

# HTML

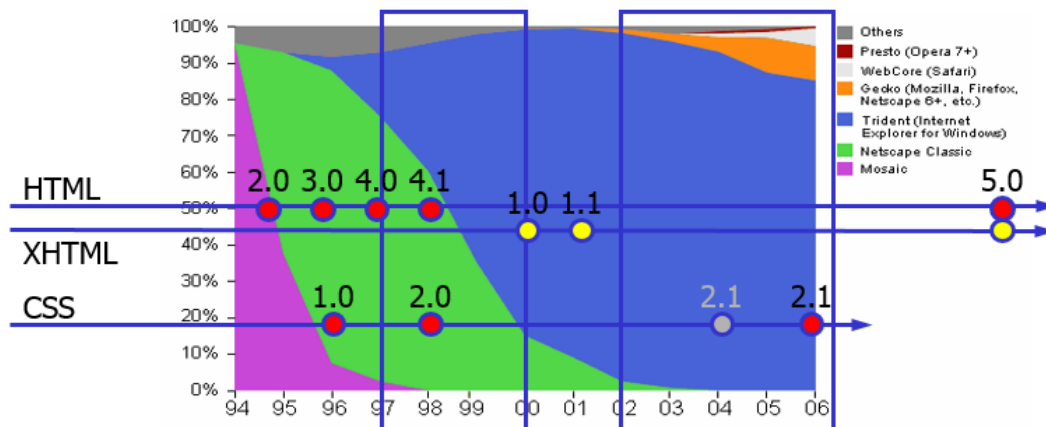
## Rozwój HTML/CSS

- 1980 – prototyp hipertekstowego systemu informacyjnego
- 1989 – propozycja systemu dla Internetu, projekt WWW
- 1991 – pierwsza publicznie dostępna specyfikacja HTML
- 1993 – propozycja (draft) HTML opublikowana przez IETF
- 1994 – pierwsza propozycja CHSS
- 1995 – HTML 2.0, pierwsza oficjalna specyfikacja IETF
- 1996 – CSS level 1: pierwsza oficjalna specyfikacja CSS
  - HTML 3.0; powstaje W3C
- 1997 – HTML 4.0
- 1998 – CSS level 2:
  - HTML 4.01
- 2004 – CSS level 2.1: Zalecenie (2005: draft; 2007: zalecenie)
- 2008 – wersja robocza (working draft) HTML/XHTML 5.0

## Przeglądarki internetowe

Wojny przeglądarek

- Pierwsza wojna przeglądarek: 1997-1999  
MS Internet Explorer vs Netscape Navigator  
Niestandardowe rozszerzenia ("fajniejsze" strony WWW);
- Druga wojna przeglądarek: 2002-?  
MS Internet Explorer vs (Firefox + Safari + Opera)  
Zgodność ze standardami likwidacja monopolu MS;



## Struktura dokumentu html

Dokument HTML ma postać:

```
<!DOCTYPE ... - określenie typu (DTD)
<html>
  <head>          - elementy nagłówka
  </head>
  <body>          - elementy ciała
  </body>
</html>
```

Dokument HTML składa się z dwóch elementów:

- Wskazania DTD
  - Elementu (znacznika) html
- Element html i jego zawartość powinny być zgodne z DTD.

```
|
|-- DTD
|-- <html>
```

Element html powinien zawierać tylko dwa elementy:

- Element head – nagłówek
- Element body – ciało dokumentu

```
|
|-- DTD
|-- <html>
|   |-- <head>
|   |-- <body>
```

Nagłówek zawiera dane ważne dla przeglądarki (np. sposób kodwania znaków, tytuł strony) i wyszukiwarek (np. słowa kluczowe, język dokumentu). Wszystkie elementy widoczne w oknie przeglądarki – tekst, obrazy i inne obiekty, stanowią zawartość elementu body;

```
|
|-- DTD
|-- <html>
|   |-- <head>
|   |   |-- <meta>
|   |   |-- <title>
|   |   |-- <link>
|   |-- <body>
|   |   |-- <h1>
|   |   |-- <div>
|   |   |   |-- <img>
|   |   |   |-- <p>
|   |   |-- <p>
```

## Określenie typu dokumentu

Określenie typu dokumentu (DTD, *document type declaration*) informuje przeglądarkę o tym, który z wariantów języka html/xhtml jest używany.

- Determinuje to sposób interpretacji dokumentu.
- Jeżeli DTD nie zostanie podany (lub zawiera błędy), przeglądarka pracuje w trybie kompatybilności (tzw. tryb *quirks*, tryb dziwactw, pozwalający w miarę prawidłowy sposób prezentacji starych dokumentów HTML); przeglądarka może ignorować niektóre znaczniki lub atrybuty znaczników bądź interpretować je niestandardowo, ale też może uwzględniać znaczniki i atrybuty obecnie nieużywane.

