

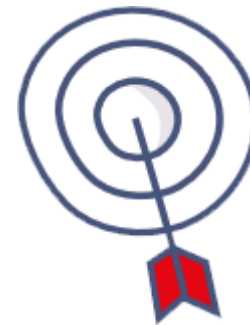
[HTTPS://MATS.TEBU.EDU.PL](https://mats.tebu.edu.pl)



<!DOCTYPE HTML>

— <HTML> —

<!-- HYPERTEXT MARKUP LANGUAGE  
HIPERTEKSTOWY JĘZYK ZNACZNIKÓW -->





# HTML – DO CZEGO SŁUŻY?



**HTML pozwala opisać strukturę informacji zawartych wewnątrz strony internetowej, nadając odpowiednie znaczenie semantyczne poszczególnym fragmentom tekstu** np. formując hiperłącza, akapity, nagłówki, listy, tabele – oraz **osadza w tekście dokumentu obiekty plikowe**, na przykład multimedia, lub elementy baz danych, na przykład interaktywne formularze danych.

**HTML umożliwia określenie wyglądu dokumentu w przeglądarce internetowej.**

Do szczegółowego opisu formatowania akapitów, nagłówków, użytych czcionek i kolorów, zalecane jest wykorzystywanie kaskadowych arkuszy stylów (w skrócie CSS).

Aktualnie obowiązującą wersją HTML jest W3C HTML 5 (2014r., 2017r. v5.2). 

 **HTML opisuje strukturę stron internetowych, a CSS określa formę jej prezentacji w przeglądarce.**





# HTML – KOMPONENTY, SKŁADOWE

- Różne typy danych
- Referencje znakowe, odwołania w postaci encji
- **Deklaracja typu dokumentu**
- **Znaczniki (i ich atrybuty)**





# HTML – KOMPONENTY, TYPY DANYCH

HTML definiuje **kilkanaście typów danych**, wprowadzanych m.in. jako wartości elementów lub atrybutów.

Są to m.in. skrypty (script data), dane arkuszy stylów (stylesheet data), identyfikatory, nazwy, kolory, adresy URI, liczby, języki, deskryptory mediów, kolory, kodowania znaków, data i czas, itp.

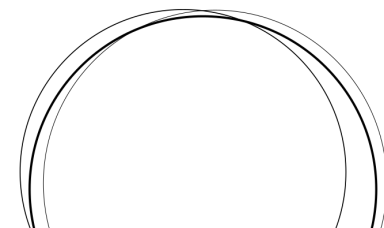
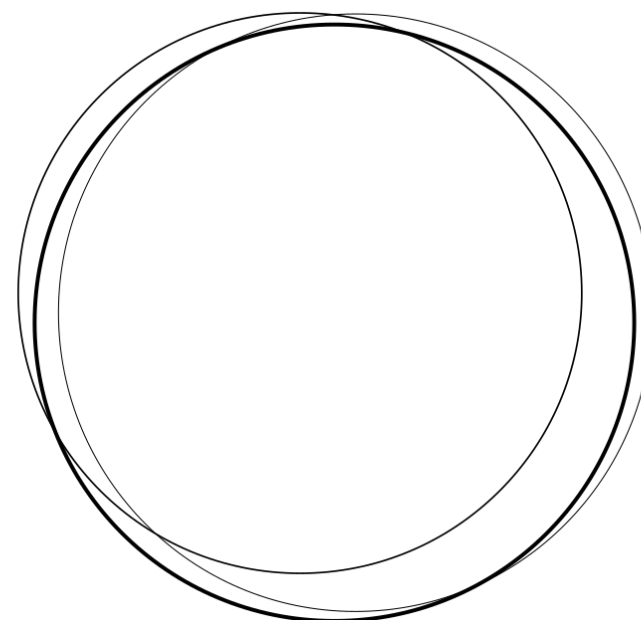
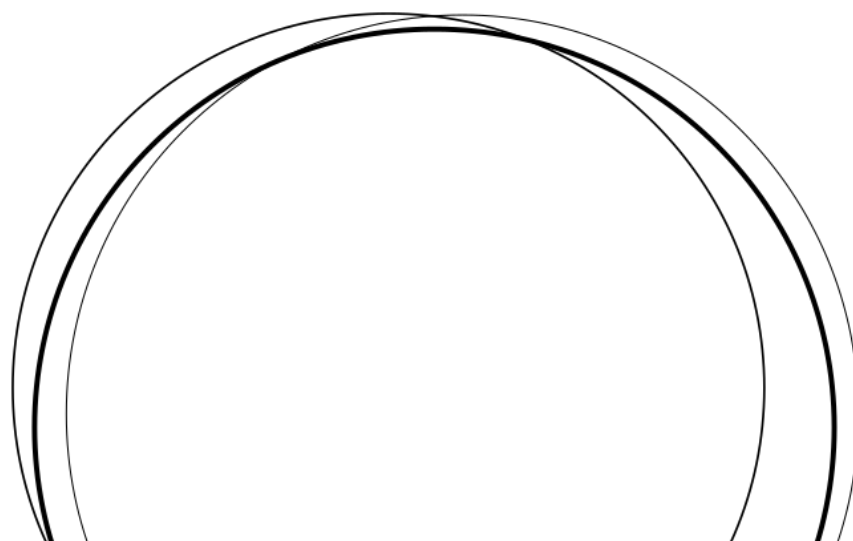
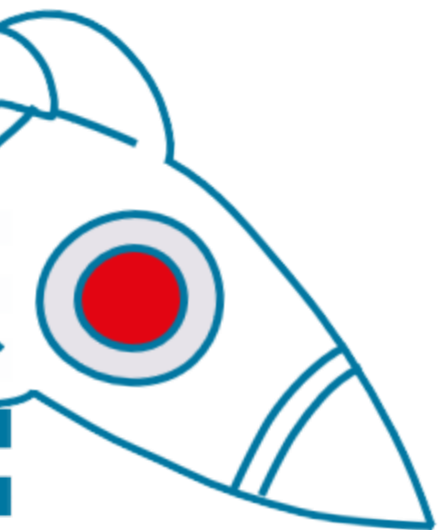




# HTML – KOMPONENTY, REFERENCJE ZNAKOWE I ENCJE

Począwszy od wersji 4.0, HTML definiuje listę 252 encji, pozwalających na zapisanie określonych znaków za pomocą specjalnych stałych.

