajmuje się tą praktyczną częścią strony, czyli odpowiada za treść I zawartość strony. Jej formą, czyli wyglądem zajmuje się CSS.

### Przydatne atrybuty

#### Tytuł po najechaniu myszą

<a href="" title="napis">aa</a>

#### Id i klasa elementu

<p class="klasa" id="unikalny"></p>

Te dwa atrybuty ułatwiają odnajdywanie właściwych tagów w drzewie dokumentu. To znaczy, że dzięki nim możemy wskazać które tagi mają zostać ostylowane lub zmienione za pomocą kodu javascript. Robot jest w stanie na podstawie nazw klas wnioskować strukturę strony i lepiej dopasować wyniki wyszukiwania.

Id ma tę własność, że tylko jeden element na stronie (w jednym pliku html) może mieć dane id. Tej wady nie posiadają klasy - wiele tagów może mieć tę samą klasę, a nawet jeden tag może mieć kilka klas, w takim przypadku w ramach atrybutu class rozdziela się te klasy spacją:

<p class="klasa1 klasa2"></p>

### Prawidłowa struktura dokumentu HTML

Dokument HTML ma narzuconą strukturę głównych znaczników określających część nagłówkową (jakie pliki są jeszcze potrzebne do wyświetlenia strony, informacje dotyczące strony) i określających treść, ciało (body) strony.

W HTML5 struktura prezentuje się następująco:

<!doctype html>  
 <html lang="pl-PL">  
 <head>  
 <meta charset="UTF-8">  
 <title></title>  
 </head>  
 <body>

</body>  
 </html>

Linijka doctype określa wersję języka HTML z której korzystamy. Wcześniejsze wersje HTML miały tę linijkę znacznie dłuższą i mniej przyjazną, na przykład dla XHTML:

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

I dla HTML4:

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

Następnie jest tag <html> obejmujący całą stronę. Jest elementem najwyższego rzędu. Jego atrybut *lang* określa w jakim języku (i w jakim dialekcie) jest nasza strona.

Sekcja *head* obejmuje informacje o naszej stronie. W tym miejscu znajduje się znacznik metadanych opisujący użyte przez nas kodowanie - w tym przypadku UTF-8 oraz znacznik <title>który wyznacza tytuł naszej strony. *Title* pojawia się w belce tytułowej przeglądarki oraz w wynikach wyszukiwania Google jako tytuł strony.

Oprócz opisanych tagów, można umieścić dodatkowe tagi <meta> opisujące metadane strony www, tagi <link> i <style> dodające CSS i w końcu <script> dołączający pliki javascript.

Sekcja *body* zawiera tagi które umówiliśmy wcześniej - to one najbardziej interesują użytkownika, to zawartość sekcji *body* stanowi o treści naszej strony www.

### Niektóre przestarzałe tagi i atrybuty

<center> - wyśrodkowanie  
<marquee> - tekst przesuwający się  
<font> - określający formatowanie tagu  
<blink> - tekst migający  
bgcolor="" przypisany body, określał kolor tła  
align="" przypisany obrazkowi, ustawiał opływanie  
vspace="" i hspace="" odległość pozioma i pionowa między elementami  
cellspacing="", cellpadding="" odstępy w komórkach tabeli i między nimi

Te tagi i atrybuty we wcześniejszych wersjach specyfikacji HTML zostały oznaczone jako przestarzałe i nie powinno się ich używać. Służą one do opisu wyglądu elementów, zaburzając tym samym strukturę semantyczną. W HTML do opisu wyglądu powinniśmy użyć CSS.

### Formularze

Obok odsyłaczy hipertekstowych (linków) w samym HTML-u mamy do czynienia z jeszcze jednym interaktywnym obiektem strony.

Jest nim formularz, którym użytkownik przesyła zestandaryzowane dane na pocztę właściciela strony WWW lub do wskazanego adresu URL. Wypełnianie takiego formularza w dużym stopniu przypomina wypełnianie wydrukowanych ankiet lub kwestionariuszy.

Formularz realizowany jest w bloku znaczników <form> … </form> Znacznik form posiada ważne dla jego działania atrybuty takie jak:

• method - sposób wysyłania formularza, który przyjmuje dwie wartości *post* lub *get*. W formularzach html wartością, którą powinno się wpisywać jako wartość tego atrybutu jest wartość „post”. Formularz wysłany metodą GET zostanie wysłany do serwera z wartościami formularza wpisanymi w adres URL. Metoda POST nie ujawnia w pasku adresu wysyłanych danych.