Poprzednia wersja języka, HTML 4.01, definiowała 3 tryby:

- Strict – tryb pełnej zgodności ze standardem; znaczniki oraz atrybuty przestarzałe są ignorowane

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

- Transitional – tryb przejściowy; akceptuje niektóre starsze znaczniki, jednak bez tzw. ramek

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

- Frameset – tryb dla ramek;

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

Obecnie najbezpieczniej jest stosować HTML 5:

- Wersja zalecana

```
<!DOCTYPE html>
```

- Wersja dopuszczalna

```
<!DOCTYPE html SYSTEM "about:legacy-compat">
```

## Element HTML

Elementy HTML (znaczniki i ich zawartość) powinny być zgodne z wybranym DTD.

Element w ogólności ma postać:

- Znacznik otwierający
- Atrybuty
- Zawartość – tekst do wyświetlenia lub inne elementy HTML
- Znacznik zamykający

`<znacznik atrybut="wartość" > zawartość </znacznik>`

Niektóre elementy nie mają znacznika zamykającego, zawartości lub atrybutów, a większość atrybutów jest opcjonalna. Przykłady elementów i znaczników

- Hiperłącze  
`<a href="adres URL">Opis łącza</a>`
- Akapit  
`<p>Treść akapitu`
- Obraz  
``
- Wymuszenie końca linii  
`<br>`

## Nagłówek HTML

Formalnie nieobowiązkowy, jednak zawarte w nim informacje mają istotne znaczenie zarówno dla przeglądarki oraz poprawnego wyświetlania strony, jak i dla wyszukiwarek.

Najważniejsze elementy nagłówka:

- Sposób kodowania znaków i język dokumentu
- Autor, tytuł oraz ikona strony
- Słowa kluczowe, opis zawartości
- Data utworzenia, modyfikacji, ważności
- Generator stron

W nagłówku można dołączyć lub osadzić arkusz stylów (CSS) oraz dołączyć lub osadzić skrypt (JavaScript, VBScript) Przykładowy nagłówek:

```
<head>
<meta http-equiv="Content-Type"
      content="text/html; charset=utf-8">
<meta name="keywords" content="w3C, world wide web, ...">
<meta name="description" content="The world wide web...">
<title>world wide web Consortium - web standards</title>
<link rel="stylesheet" type="text/css"
      href="/StyleSheets/home.css">
<style>
  div.spot-image img {
    margin-bottom: 20px;
  }
</style>
</head>
```

## Nagłówek HTML

- Sposób kodowania znaków  
Istnieje kilka metod kodowania polskich znaków. Jedyne poprawne to ISO 8859-2 (tzw. latin 2) oraz Unicode, w tym popularny UTF-8. Sposób znany z Windows (CP1250) nie jest uznanym standardem międzynarodowym

```
<meta http-equiv="Content-Type"
      content="text/html; charset="iso-8859-2">
<meta http-equiv="Content-Type"
      content="text/html; charset="UTF-8">
```

- Język dokumentu

```
<meta http-equiv="Content-Language"    content="pl">
```

- Tytuł strony, wyświetlany na pasku tytułu przeglądarki oraz na liście zakładek

```
<title>Tytuł strony</title>
```

- Ikona strony na pasku tytułu przeglądarki oraz na liście zakładek

```
<link rel="Shortcut icon" href="URI">
```

- Autor strony

```
<meta name="Author" content="Imię Nazwisko">
```

- Słowa kluczowe, mogą być wykorzystane przez wyszukiwarki

```
<meta name="Keywords"
      content="wyraz1, wyraz2, wyraz3...">
```

- Opis zawartości

```
<meta name="Description"    content="Opis strony">
```

- Data utworzenia i ostatniej modyfikacji

```
<meta http-equiv="Creation-Date"    content="data">
<meta http-equiv="Last-Modified"    content="data">
```

- Data ważności (może być sygnałem do przeładowania strony)

```
<meta http-equiv="Expires"    content="data">
```

- Generator lub edytor strony (niektóre narzędzia do tworzenia stron go dodają)

```
<meta name="Generator" content="edytor">
<meta name="Authoring_tool" content="edytor">
```

## Znaczniki HTML

### Zasady ogólne

- Znaczniki zamykające – tam gdzie są wymagane – powinny być wpisywane, nawet jeżeli przeglądarka radzi sobie bez nich
- Znaczniki i atrybuty mogą być pisane wielkimi i małymi literami; lepiej jest wszystkie znaczniki pisać w jednakowy sposób, zaś aby ewentualna konwersja na xhtml była łatwiejsza – małymi literami
- Znaczniki nie mogą się "zazębiać",

Nieprawidłowo:

```
<b>Bold<i>Bold+Italic</b>Italic</i>
```

Prawidłowo:

```
<b>Bold<i>Bold+Italic</i></b><i>Italic</i>
```

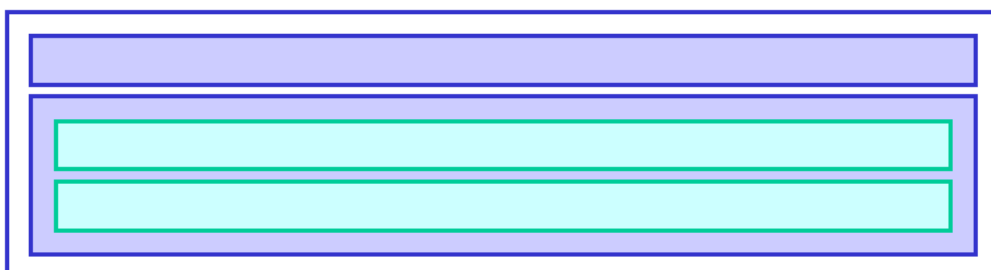
- Elementy dokumentu zawsze mają strukturę drzewa:

```
|
|-- <b>
|   |-- Bold
|   |-- <i>
|       |-- Bold+Italic
|-- <i>
|   |-- Italic
```