Przykłady referencji znakowych **&amp;** **&lt;** **&nbsp;**  





# HTML – KOMPONENTY, DEKLARACJA <!DOCTYPE>

Aby uaktywnić definicję typu dokumentu (DTD) do celów poprawnej walidacji utworzonej strony, a także w celu uniknięcia działania przeglądarek w trybie quirks\*, dokument powinien rozpoczynać się deklaracją typu dokumentu - DOCTYPE.

DTD, do którego odwołuje się sekcja DOCTYPE, **zawiera gramatykę i zasady wykorzystywania znaczników w dokumencie podporządkowanemu określönemu DTD.**

Większość graficznych przeglądarek korzysta z sekcji DOCTYPE, a także innych dodatkowych informacji, warunkując tryb renderowania (finalnego widoku) danej strony.

\* **Quirks** mode, zwany też trybem osobliwości – sposób interpretacji kodu przez przeglądarki internetowe, zakładający wykorzystywanie przestarzałych konstrukcji językowych niezgodnych ze standardami World Wide Web Consortium (W3C)



# HTML – KOMPONENTY, DEKLARACJA `<!DOCTYPE>` - ZNACZNIK SPECJALNY

## Przykłady deklaracji DOCTYPE dla dokumentów pisanych w starszych wersjach

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

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

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

 Aby określić definicję typu dokumentu na HTML5 na początku pliku określamy DOCTYPE

`<!DOCTYPE html>`

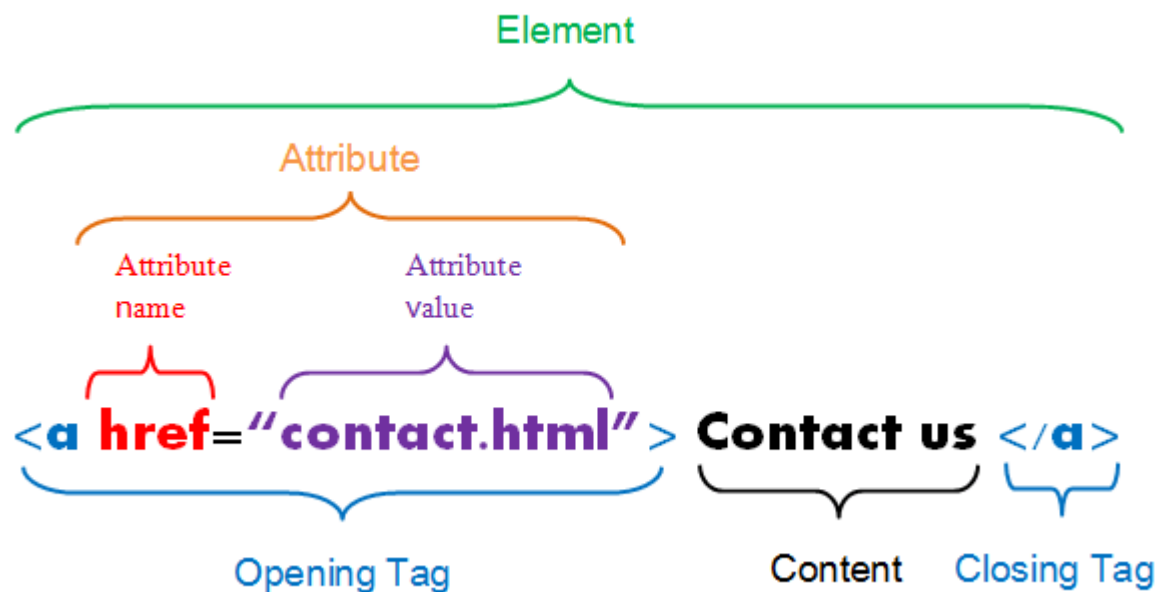






# HTML – KOMPONENTY, ZNACZNIKI

Tag (znacznik) – znak lub słowo kluczowe przypisane do określonego fragmentu informacji, na przykład tekstu, pliku multimedialnego czy strony internetowej. 



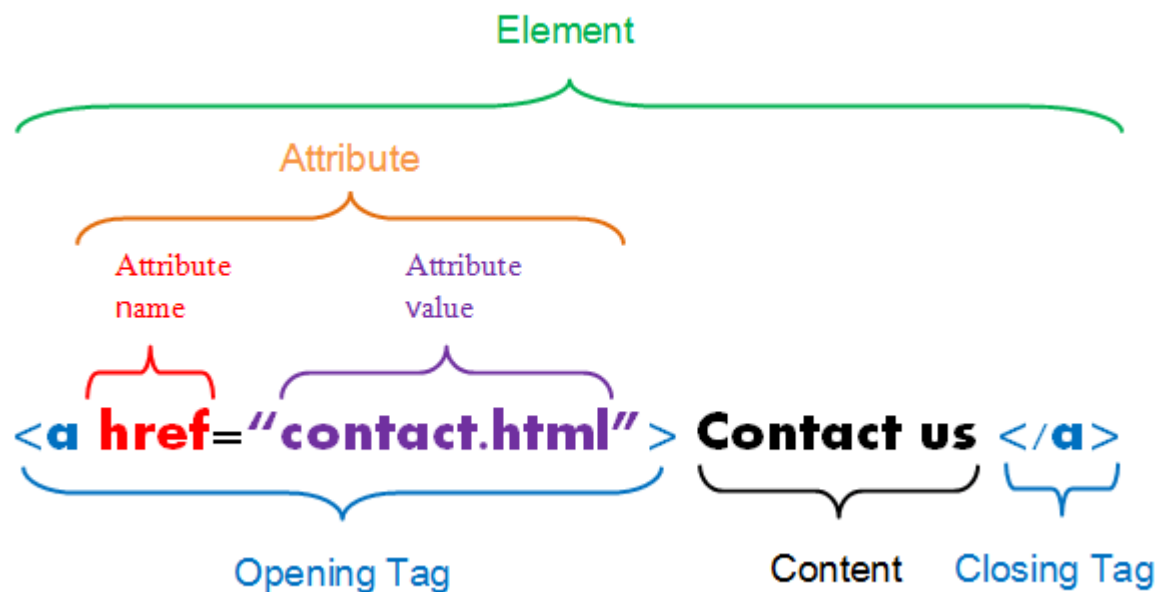
Wartości atrybutów umieszczamy pojedynczym lub podwójnym cudzysłowiu.