Oznacza to, że lepiej wysyłać formularze metodą POST. Specyfikacja HTTP mówi, że żądania które mają odnieść trwały skutek (np. zapisanie zmian lub wylogowanie się z panelu administracyjnego) powinny być wysłane metodą POST. Dalsze wyjaśnienie różnic pomiędzy GET i POST wykracza poza zakres tematyczny tego podręcznika

• action – akcja wykonywana po naciśnięciu przycisku wysyłki formularza. W tym miejscu możemy podać adres URL skryptu który zajmie się przetworzeniem danych z formularza. Ze względu na to, że w HTML-u mamy do czynienia tylko z formularzami pocztowymi, wartością tego atrybutu jest adres mailowy w następującej postaci:

<form method=”post” action=”mailto:wagnieszka@acn.waw.pl?subject=zgłoszenie”>

Człon adresu występujący po znaku zapytania pozwala na wpisanie tytułu e-maila, który zawiera wypełniony i wysłany formularz.

• enctype – określa sposób prezentacji danych z formularza. Jego wartością w przypadku formularza pocztowego powinno być zawsze „text/plain”. Dzięki temu atrybutowi i tak określonej jego wartości dane zostaną w e-mailu zaprezentowane w sposób czytelny dla odbiorcy

W bloku *form* znajdują się elementy formularza, z których najważniejszym jest przycisk wysłania *submit*.

<input type=”submit” value=”wyślij”>

Znacznik input (stosowany nie tylko dla przycisków, lecz i dla innych elementów) przyjmuje atrybut *type* z wartością *submit* i *value* z taką wartością, która ukaże się na nim jako jego nazwa.

Obok przycisku submit istnieje jeszcze jeden przycisk funkcyjny *reset*, po przyciśnięciu, którego kasowana jest dotychczasowa wypełniona przez użytkownika zawartość

<input type=”reset” value=”rezygnuj”>

Inne przyciski, które nie spełniają specjalnej funkcji a które obsługiwane są zazwyczaj przez polecenia języka skryptowego to button

<input type=”button” value=”kliknij mnie”>

Jeśli nie określimy atrybutu type w znaczniku input to będziemy mieć do czynienia z jednowierszowym polem tekstowym, do którego można wpisać jakąś informację, np. imię, nazwisko, ulicę, numer telefonu.

Aby umieścić pole tekstowe formularza zastosujemy atrybut type z wartością text

<input type=”text” name=”imie” value=”” size=”30” maxlength=”80”>

Atrybut name zarówno w tym typie elementu formularza, jak i w pozostałych ma istotne znaczenie. Pojawia się on w mailu jako etykieta przysłanej danej np. imie=Marek.

Value w tym wypadku określa wartość domyślną, która może się pojawić przed wypełnieniem pola. W tym wypadku wartość ta zawiera pusty znak, czyli nic. size – określa długość pola w znakach (w tym wypadku liczba widocznych w formularzu znaków wynosi 30).

Maxlength – maksymalna liczba znaków mieszcząca się w polu. Odmianą pola tekstowego jest pole password <input type=”password” name=”haslo” value=”” size=”20” >, w którym podczas jego wypełniania znaki zastępowane są gwiazdkami lub dużymi czarnymi kropkami.

Polem tekstowym „swobodnej” wypowiedzi, gdzie można wpisywać wielowierszowe uwagi jest

<textarea> … </textarea>

<textarea cols=”50” rows=”15” name=”uwagi”> Tu wpisz swoje uwagi </textarea>

Atrybuty *cols* i *rows* określają wielkość pola tekstowego kolejno szerokość na 50 kolumn (znaków) i wysokość na 15 wierszy. Tekst umieszczony w bloku *textarea* umieszczany jest opcjonalnie jako domyślna wartość tego pola. Należy uważać, aby wewnątrz bloku textarea nie znalazły się znaki < i >. Takie znaki należy zawsze zamieniać na encje!

Lista wyboru to kolejny ważny element realizowany w bloku <select> … </select>

<select name=”kolory”>

<option value=””>wybierz kolor</option>

<option value=”blue”>niebieski</option>

<option value=”green”>zielony</option>

<option value=”red”>czerwony</option>

<option value=”pink”>różowy</option>

</select>

Kolejne pozycje listy wyznaczają podbloki bloku select w znacznikach <option> … </option> Wartość value tego znacznika określa wartość wysłaną formularza w postaci kolor=pink. W przypadku pominięcia atrybutu value warością wysłaną będzie ta, która znajduje się pomiędzy znacznikami <option> … </option>

Istnieją również atrybuty znacznika select. Poza name mamy do czynienia tu również z atrybutem size, który określa liczbę pozycji widocznych na liście bez jej rozwinięcia i przełącznik multiple, który pozwala na dokonanie wielokrotnego wyboru (wyboru więcej niż jednej opcji) przy użyciu myszy i przycisku Ctrl.