### Elementy blokowe, liniowe i pływające

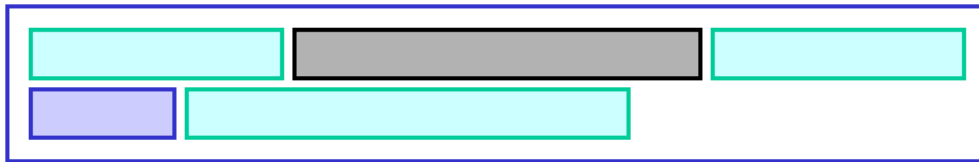
- Elementy blokowe wyświetlane są w prostokątnym obszarze, zajmującym całą szerokość bloku, do którego należą; niektóre elementy blokowe mogą zawierać inne bloki

```
<h1>...</h1>
<div>
  <p>...
  <p>...
</div>
```



- Elementy liniowe nie mają własnego prostokątnego bloku, wyświetlane są między innymi elementami liniowymi bloku swojego właściciela

```
<p>
  <b>...</b>Treść<i>...</i><b>...</b><i>...</i><b>...</b>
</p>
```



## Atrybuty HTML

- Atrybuty mogą być podane w dowolnej kolejności
- Wartości atrybutów powinny być podane w cudzysłowach ( " ") lub apostrofach ( ' ')

```
<a class="footer" href="http://domena.com">Domena</a>
```

- W przypadku zagnieżdżenia, ( " ") i ( ' ' ) należy stosować naprzemiennie

```
<p style="font-face: 'Lucida Console';">Treść
```

- Atrybuty typu logicznego mogą być zapisywane w formie skróconej – jest to zalecany sposób

```
<option selected="selected" id="op1" ...
<option selected id="op1" ...
```

## Wspólne atrybuty znaczników (core attributes):

- id – identyfikator (nazwa) elementu; powinien być unikalny w obrębie dokumentu; może być wykorzystany w skryptach oraz arkuszach stylów

```
<div id="header"> ... </div>
<div id="footer"> ... </div>
```

- class – przypisuje elementowi nazwę lub nazwy klas arkusza stylów, do których element zostaje przypisany

```
<p class="article">Treść
<p class="comment">Komentarz
```

- style – styl elementu, nadpisujący styl odziedziczony

```
<p style="color: red">Czerwony
```

- title – pomocniczy tytuł elementu, wyświetlany jako "hint"

```
<a href="..." title="link do ...">...</a>
```

- onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup – zdarzenia, na które może reagować element;

```

```

- Niektóre elementy mogą być pozbawione obsługi niektórych zdarzeń, inne mogą obsługiwać dodatkowe zdarzenia, np.:
  - body – onload, onunload,
  - form – onsubmit, onreset,
  - input, textarea – onselect, onchange

## **Znaczniki HTML – przewodnik na stronach W3Schools:**

- Szczegółowe wyjaśnienia z przykładami użycia, poradnik HTML na stronach W3C:  
<https://www.w3schools.com/html/default.asp>
- Indeks elementów HTML:  
<https://www.w3schools.com/tags/default.asp>
- Formularze HTML:  
[https://www.w3schools.com/html/html\\_forms.asp](https://www.w3schools.com/html/html_forms.asp)

## **Elementy strukturalne – blokowe**

- h1, h2, ..., h6 – nagłówki
- p – akapity
- ol, ul, li oraz dl, dt, dd – listy
- table, tr, td – tabele
- div – bloki
- form, fieldset - formularze

Rzadziej używane elementy blokowe:

- blockquote – cytat
- pre – tekst preformatowany
- address - adres

## **Elementy strukturalne – liniowe**

- em – emfaza (wyróżnienie); zazwyczaj kursywa
- strong – silna emfaza; zazwyczaj pogrubienie
- cite – cytowanie lub wskazanie źródła
- code – kod komputerowy; czcionka monotypiczna
- kbd – tekst, który ma być wprowadzony z klawiatury
- abbr, acronym – skrót, akronim – objaśniony przez atrybut title
- sub – indeks dolny
- sup – indeks górny

Rzadziej używane elementy blokowe:

- var – nazwa zmiennej
- samp – przykładowy wydruk programu komputerowego



## Nagłówki: h1, h2, ..., h6

- Definiują logiczną strukturę dokumentu; powinny być używane w układzie hierarchicznym

```
<h1>Tytuł</h1>
  <h2>Podtytuł</h2>
    <p>Treść
  <h2>Podtytuł</h2>
    <p>Treść
```