Można także opuścić cudzysłów lecz nie jest to zalecane.



# HTML – KOMPONENTY, ZNACZNIKI

Tag (znacznik) – znak lub słowo kluczowe przypisane do określonego fragmentu informacji, na przykład tekstu, pliku multimedialnego czy strony internetowej. 



Wartości atrybutów umieszczamy pojedynczym lub podwójnym cudzysłowiu.

Można także opuścić cudzysłów lecz nie jest to zalecane.



# HTML – KOMPONENTY, ZNACZNIKI

## Znacznik pojedynczy

znacznik  
otwierający `<img` `src="photo.jpg"` znacznik  
zamykający `>`

nazwa  
znacznika atrybut



Zamknięcie znacznika pojedynczego jak podwójnego nie jest konieczne. Jego wykonanie natomiast nie powoduje błędu walidacji.

## Znacznik podwójny

znacznik  
otwierający `<h1` `class="note"` `Tekst nagłówka` `</h1>` znacznik  
zamykający

nazwa  
znacznika atrybut Treść modyfikowana  
przez znacznik





# HTML – KOMPONENTY, ZNACZNIKI (TAGI) SAMOZAMYKAJĄCE

ang. self-closing tags / empty tags / void tags

Tagi HTML, które nie muszą być zamknięte, zwane także tagami pustymi, pojedynczymi, bez treści, bez tagów potomnych (dzieci). Wg specyfikacji w kodzie zapisujemy je bez tagu zamykającego **</tag>**

## Znaczniki samozamykające w HTML5

- |                         |                        |
|-------------------------|------------------------|
| → <b>&lt;br&gt;</b>     | → <b>&lt;area&gt;</b>  |
| → <b>&lt;img&gt;</b>    | → <b>&lt;base&gt;</b>  |
| → <b>&lt;input&gt;</b>  | → <b>&lt;col&gt;</b>   |
| → <b>&lt;link&gt;</b>   | → <b>&lt;embed&gt;</b> |
| → <b>&lt;meta&gt;</b>   | → <b>&lt;hr&gt;</b>    |
| → <b>&lt;param&gt;</b>  | → <b>&lt;track&gt;</b> |
| → <b>&lt;source&gt;</b> | → <b>&lt;wbr/&gt;</b>  |





# HTML – STRUKTURA DOKUMENTU – ZNACZNIKI NIEZBĘDNE

```
<!DOCTYPE HTML>
<html lang="pl">
|   <head>
|       ...
|   </head>
|   <body>
|       ...
|   </body>
</html>
```

Poza DOCTYPE pierwszym znaczącym znacznikiem dowolnego dokumentu HTML jest znacznik **<html>**.

**Wskazuje zakres dokumentu HTML.**

Wszystkie inne tagi i treść, z których składa się Twoja strona internetowa, znajdują się wewnątrz tego tagu **<html>** .

**Jeśli umieścisz znacznik HTML poza znacznikiem <html> , przeglądarka może go nie uwzględnić.**



# HTML – STRUKTURA DOKUMENTU – ZNACZNIKI NIEZBĘDNE

```
<!DOCTYPE HTML>
<html lang="pl">
  <head>
    ...
  </head>
  <body>
    ...
  </body>
</html>
```

Znacznik **<head>** (głowa) pojawia się na górze każdego dokumentu HTML.

Wyświetla wewnątrz kilka innych tagów, które służą do podania dodatkowych informacji o stronie internetowej, załadowania danych lub określeniu zachowania przeglądarki.

Tag **<body>** (ciało) zawiera tagi HTML opisujące strukturę i budowę strony internetowej. To w tym znaczniku zawieramy resztę znaczników HTML układających podaną treść w logiczną całość.



# HTML – STRUKTURA DOKUMENTU - <HEAD>



Znacznik **<head>** w HTML jest miejscem m.in. do definiowania metadanych i łączenia zasobów zewnętrznych z naszym dokumentem czyli np. dodawania ostylowania (kaskadowe arkusze stylów / CSS) czy skryptów JavaScript. W sekcji head możemy także dodać wewnętrzne style dokumentu.

**Metadane to zazwyczaj informacje o naszych danych.** Informacje te są wykorzystywane przez wyszukiwarki, przeglądarki, roboty i osoby odwiedzające witrynę w celu określenia treści technicznych wyświetlanych na naszej stronie internetowej.

Informacje umieszczone w tym tagu nie są wyświetlane w przeglądarce w podglądzie strony (jedynie w źródle strony).

**Element <head> jest kontenerem kilku elementów i pełni ważną rolę w definiowaniu informacji o naszej stronie oraz linkowaniu dokumentów.**



```
<html lang="pl">
  <head>
    <title>Podsumowanie HTML</title>
    <base href="https://localhost:5050/">
    <meta charset="UTF-8">
    <meta name="description" content="Strona podsumowanie wiadomości z zakresu HTML">
    <meta name="keywords" content="programownie, HTML, podstawy">
    <meta name="author" content="Marcin Janowski TEB Technikum">
    <meta name="robots" content="noindex,nofollow">
    <meta http-equiv="X-UA-Compatible" content="IE=edge, chrome=1">
    <meta http-equiv="refresh" content="45">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="assets/styles/style.css">
    <style>
      .lead {
        color: ■ #0082fa;
      }
    </style>
    <script src="text/javascript">
      alert('Wyswietl alert!');
    </script>
    <script src="/assets/scripts/skrypty3.js"></script>
    <script src="/assets/scripts/skrypty2.js" async></script>
    <script src="/assets/scripts/skrypty1.js" defer></script>
  </head>
```





# HTML – STRUKTURA DOKUMENTU - <HEAD>

Element **<title>** służy do definiowania tytułu dokumentu HTML. Jest on obowiązkowym elementem każdego dokumentu. Jego zawartość jest wyświetlana przez przeglądarki internetowe na pasku tytułu i w menu ulubionych oraz przez wyszukiwarki internetowe w wynikach wyszukiwania.