Np. <select name=”kolory” size=”4” multiple>

Element hidden nie jest wyświetlany w formularzu. To tzw. pole ukryte przeznaczone do tego, aby opisać np. znaczenie przysłanych danych lub, co należy z nimi zrobić. Użytkownik nie ma wpływu na zawartość tego pola.

<input type=”hidden” name=”coto” value=”wprowadzić do arkusza z danymi klientów”>

Istnieją w formularzach grupy elementów, które służą do dokonania pojedynczego lub wielokrotnego wyboru. Są to elementy radio i checkbox. np.

Zaznacz przedział wiekowy   
<input type=”radio” name=”wiek” value=”pon. 18”> pon. 18</br>   
<input type=”radio” name=”wiek” value=”19 -28”> 19 -28</br>   
<input type=”radio” name=”wiek” value=”29 -35”> 29 -35</br>   
<input type=”radio” name=”wiek” value=”pow. 35”> pow. 35</br>

W grupie przycisków o wspólnej nazwie (name=”wiek”) można dokonać tylko pojedynczego wyboru. Ponowne zaznaczenie innego elementu w grupie odznacza poprzedni. kolory które lubisz

<input type=”checkbox” name=”kolor” value=”czerwony”>czerwony<br/>   
<input type=”checkbox” name=”kolor” value=”zielony”>zielony<br/>   
<input type=”checkbox” name=”kolor” value=”niebieski”>niebieski<br/>

W tym wypadku można zaznaczyć jedno dwa lub wszystkie pola w grupie kolor.

HTML5 wprowadził nowe typu pól formularzy, jednak ich wsparcie w różnych przeglądarkach, nawet w Firefox i Chrome, jest różne. Dlatego lepiej jest trzymać się powyższych typów pól, ewentualnie wykorzystać skrypty generujące formularze.

### Ramki

W ostatniej dekadzie ubiegłego wieku podstawowym narzędziem do budowy stron były ramki. Okno przeglądarki było podzielone na niezależne od siebie fragmenty, a strony załadowane w ramach tych fragmentów mogły oddziaływać między sobą, zwłaszcza załadować do ramki obok nowy adres URL. Do pracy z ramkami służył tag <frameset>.

Wiązały się z tym pewne problemy - niektóre przeglądarki niepotrafiły sobie poradzić z ramkami. Wyszukiwarki traktowały witrynę wykorzystującą ramki jako zbiór oderwanych od siebie plików html. Istniało ryzyko, że użytkownik otworzy tylko podstronę, a nie szkielet ramek, przez co zobaczy np. samo menu witryny.

Z biegiem czasu ramki jako narzędzie do budowy layoutu zostały wyparte przez tabele, a te przez warstwy. Jedynym polem w którym ramki są wykorzystywane jest dołączenie do naszej strony całkowicie obcej strony www. W ten sposób funkcjonuje np. Youtube który jest dołączony w osadzonej ramce w zewnętrznych serwisach. Ten sam mechanizm wykorzystuje również Facebook żeby wyświetlać guzik "like". Ramka jest wykorzystana by osadzić na stronie www mapkę z Google'a.

Tagiem który na to pozwala jest <iframe>

<iframe src="ramka\_iframe.html" width="400" height="300">   
Twoja przeglądarka nie obsługuje ramek pływających!   
</iframe>

Najważniejszym atrybutem jest src który określa adres URL do otworzenia w ramce. Width i height określają szerokość i wysokość takiej ramki.

## Komentarz warunkowy

Jest to rozszerzenie standardu HTML zaimplementowane w przeglądarce Microsoftu pozwalające ukryć część kodu przed innymi przeglądarkami. Podstawowa składnia prezentuje się następująco:

<!--[if IE]> Ten tekst zostanie wyświetlony w IE <![endif]-->

Jest też możliwość wyszczególnienia która wersja IE ma wyświetlić kod (a pozostałe - ten kod pominą):

<!--[if IE 7]> Ten tekst wyświetli tylko IE7 <![endif]-->

Komentarz warunkowy daje również możliwość przeprowadzenia operacji odwrotnej, to znaczy ukryć kod przed Explorerem:

IE to <!--[if !IE]>--> **nie** <!--<![endif]--> jest dobra przeglądarka!

Komentarze warunkowe są sposobem na okiełznanie wad IE bez rezygnowania z dobrodziejstw HTML5 I CSS3.

IE10 jest wersją przeglądarki która zrywa z komentarzami warunkowymi - traktuje ona komentarze tak, jak Firefox czy Chrome - pomija je. To oznacza, że nie da się stworzyć stylu warunkowego dla IE10 w ten sposób.