- Znacznik zamykający – wymagany
- Atrybuty:  
id, class, style, title,  
*align* – sposób wyrównania (*left, center, right, justify*)

## Akapit - p

- Reprezentuje akapit (paragraf) tekstu; następujące po sobie akapity są oddzielone przerwą; element p nie może zawierać innych elementów blokowych (w tym innych akapitów)

```
<h2>Tytuł</h2>
<p>Treść
<p>Treść
```

- Znacznik zamykający – opcjonalny;
- Atrybuty:  
id, class, style, title,  
*align* – sposób wyrównania (*left, center, right, justify*)

## Listy numerowane: ol

- Lista uporządkowanych (numerowanych) elementów; poszczególne elementy są typu li; listy mogą być zagnieżdżone

```
<ol>
  <li>Element
  <li>Element
</ol>
```

- Znacznik zamykający – wymagany
- Atrybuty:  
id, class, style, title,  
*start* – numer elementu początkowego (*start="3"*)  
*type* – sposób numerowania (*l, a, A, i, I, np. type="i"*)

## Listy wypunktowane: ul

- Lista nieuporządkowanych (wypunktowanych) elementów; poszczególne elementy są typu li; listy mogą być zagnieżdżone

```
<ul>
  <li>Element
  <li>Element
</ul>
```

- Znacznik zamykający – wymagany
- Atrybuty:  
id, class, style, title,  
*type* – sposób wypunktowania (*circle*, *disc*, *square*)

## Element listy numerowanej lub wypunktowanej: li

- Zawartość list ol oraz ul mogą stanowić wyłącznie elementy list, definiowane jako li

```
<ul>
  <li>Element
  <li>Element</li>
</ul>
```

- Znacznik zamykający – opcjonalny
- Atrybuty:  
id, class, style, title,  
*value* – numer bieżącego elementu listy (np. *value*="3")

## Listy definicji: dl

- Lista definicji, składająca się z terminów oraz ich definicji;

```
<dl>
  <dt>Termin
  <dd>Definicja terminu
</dl>
```

- Znacznik zamykający – wymagany
- Atrybuty:  
id, class, style, title,

## Elementy listy definicji: dt oraz dd

- Zwykle na liście definicji na przemian występują termin (element dt) oraz definicja terminu (dd)

```
<dl>
  <dt>Termin
  <dd>Definicja terminu
</dl>
```

- Znacznik zamykający – opcjonalny
- Atrybuty:  
id, class, style, title,

## Elementy strukturalne – liniowe

- em – emfaza (wyróżnienie); zazwyczaj kursywa
- strong – silna emfaza; zazwyczaj pogrubienie
- cite – cytowanie lub wskazanie źródła
- code – kod komputerowy; czcionka monotypiczna
- kbd – tekst, który ma być wprowadzony z klawiatury
- sub – indeks dolny
- sup – indeks górny

Sposób wyświetlania może zależeć od przeglądarki

- Znacznik zamykający – wymagany
- Atrybuty:  
id, class, style, title,

## Elementy formatowania tekstu – liniowe

- b – wytłuszczenie (bold)
- i – kursywa (italic)
- tt – dalekopis (teletype), czcionka monotypiczna
- big – czcionka powiększona
- small – czcionka pomniejszona

Sposób wyświetlania może zależeć od przeglądarki

- Znacznik zamykający – wymagany
- Atrybuty:  
id, class, style, title,

## Obraz: img

- Wstawia do dokumentu obraz

```

```

- Znacznik zamykający – zabroniony
- Atrybuty:  
id, class, style, title,  
src – lokalizacja pliku graficznego – obowiązkowy  
alt – tekst wyświetlany zamiast obrazka – obowiązkowy  
width – szerokość [px],  
height – wysokość [px],  
ismap, usemap – obrazek jako mapa odnośników,

*border – szerokość ramki [px]*

*hspace, vspace – odstępy wokół obrazka [px]*

*align – wyrównanie (top, middle, bottom, left, right)*

## Obraz: atrybut src

- Lokalizacja pliku graficznego może być bezwzględna lub względna; obrazek może pochodzić z innej domeny niż dokument, na którym jest umieszczony (prawa autorskie!)

```
  

```

- Nazwa pliku nie powinna zawierać znaków specjalnych (@, #) oraz znaków diakrytycznych, nawet jeżeli system operacyjny na to pozwala, a także powinna być zapisana małymi literami – inaczej serwer może nie odnaleźć pliku.