Element **<meta>** służy do definiowania metadanych dokumentu, tzn. informacji opisujących ten dokument. Informacje te określa się w postaci par nazwa-wartość. **Atrybutem obowiązkowym elementu meta jest content (czyli wartość), jeśli zdefiniowany jest atrybut http-equiv lub name.** Jeśli żaden z tych atrybutów nie jest zdefiniowany, atrybut content jest zabroniony (np. `<meta charset=„UTF-8”>`).





# HTML – STRUKTURA DOKUMENTU - <HEAD>

```
<!DOCTYPE HTML>
<html lang="pl">
|   <head>
|       <meta charset="UTF-8">
|   </head>
</html>
```

Atrybut **charset** służy do określania kodowania znaków całej strony internetowej - tag <meta> (w HTML5).

Stosowanych jest wiele standardów kodowania znaków, w tym pierwszy, czyli standard ASCII. Nie obejmuje on jednak wszystkich znaków i symboli np. diakrytycznych. Dlatego zgodnie ze specyfikacją HTML5 zaleca się programistom stosowanie UTF-8 (8-bitowy format tłumaczenia Unicode), ponieważ obejmuje on większość symboli i znaków.





# HTML – STRUKTURA DOKUMENTU - <HEAD>

```
<!DOCTYPE HTML>
<html lang="pl">
  <head>
    <meta name="description" content="Strona podsumowanie wiadomości z zakresu HTML">
    <meta name="keywords" content="programownie, HTML, podstawy">
    <meta name="author" content="Marcin Janowski TEB Technikum">
    <meta name="robots" content="noindex,nofollow">
    <meta http-equiv="X-UA-Compatible" content="IE=edge, chrome=1">
    <meta http-equiv="refresh" content="45">
  </head>
</html>
```

## Przykłady użycia meta wg nazwy atrybutu meta

- „application-name” - określa nazwę aplikacji reprezentowanej przez stronę
- „author” - określa autora dokumentu
- „description” - określa krótki opis zawartości dokumentu (max. 170 znaków)
- „keywords” - określa oddzielane przecinkami słowa i frazy kluczowe opisujące zawartość strony ( max. kilkanaście słów )
- „http-equiv” - określa treść nagłówka odpowiedzi HTTP dokumentu np. Referer, Refresh, Cache-control
- „robots” – prośba do robotów o indeksowanie / nie indeksowanie strony





# HTML – STRUKTURA DOKUMENTU - <HEAD>

```
<!DOCTYPE HTML>
<html lang="pl">
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
  </head>
</html>
```



Naszą stronę internetową mogą przeglądać użytkownicy na różnych urządzeniach, od ekranów telefonów po ekrany komputerów. Ponadto ekrany telefonów komórkowych i komputerów mogą się różnić w zależności od marki i wersji.

**Aby nasza strona była responsywna na wszystkie ekrany, ustawiamy „viewport” (rzutnię).**  
**Powyższe ustawienie sprawi aby strona była responsywna czyli dostosowywała się do ekranu urządzenia użytkownika.**

Viewport to zdefiniowany obszar widoczny dla użytkownika na urządzeniu.  
Ustawiamy go za pomocą atrybutu. Przeglądarki mogą ignorować ten znacznik.





# HTML – STRUKTURA DOKUMENTU - <HEAD>

```
<!DOCTYPE HTML>
<html lang="pl">
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
  </head>
</html>
```



Możliwe wartości atrybutu viewport w dokumentacji





# HTML – STRUKTURA DOKUMENTU - <HEAD>

```
<html lang="pl">
  <head>
    <link rel="alternate" href="https://example.com" hreflang="en-gb" />
    <link rel="stylesheet" href="assets/styles/style.css" media="all">
    <link rel="stylesheet" href="assets/styles/style.css" media="print">
    <!-- Przykład ładowanie czcionki ze źródła zewnętrznego: https://fonts.google.com/ -->
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Roboto:ital,wght@0,100;0,300;0,400;0,500;0,700;0,900;1,100;1,300;1,400;1,500;1,700;1,900&display=swap" rel="stylesheet">
  </head>
</html>
```

Element HTML <link> służy do łączenia dokumentu z innymi dokumentami lub zasobami. Najczęściej elementu link używa się do dołączania („includowania”) arkuszy stylów. Zasób, z którym dokument jest połączony za pomocą elementu link określa się w atrybucie href.

Dla <link> można określić także atrybuty szczegóły w dokumentacji 





# HTML – STRUKTURA DOKUMENTU - <HEAD>

```
<!DOCTYPE HTML>
<html lang="pl">
  <head>
    <script type="text/javascript" src="/asstes/scripts/script.js"></script>
    <script type="text/javascript">
      console.log('Ładuję!');
      alert('Skrypt załadowany!');
    </script>
    <!--
    <script src="" async></script> - asynchronicznie jak szybko tylko możliwe
    <script src="" defer></script> - po załadaowaniu wszystkich elementów strony
    -->
  </head>
</html>
```

Element `<script>` służy do wstawiania do dokumentów HTML skryptów napisanych w językach programowania działających po stronie klienta. Skrypt może znajdować się zarówno bezpośrednio w treści tego elementu lub być dołączany z zewnętrznego pliku za pomocą atrybutu `src`.

Znacznik `<script>` może być także wywoływany w sekcji `<body>`.



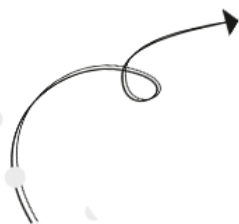


# HTML – STRUKTURA DOKUMENTU – <BODY>

```
<!DOCTYPE HTML>
<html lang="pl">
  <head>
    ...
  </head>
  <body>
    ...
  </body>
</html>
```

Element <body> służy do wstawiania do dokumentów HTML tzw. contentu czyli zawartości witryny. Zawartość tą można dodawać z wykorzystaniem innych znaczników.