## Obraz: atrybut alt

- Alternatywny tekst – jest używany gdy przeglądarka nie wyświetla obrazków

```

```

- Należy unikać opisów niczego nie wnoszących, w rodzaju

```

```

- Atrybut alt nie zastępuje atrybutu title – można używać obu:

```

```

## Obraz: atrybuty width oraz height

- Szerokość i wysokość – jeżeli zostanie podany JEDEN parametr, obrazek zostanie przeskalowany proporcjonalnie;

```

```

Podanie obu wymiarów umożliwia nieproporcjonalne przeskalowanie obrazka

```

```

- Podanie rozmiarów nie wpływa na rozmiar pobieranego pliku (przeskalowania dokonuje przeglądarka, nie serwer)
- Dobrze jest podać rozmiar nawet gdy obrazek nie jest przeskalowany – przeglądarka może prawidłowo ustalić położenie obrazka na stronie zanim zostanie pobrany

## Obraz: element picture (HTML 5)

- Wstawia do dokumentu obraz; Umożliwia podanie wielu alternatywnych plików graficznych (zależnie od medium i możliwości przeglądarki)

```
<picture>  
  <source media="(min-width:650px)" srcset="img1.jpg">  
  <source media="(min-width:465px)" srcset="img2.jpg">  
    
</picture>
```

- Znacznik zamykający – obowiązkowy
- Elementy:
  - source – alternatywne pliki graficzne
  - img – jeżeli żadne medium nie zostanie dopasowane oraz dla starszych przeglądarek, nieobsługujących <picture>

### Film: element video (HTML 5)

- Wstawia do dokumentu film; Umożliwia podanie wielu alternatywnych plików wideo (zależnie od możliwości przeglądarki)

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video...
</video>
```

- Znacznik zamykający – obowiązkowy
- Elementy:
  - source – alternatywne pliki wideo
 Obsługiwane są formaty mp4, ogg oraz wav (Chrome, Firefox i Opera obsługują wszystkie, inne przeglądarki niekoniecznie)

### Odnośniki

Odsyłacze (łącza, linki) na stronach WWW mogą pełnić kilka różnych funkcji:

- Tworzenie "spisu treści" dokumentu (kliknięcie odsyła do innego miejsca w tym samym dokumencie) oraz "menu" serwisu (kliknięcie odsyła do innego dokumentu w tym serwisie WWW)
- Tworzenie łączy do innych stron WWW; Autor zwykle starannie wybiera adresy, korzystanie z nich jest dużo lepsze niż korzystanie z wyszukiwarki
- Tworzenie łączy do plików dowolnego typu umieszczanych w serwisach WWW (kliknięcie powoduje otwarcie dokumentu w oknie przeglądarki albo propozycję zapisania na dysku)
- Tworzenie łączy do innych usług Internetu, jak poczta lub ftp (kliknięcie może powodować uruchomienie innej aplikacji)

### Odnośnik: element a (anchor)

- Kotwica – element łączy hipertekstowego; Odsyłacz do dokumentu HTML lub innych zasobów bądź usług dostępnych w sieci Internet

```
<a href="adres-URI">Treść</a>
```

- Znacznik zamykający – wymagany
- Treść – tekst lub obraz wyświetlany jako łącze (np. tak: łącze)
- Atrybuty:
  - id, class, style, title,
  - href – lokalizacja zasobu,
  - download – jeżeli odnośnik wskazuje plik do pobrania
  - target – wskazuje gdzie ma zostać otwarty docelowy dokument, np. "\_blank" – nowa karta (domyślnie "\_self", to samo okno)

## Odnośnik zewnętrzny do strony WWW:

`href="http://domena/folder/plik#etykieta"`

- Adres powinien zawierać określenie protokołu (http), adres domeny oraz opcjonalnie folderu, pliku i etykietę, np.:

`http://domena.com/publ/oferta/kat.htm#zestawy`



- Przykłady

```
href="http://domena.com"
href="http://domena.com/folder"
href="http://domena.com/folder/plik.htm"
href="domena.com:10000/"
```

- Wiele łączy ma adres zdegradowany:
  - http jest domyślnym protokołem, można pominąć
  - bez wskazania pliku – serwer poszuka dokumentu domyślnego (zwykle index.html, index.htm, index.php, default.html, ...)
- Można też podać numer portu (jeżeli nie – domyślnie 80); Niektóre usługi serwera WWW (webmin, usermin) mogą używać portu innego niż 80

## Odnośnik wewnętrzny – do podstrony:

```
href="./ścieżka/względna/dokumentu.htm"
href="../albo/katalogu/"
```

- Adres można podać jako względny ("./" – katalog bieżący, "../" – katalog poziom wyższy od bieżącego) w stosunku do dokumentu aktualnie wyświetlanego;
- Można podać adres
  - konkretnego dokumentu html (plik html, htm, xht, xhtml) lub skryptu generującego html (plik php, jsp, jsf, asp, ...)
  - katalogu – serwer poszuka w nim dokumentu domyślnego (zwykle index.html, index.htm, index.php, default.html, ...)

## Odnośnik wewnętrzny – do etykiety:

`href="#nazwaEtykiety"`

- Etykieta może zostać zdefiniowana jako dowolny element z atrybutem *id*:

```
<h3 id="nazwaEtykiety">Seksja 2.4</h3>
```

## **Odnosnik wewnętrzny lub zewnętrzny do pliku nie-html:**