W tej sekcji poznamy podstawowe znaczniki HTML.  
Zacniemy jednak od ich podziału na elementy (znaczniki) blokowe i liniowe.







# HTML – KOMPONENTY, ZNACZNIKI – SKŁADNIA RAZ JESZCZE 😊

## Znacznik pojedynczy

znacznik otwierający  znacznik zamykający

nazwa znacznika atrybut

## Znacznik podwójny

znacznik otwierający <h1 class="note">Tekst nagłówka</h1> znacznik zamykający

nazwa znacznika atrybut Treść modyfikowana przez znacznik

<https://e-pasje.pl/html-podstawy-co-to-jest-kod-html/>





# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

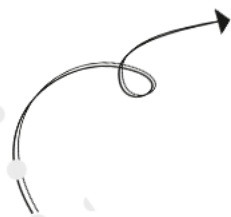
<!-- -->



## Oznaczenie komentarza i/lub bloku instrukcji w HTML

Element wewnątrz znacznika **nie jest widoczny dla użytkownika** lecz **jest widoczny w kodzie strony!**

```
<div id="moje_id" class="klasa_div">  
  <!-- Ten wpis zobaczycie tylko w kodzie strony! -->  
  <p>To jest paragraf, który został uporządkowany w strukturze strony przez znacznik div</p>  
</div>
```





# HTML – STRUKTURA DOKUMENTU – ATRYBUTY DO ZADAŃ SPECJALNYCH ;)



```
<div id="id_znacznika" class="klasa_znacznika">
```

**id** ="id\_znacznika"

Atrybut identyfikatora znacznika, który pozwala odróżnić elementy html i odnosić się do nich w sposób bezpośredni.

**Jedno id służy do oznaczania wyłącznie jednego elementu w zakresie całej wyświetlanej strony tzn. nie może się powtarzać**

**class** ="klasa\_znacznika"

Atrybut klasy znacznika. Podobnie jak atrybutem id można nim rozróżniać poszczególne elementy html.


Stosuje się go w wypadku gdy chcemy oznaczyć pewną właściwość lub grupę / typ elementów html czyli np. nadać taki sam wygląd dla grupy elementów html. Czyli ten sam styl dla kilku <div>-ów lub <p>-paragrafów.

**Atrybut class o tej samej wartości może wielokrotnie pojawiać się w ramach wyświetlanej strony.**



# HTML – STRUKTURA DOKUMENTU – ATRYBUTY GLOBALNE

Jeżeli zastanawiasz się czy możesz użyć danego atrybutu i jego wartości dla danego znacznika zawsze warto zajrzeć do dokumentacji.

Pamiętaj, że każdy ze znaczników może posiadać atrybuty do użycia jedynie z nim. Jednak istnieje pula atrybutów globalnych, których można użyć z każdym znacznikiem HTML. 

The global attributes are attributes that can be used with all HTML elements.

Attribute	Description
<u>accesskey</u>	Specifies a shortcut key to activate/focus an element
<u>class</u>	Specifies one or more classnames for an element (refers to a class in a style sheet)
<u>contenteditable</u>	Specifies whether the content of an element is editable or not
<u>data-*</u>	Used to store custom data private to the page or application
<u>dir</u>	Specifies the text direction for the content in an element
<u>draggable</u>	Specifies whether an element is draggable or not
<u>hidden</u>	Specifies that an element is not yet, or is no longer, relevant
<u>id</u>	Specifies a unique id for an element
<u>lang</u>	Specifies the language of the element's content
<u>spellcheck</u>	Specifies whether the element is to have its spelling and grammar checked or not
<u>style</u>	Specifies an inline CSS style for an element
<u>tabindex</u>	Specifies the tabbing order of an element
<u>title</u>	Specifies extra information about an element
<u>translate</u>	Specifies whether the content of an element should be translated or not





# HTML – STRUKTURA DOKUMENTU – CZAS NA DZIAŁANIE

Otwórz plik wskazany przez nauczyciela i stosuj się do jego poleceń 😊





# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

`<div></div>`



Element **<div>** jest ogólnym kontenerem zawartości przepływu. Nie ma to wpływu na treść ani układ, dopóki nie zostanie w jakiś sposób stylizowany za pomocą CSS.

Jako „czysty” kontener element `<div>` z natury niczego nie reprezentuje. Zamiast tego **służy do grupowania treści**, dzięki czemu można ją łatwo stylizować za pomocą atrybutów **class** lub **id**

**Element `<div>` powinien być używany tylko wtedy, gdy żaden inny element semantyczny (taki jak `<article>` lub `<nav>`) nie jest odpowiedni.**

```
<div id="moje_id" class="klasa_div">
  <!-- Ten wpis zobaczycie tylko w kodzie strony! Kliknij ctrl+u w przeglądarce aby zobaczyć źródło strony -->
  <p>To jest paragraf</p>
  <p>To jest pierwsza linia tekstu<br>A tu zaczyna się nowa linia tekstu.</p>
</div>
```



# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

**<p></p>**

Znacznik opisujący akapit tekstu. Ułatwia dzielenie dużych tekstów na akapity.

```
<p>To jest paragraf</p>
```

**<br>**

Znacznik oznaczający zakończenie linii, wstawienie zakończenia linii. Po nim treść rozpocznie się o wiersz niżej.

**Nie używamy zamiennie do wydzielenia akapitów!**

```
<p>To jest pierwsza linia tekstu<br>A tu zaczyna się nowa linia tekstu.</p>
```

**<hr>**

Znacznik wprowadzający linię horyzontalną. Oznacza rozdział tematyczny treści.

```
<hr>
```



# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

## **<span></span>**

Znacznik ten jest wbudowanym kontenerem używanym do oznaczania części tekstu lub części dokumentu.

Tag <span> można łatwo stylizować za pomocą CSS lub manipulować za pomocą JavaScript przy użyciu atrybutu class lub id.