```
href="http://domena/folder/plik.pdf"
href="./pliki/plik.pdf"
```

- Adres powinien zawierać określenie protokołu (http), adres domeny, folderu oraz nazwę pliku; dla łączy wewnętrznych można podać tylko ścieżkę względną i plik
- Działanie przeglądarki zależy od typu pliku, samej przeglądarki oraz zainstalowanych wtyczek: dokument może zostać otwarty w oknie przeglądarki lub zapisany na dysku.

## **Odnosnik zewnętrzny do usługi ftp:**

```
href="ftp://domena/folder"
href="ftp://uzytkownik@domena/folder"
```

- Adres powinien zawierać określenie protokołu (ftp), adres domeny, folderu oraz nazwę pliku; opcjonalnie można podać nazwę użytkownika
- Działanie zależy od przeglądarki; na ogół przeglądarka ma wbudowanego klienta ftp.

## **Odnosnik zewnętrzny do usługi mail:**

```
href="mailto:uzytkownik@domena"
```

- Adres powinien zawierać określenie protokołu (mailto) oraz pełen adres email
- Działanie zależy od przeglądarki; na ogół przeglądarka otwiera domyślnego klienta poczty z nowo utworzoną wiadomością, z wypełnionym polem adresu
- Można również podać kilku adresatów – kolejne adresy należy rozdzielić znakiem ; (średnik):

```
href="mailto:uz1@domena; uz2@innadomena"
```

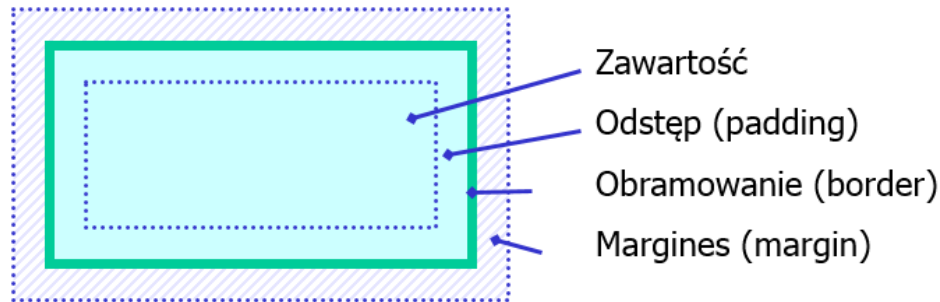
- Można podać dodatkowo temat, adresata kopii (cc), a nawet treść wiadomości:

```
href="mailto:uz@domena? subject= temat
& cc= adresatKopii@domena
& bcc= adrukrytejKopii@domena
& body= trescwiadomosci"
```

(odsyłacz nie powinien zawierać spacji ani znaków końca linii)

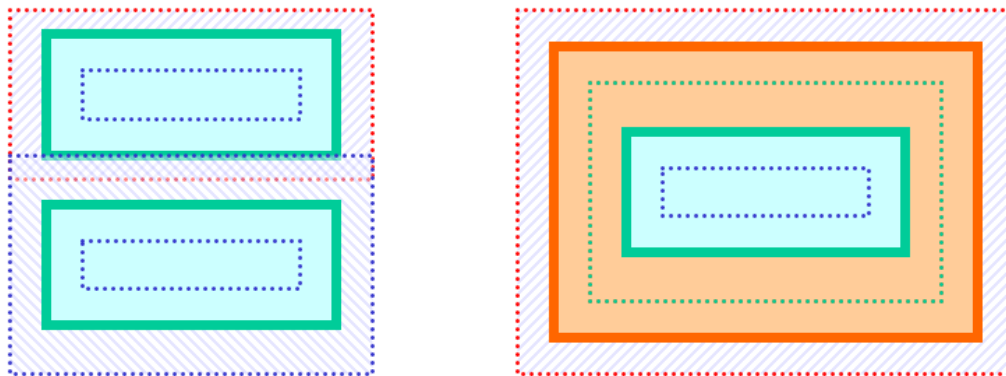
## Model pudełkowy HTML

Elementy blokowe – w tym table oraz div – są wyświetlane według tzw. modelu pudełkowego:



- Szerokość każdej krawędzi można ustawić oddzielnie (CSS)
- Tło odstępu jest takie samo, jak zawartości, natomiast marginesy są przezroczyste – widać tło pudełka spod spodu

Korzystanie z modelu pudełkowego komplikuje się gdy wchodzi w grę interakcje:



- Marginesy sąsiednich elementów nie sumują się
- Jeżeli pudełko zawiera inne pudełko – margines wewnętrznego pudełka "sumuje się" z odstępem zewnętrznego

### Tabela: table

- Wstawia do dokumentu tabelę; zawartość i układ tabeli jest zdeterminowany przez elementy zawarte w table (caption, colspan, col, thead, tfoot, tbody, tr, th oraz td):

```
<table> ... </table>
```

- Znacznik zamykający – obowiązkowy
- Atrybuty:
  - id, class, style, title,
  - summary – podsumowanie zawartości
  - width – szerokość [px] lub [%],
  - frame, rules, border – układ i styl obramowania,
  - cellspacing, cellpadding – odstępy między komórkami tabeli

*bgcolor* – kolor tła tabeli

*align* – wyrównanie tabeli względem strony (*left*, *center*, *right*)



### Tabela: atrybut width

- Określa szerokość tabeli w stosunku do zawierającego ją pudełka; może być określona w pikselach lub procentach:

```
<table width="600px"> ... </table>
<table width="75%"> ... </table>
```

- Określając szerokość tabeli w [px] należy uwzględnić rozdzielczość ekranu: obecnie w 99% 800x600 lub wyższe. Tabele z szerokością w [px] są często źle drukowane!
- Domyślnie szerokość tabeli zależy od zawartości

### Tabela: atrybuty frame, rules, border

- Określają sposób wyświetlania krawędzi tabeli:

```
<table frame="hsides" rules="rows" border="2">
...
</table>

<table border="1">
...
</table>
```

- frame – rodzaj obramowania wokół tabeli: void, above, below, hsides, vsides, lhs, rhs, box, border (domyślnie);
- rules – krawędzie wewnętrzne: none, groups (np. między thead a tbody), rows, cols, all (domyślnie)
- border – szerokość obramowania [px]; domyślnie tabela ma obramowanie i wszystkie wewnętrzne krawędzie o szerokości 1px; Ustawienie border="0" implikuje frame="void" oraz rules="none"

### Tabela: atrybuty cellpadding, cellspacing

- Określają szerokość odstępów między komórkami tabeli (jak w modelu pudełkowym):

```
<table cellpadding="10" cellspacing="5">
...
</table>
```

- Oba parametry można podawać w [px] lub [%];  
Wartości domyślne mogą być różne w różnych przeglądarkach.

### Tabela: atrybut bgcolor

- Określa kolor tła tabeli:

```
<table bgcolor="#E0E0FF">
...
</table>
```

- Kolor można podać w postaci nazwy (tylko 16 podstawowych kolorów) albo szesnastkowo RGB (np. teal = #008080)
- Jeżeli komórki mają inny kolor tła, niż tabela, to kolor tła tabeli jest widoczny tylko w marginesach komórek

## Tabela: struktura tabeli

- Tylko wiersze (elementy tr) i komórki (elementy th i td) są niezbędne do utworzenia tabeli; pozostałe elementy są stosowane wg potrzeb

```
<table>
| -- <caption> opis (tytuł)
| -- <colgroup> parametry kolumn
|   | -- <col>
| -- <thead> nagłówek
| -- <tfoot> stopka
| -- <tbody> właściwa zawartość
|   | -- <tr> wiersz
|     | -- <th> komórka nagłówka
|     | -- <td> komórka danych
```

- caption – tekst stanowiący tytuł tabeli; domyślnie wyświetlany nad tabelą
- colgroup, col – opcjonalne elementy pozwalające określić parametry kolumn, np. wyrównanie tekstu, czcionka,
- thead – część nagłówkowa tabeli (zawartość - jak tbody); opcjonalna, jednak jeżeli występuje to musi być przed tbody
- tfoot – stopka tabeli, (zawartość - jak tbody); opcjonalna, jeżeli występuje to za thead i przed tbody
- tbody – ciało (zasadnicza część tabeli); może wystąpić kilka sekcji tbody w jednej tabeli (patrz rules="groups"); element tbody powinien zawierać jeden lub więcej wierszy (tr)
- tr – wiersz tabeli; powinien zawierać przynajmniej jedną komórkę – th albo td
- th – komórka nagłówkowa (przeglądarki wyróżniają komórki nagłówkowe, zazwyczaj przez wytłuszczenie tekstu)
- td – komórka danych

## Rząd tabeli: tr

- Reprezentuje rząd tabeli, składający się z komórek th i/lub td:

```
<table>
  <tr align="center" valign="middle">
    <td> ...
```

- Znacznik zamykający – opcjonalny
- Atrybuty:
  - id, class, style, title,
  - align – wyrównanie komórek w poziomie (left, center, right, center, justify, char)
  - valign – wyrównanie zawartości komórek w pionie (top, middle, bottom, baseline)
  - char – znak wg którego wyrównana będzie zawartość dla align="char"

## Komórki tabeli: th oraz td

- Reprezentuje komórkę tabeli: th – komórka nagłówkowa, td – komórka danych:

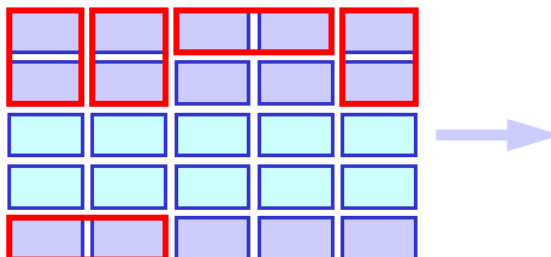
```
<tr> <th align="center">Cena
<tr> <td align="char" char=",">249,99
```

- Znacznik zamykający – opcjonalny
- Atrybuty:
  - id, class, style, title,
  - axis, headers, scope – powiązania komórki (np. dla niewidomych)
  - rowspan – liczba wierszy zajętych przez komórkę
  - colspan – liczba kolumn zajętych przez komórkę
  - align – wyrównanie zawartości komórki w poziomie (jak w tr)
  - valign – wyrównanie zawartości komórki w pionie (jak w tr)
  - char – znak wg którego wyrównana będzie zawartość dla align="char"

## Komórki tabeli: atrybuty rowspan, colspan

- Umożliwiają tworzenie komórek obejmujących kilka wierszy i/lub kolumn:

```
<td rowspan="2" colspan="3">
```



Lp.	Miasto	Zyski			Suma
		III	IV	V	
1	Gdańsk	20	20	20	60
2	Sopot	5	5	10	20
3	Gdynia	10	20	10	40
RAZEM		35	50	40	125

## Pudełko: div

- Umożliwia grupowanie elementów i tym samym nadanie dokumentowi HTML pożądanej struktury; Pożądane cechy uzyskuje się przez arkusze stylów CSS:

```
<div class="contens"> ... </div>
```

- Znacznik zamykający – obowiązkowy
- Atrybuty:
  - id, class, style, title,

## Pudełka do strukturyzacji treści

- Spełniają podobną rolę i mają takie same możliwości, co zwykłe pudełka (div), ale dodatkowo informują o roli pudełka w strukturze strony
- Pożądane cechy uzyskuje się przez arkusze stylów CSS:

```
<section class="blog">
  <article class="post"> ... </article>
  <article class="comment"> ... </article>
</section>
```

- Rodzaje: header, footer, nav, aside, main, section, article
- Znacznik zamykający – obowiązkowy
- Atrybuty:  
id, class, style, title,

## Pudełka do strukturyzacji treści - rodzaje

- <main> dotyczy treści unikalnych dla strony; należy go użyć tylko raz na stronie, najlepiej bezpośrednio w <body> (nie powinien być zagnieżdżony w innych elementach)
- <article> blok powiązanych treści, które same w sobie mają sens bez reszty strony (np. pojedynczy post na blogu)
- <section> jest podobny do <article>, ale służy raczej do grupowania elementów tworzących funkcjonalnie jedną całość (np. zestaw nagłówków artykułów i streszczeń). Za najlepszą praktykę uważa się rozpoczynanie każdej sekcji nagłówkiem (h1...h6); Można podzielić <article> na kilka <section> lub <section> na kilka <article>, w zależności od kontekstu.
- <aside> zawiera treści, które nie są bezpośrednio związane z główną treścią, ale mogą zawierać dodatkowe informacje, które są z nią pośrednio związane (menu boczne, powiązane linki, notka o autorze, ankieta związana z główną treścią itp.)
- <header> reprezentuje grupę treści wprowadzających. Jeśli jest to element podrzędny <body>, określa globalny nagłówek strony internetowej, ale jeśli jest elementem podrzędnym <article> lub <section>, definiuje nagłówek dla tej sekcji
- <nav> zawiera główne funkcje nawigacji na stronie, zwykle jednakowe na wszystkich podstronach serwisu
- <footer> reprezentuje grupę końcowych treści strony

## Struktura strony WWW (layout)

Istnieje kilka metod nadania stronie WWW pożądanej struktury wizualnej (tzw. layout):

- Strukturalny HTML – nagłówki, akapity, listy, odnośniki; Tylko bardzo proste strony, niewielkie możliwości
- Tabele – kilka wierszy i kolumn dzieli stronę na nagłówek, menu, treść i stopkę – prosta realizacja, gwarantowany rezultat; W3C odradza tę metodę, poza tym strona nie jest responsywna
- Pudełka, w tym strukturalne (header, nav, main, aside, footer) – wymaga dużego doświadczenia w CSS, zwłaszcza pozycjonowania, a także JavaScript/jQuery oraz mnóstwa pracy Metoda zalecana przez W3C i powszechnie stosowana

**Rozwiązanie:** gotowe szablony stron, darmowe lub odpłatne;  
Najlepsze bazują na Bootstrapie oraz jQuery i są responsywne

## Zadania

Proszę opracować dokument HTML jak podstronę witryny „Życiorysy sławnych ludzi”. Należy jak najdokładniej odtworzyć strukturę treści, jednak bez jakiegokolwiek formatowania (to będzie zrobione później, przy pomocy CSS)

1. Dokument powinien mieć prawidłową strukturę, z DTD oraz html, head i body;
2. Nagłówek HTML powinien zawierać wszystkie obowiązkowe i zalecane elementy
3. W treści należy użyć wyłącznie elementy strukturalne HTML, blokowe i liniowe;
4. Do strukturyzacji treści należy użyć elementów header, nav, aside, main oraz section; Panel boczny powinien zawierać spis treści w formie odnośników, sekcja główna – życiorys, podzielony na sekcje.
5. Obrazki i ich podpisy należy umieścić w osobnych akapitach
6. Tabele należy przenieść z zachowaniem łączenia komórek oraz rozróżnienia między komórkami nagłówkowymi i komórkami danych
7. Menu i menu boczne powinno mieć formę listy wypunktowanej z odnośnikami,
8. Spis źródeł (na końcu dokumentu) powinien mieć formę listy numerowanej z odnośnikami