Znacznik <span> jest bardzo podobny do elementu <div>, ale <div> jest elementem blokowym, a <span> elementem liniowym.

Staraj się nie używać bezpośrednio styli w samym znaczniku tj. za pomocą atrybutu style. Zamiast tego użyj CSS!

```
<style>
  .red-text {
    color: ■ red;
  }
</style>
<p>To jest pierwsza linia tekstu <span class="red-text" title="Trafiony myszą!">z czerwoną wstawką i tytułem po
najejchaniu myszą</span> i ponownie czarny.</p>
```

## **<b></b>**

Znacznik pogrubiający część tekstu

```
<p><b>Tekst pogrubiony</b> i powrót do standardowego.</p>
```





# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

**`<strong></strong>`**

Znacznik pogrubiający część tekstu. Lecz tym znacznikiem powinno zaznaczać się ważny semantycznie tekst.

```
<p><strong>Tekst pogrubiony o istotnym znaczeniu</strong> i powrót do standardowego.</p>
```

**`<h1></h1> ... <h6></h6>`**

Znaczniki wyróżniające nagłówki. Stopniowane od h1 najważniejszy do h6 o najmniejszej wadze.

Nagłówek h1 powinien wystąpić na stronie wyłącznie 1 raz. Pozostałe nagłówki zależnie od potrzeb w sposób umiarkowany!

Kwestia poprawnego użycia nagłówków zostanie omówiona w dziale semantyki.

```
<h1>Nagłówek h1</h1>
<h2>Nagłówek h2</h2>
<h3>Nagłówek h3</h3>
<h4>Nagłówek h4</h4>
<h5>Nagłówek h5</h5>
<h6>Nagłówek h6</h6>
```



# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

**<u></u>**

Znacznik opisujący fragment tekstu podkreślonego.

```
<p>Lorem <u>tekst podkreślony</u> consectetur, adipisicing elit. Minima, tempore!</p>
```

**<i></i>**

Znacznik opisujący fragment tekstu pochylonego

```
<p>Lorem ipsum dolor <i>tekst pochylony</i>, adipisicing elit. Minima, tempore!</p>
```

**<em></em>**

Znacznik wprowadzający linię horyzontalną. Oznacza rozdział tematyczny treści.

```
<p>Lorem <em>tekst o szczególnym znaczeniu</em>, adipisicing elit. Minima, tempore!</p>
```



# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

## **<sup></sup>**

Znacznik opisujący fragment tekstu, który ma być wyświetlony w indeksie górnym np. potęgi.

```
<p>Lorem <sup>index wysoki</sup> stosowany np. w potęgach 2<sup>-3</sup> elit. Minima, tempore!</p>
```

## **<sub></sub>**

Znacznik opisujący fragment tekstu, który ma być wyświetlony w indeksie dolnym np. w zapisie logarytmu.

```
<p>Lorem <sub>index niski</sub> stosowany np. w logarytmach log<sub>2</sub>16 = x</p>
```

Znaczniki <sub>,<sup> (podobnie jak późniejszy <q> czy <small>) mogą być zagnieżdżane jeden w drugim.

```
<p>Przykład indeksu wysokiego zagnieżdżonego 10<sup>e<sup>x</sup></sup></p>
```



# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

## **<del></del>**

Znacznik opisujący fragment tekstu usuniętego (deleted).

```
<p>Lorem <del>usunięcie</del> adipisicing elit. Minima, tempore!</p>
```

## **<ins></ins>**

Znacznik opisujący fragment tekstu wstawionego (inserted)

```
<p>Lorem <ins>tekst wstawiony</ins> adipisicing elit. Minima, tempore!</p>
```

## **<s></s>**

Znacznik oznaczający tekst przekreślony (stroke)

```
<p>Lorem <s>Tekst przekreślony</s> adipisicing elit. Minima, tempore!</p>
```



# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

## `<q></q>`

Znacznik opisujący krótki cytat (w ciągu tekstowym), element inline. Można dodać w nim atrybut cite z linkiem, skąd pochodzi cytat. Znacznik może być zagnieżdżany.

```
<p>Zacytujmy klasyka liniowo .. <q cite="https://founders.archives.gov/documents/Franklin/01-17-02-0140">(…) Na tym świecie pewne są tylko śmierć i podatki</q> - Benjamin Franklin.</p>
```

## `<blockquote></blockquote>`

Znacznik opisujący krótki cytat długi. Występuje jako element domyślnie blokowy. Domyślnie nie wstawia nam znaków cudzysłowu.

```
<p>Zacytujmy klasyka blokowo ..</p>
<blockquote cite="https://founders.archives.gov/documents/Franklin/01-17-02-0140">(…) Na tym świecie pewne są
tylko śmierć i podatki
  <footer>
    Benjamin Franklin
  </footer>
</blockquote>
```



# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

## **<small></small>**

Znacznik opisujący fragment tekstu pomniejszonego np. opisu dla \*, małego druczku 😊

```
<p>Przykład <small>tekstu pomniejszonego</small></p>
```

## **<mark></mark>**

Znacznik oznaczający tekst podbiciem.

```
<p>Przykład tekstu <mark>zaznaczonego</mark></p>
```

## **<cite></cite>**

Znacznik wykorzystywany do oznaczania np. tytułów.

```
<p><cite>Przykład użycia tagu cite</cite> często używanego do określania tytułu prac np. książek, filmów etc. </p>
```



# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

## `<pre></pre>`

Treść opisana znacznikiem `<pre>` jest wyświetlana ze stałą szerokością czcionki, z uwzględnieniem spacji czy „enterów”. W założeniu tekst ma być wypisany tak jak został wpisany w HTML.

```
<pre>
    * {
        font-family: "Comic Sans MS", "Comic Sans", "Marker Felt" !important;
    }
</pre>
```

I masę innych przydatnych znaczników inline 😊

```
<div>
    <p>The CSS background-color property defines the background color of an element.</p>
    <p><dfn>HTML</dfn> is the standard markup language for creating web pages.</p>
    <p>keyboard input: <code>control + space</code> &lt;code></p>
    <p>meter: <code><meter value="423" min="200" max="500">awesomeness</meter> &lt;code></p>
    <p>progress: <code><progress value="90" max="100">70 %</progress> &lt;code></p>
    <p>sample text: <code><samp>Error</samp> &lt;code></p>
    <p>time: <code><time datetime="1985-12-09 19:00">Dec 9</time> &lt;code></p>
    <p>variable: <code><var>cat</var> + <var>dog</var> = <var>catdog</var> &lt;code></p>
</div>
```



# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

`<a></a>` 

Znacznik opisujący odnośnik / link. Przy czym **linkiem/odnośnikiem jest jedynie gdy ma zdefiniowany atrybut href**, który nie jest obowiązkowy. W innym przypadku jest jedynie „kotwicą” i jako taką czasami jest używany. To w wartości atrybutu href wskazujemy gdzie odnośnik ma nas zaprowadzić.

```
<a href="https://onet.pl">Kliknij aby przejść na stronę onet.pl</a>
<a href="postrona.html">Kliknij aby przejść na podstronę podstrona.html</a>
```

Znacznik ten może posiadać atrybut **target**, który określa cel linku, czyli inaczej gdzie docelowo w przeglądarce ma zostać otwarty linkowany dokument. Może on przyjmować wartości:

- **target="\_blank"** – który oznacza otwórz w nowym oknie / karcie przeglądarki,
- **target="\_self"** - otwórz w karcie, w której jest wczytany obecny dokument,
- **target="\_top"**, **target="\_parent"** – odnoszące się do ramek, archaizmy, nie będziemy ich używać.

```
<p><a href="https://onet.pl" target="_self">Kliknij aby przejść na stronę onet.pl</a></p>
<p><a href="https://onet.pl" target="_blank">Kliknij aby przejść na stronę onet.pl</a> otwierając nową kartę
przeglądarki.</p>
```





# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

## `<ul></ul>`

Znacznik listy nieuporządkowanej (nienumerowanej). Znacznik ten to jedynie definicja listy (otwarcie, zamknięcie).

Faktyczne elementy listy zamknięte są w znacznikach `<li></li>`

```
<ul>
  <li>Start</li>
  <li>Znaczniki</li>
  <li>Bloki</li>
</ul>
```

- Start
- Znaczniki
- Bloki

## `<ol></ol>`

Znacznik listy uporządkowanej (numerowanej). Podobnie jak wyżej znacznik ten to jedynie definicja listy (otwarcie, zamknięcie).

Faktyczne elementy listy zamknięte są w znacznikach `<li></li>`

```
<ol start="1">
  <li>Start</li>
  <li>Znaczniki</li>
  <li value="3">Bloki</li>
</ol>
```

1. Start
2. Znaczniki
3. Bloki



# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

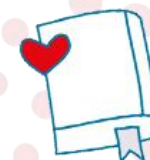
`<ol></ol>`

Wyłącznie dla listy uporządkowanej możemy dodać atrybuty:

- dla znacznika `<ol>` **start="numer"** – który oznacza od jakiego numeru ma startować pierwszy element `<li>`
- dla znacznika `<li>` **value="numer"** - który oznacza numer konkretnego elementu wypunktowania (`<li>`)

```
<ol start="1">  
  <li>Start</li>  
  <li>Znaczniki</li>  
  <li value="3">Bloki</li>  
</ol>
```

20. Start  
21. Znaczniki  
100. Bloki





# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

`<ol></ol>` `<ul></ul>`

Listy mają wiele zastosowań, od pierwotnych po elementy nawigacyjne czy elementy pseudo pola select. Dają duże możliwości jeżeli chodzi o wygląd i stylowanie w CSS.

**Listy można także dowolnie zagnieżdżać, tworząc listy wielopoziomowe.**

Dodatkowo możemy mieszać je typami (uporządkowana/nieuporządkowana) zachowując jeden typ dla konkretnego szczebla.

```
<ol>
  <li>Start</li>
  <li>Znaczniki</li>
  <li>Bloki</li>
  <li>Typografia i stylowanie</li>
  <li>Formularze
    <ol start="10">
      <li>podlink 1</li>
      <li>podlink 2</li>
      <li>podlink 3
        <ul>
          <li>podlink 1</li>
          <li>podlink 2</li>
          <li>podlink 3</li>
        </ul>
      </li>
    </ol>
  </li>
  <li>Tabele</li>
  <li>Obrazy</li>
  <li>Multimedia</li>
</ol>
```

1. Start
2. Znaczniki
3. Bloki
4. Typografia i stylowanie
5. Formularze
  10. podlink 1
  11. podlink 2
  12. podlink 3
    - podlink 1
    - podlink 2
    - podlink 3
6. Tabele
7. Obrazy
8. Multimedia



# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

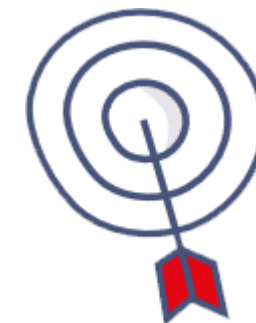
`<dl></dl>`

Ostatnim typem list jest lista definicji. Jest to lista opisowa składająca się z par elementów:

- definicja `<dt></dt>`
- opis definicji `<dd></dd>`

```
<dl>  
  <dt>Coffee</dt>  
  <dd>Black hot drink</dd>  
  <dt>Milk</dt>  
  <dd>White cold drink</dd>  
</dl>
```

Coffee	Black hot drink
Milk	White cold drink





# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

## `<table></table>`



Znacznik opisujący tabelę.

Tabela oczywiście składa się z wierszy i komórek odpowiednio oznaczonych tagami :

- `<tr></tr>` - definiujący wiersz tabeli w ramach którego występują komórki:
  - `<td></td>` - definiujący komórkę tabeli (zwykłą)
  - `<th></th>` - definiujący komórkę tabeli (nagłówkową), stosowaną do opisu wartości w innych komórkach

Zatem najprostszą tabelę możemy zdefiniować w następujący sposób

```
<table>
  <tr> <!-- wiersz -->
    <td>Komórka 1</td> <!-- pojedyncza komórka -->
    <td>Komórka 2</td>
    <td>Komórka 3</td>
  </tr>
  <tr>
    <td>Komórka 1 wiersz 2</td>
    <td>Komórka 2 wiersz 2</td>
    <td>Komórka 3 wiersz 2</td>
  </tr>
</table>
```

## Bez stylowania

Komórka 1	Komórka 2	Komórka 3
Komórka 1 wiersz 2	Komórka 2 wiersz 2	Komórka 3 wiersz 2

## Z użyciem stylu (patrz `<style>` u góry dokumentu)

Komórka 1	Komórka 2	Komórka 3
Komórka 1 wiersz 2	Komórka 2 wiersz 2	Komórka 3 wiersz 2



# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

Tabele możemy rozbudować o dodatkowe znaczniki:

**<caption></caption>** - definiujący tytuł tabeli, umieszczany bezpośrednio po otwarciu tagu <table>

**<thead></thead>** - definiujący sekcję nagłówkową tabeli w ramach którego występują komórki **<tr><th></th>..</tr>**

**<tbody></tbody>** - definiujący ciało tabeli tzw. „mięso” 😊, czyli zbiór wierszy i komórek zawierających dane właściwe  
**<tr><td></td>..</tr>**

**<tfoot></tfoot>** - definiujący sekcję stopki tabeli, najczęściej jakieś podsumowanie, element rzadko używany.

Używanie powyższych znaczników jest pożądane, głównie z uwagi na porządek i semantykę, czyli znacznie poszczególnych elementów.

inne [<colgroup>](#).





# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

```
<table>
  <caption>Wyniki najbliższego kolokwium :)</caption>
  <thead>
    <tr>
      <th>Lp</th>
      <th>Imię</th>
      <th>Wynik</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Marcin</td>
      <td>5+</td>
    </tr>
    <tr>
      <td>1</td>
      <td>Wojtek</td>
      <td>5+</td>
    </tr>
    <tr>
      <td>1</td>
      <td>Kacper</td>
      <td>5+</td>
    </tr>
    <tr>
      <td>1</td>
      <td>Kacper</td>
      <td>5+</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td></td>
      <td>Średnia</td>
      <td>5+</td>
    </tr>
  </tfoot>
</table>
```

Przykład tabeli z kompletem znaczników

Wyniki najbliższego kolokwium :)

Lp	Imię	Wynik
1	Marcin	5+
1	Wojtek	5+
1	Kacper	5+
	Średnia	5+





# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

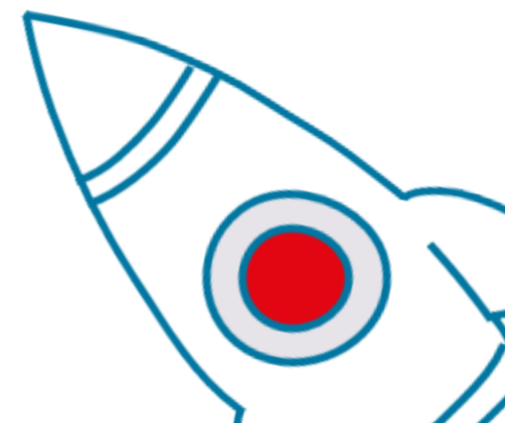
Kolejnym ważnym zagadnieniem jest możliwość łączenia (scalania) komórek tabeli zarówno wierszowo jak i kolumnowo.

Do tego celu służą 2 atrybuty komórki tabeli:

→ **rowspan** = "ilość\_wierszy\_do\_scalenia" – jeżeli łączymy komórki w zakresie wierszy

```
<!-- rowspan -->
<table>
  <tr>
    <td rowspan="2">1</td>
    <td>2</td>
    <td>3</td>
  </tr>
  <tr>
    <td>5</td>
    <td>6</td>
  </tr>
  <tr>
    <td>7</td>
    <td>8</td>
    <td>9</td>
  </tr>
</table>
```

1	2	3
	5	6
7	8	9





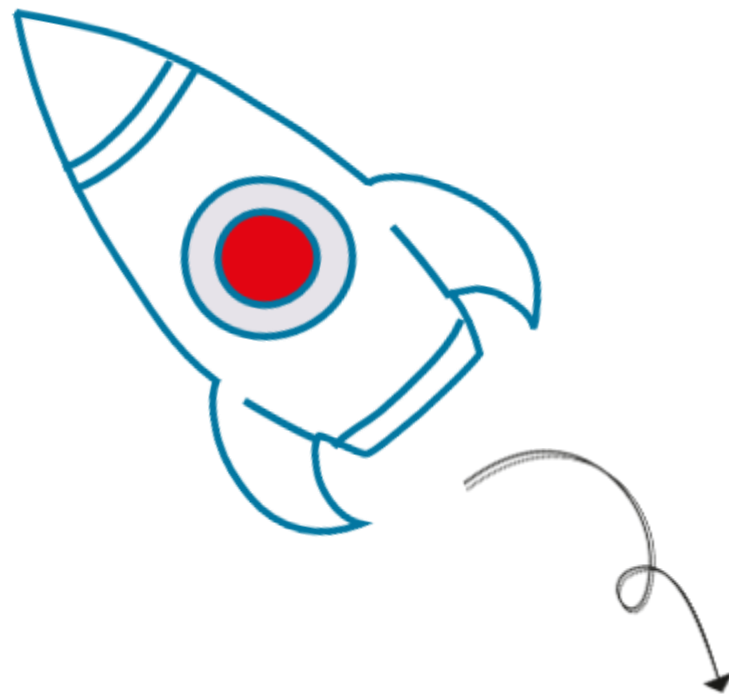


# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

→ **colspan** = "ilość\_kolumn\_do\_scalenia" - jeżeli łączymy komórki w zakresie kolumn

```
<!-- colspan -->
<table>
  <tr>
    <td colspan="2">1</td>
    <td>3</td>
  </tr>
  <tr>
    <td>4</td>
    <td>5</td>
    <td>6</td>
  </tr>
  <tr>
    <td>7</td>
    <td>8</td>
    <td>9</td>
  </tr>
</table>
```

1		3
4	5	6
7	8	9





```
<table width="40%">
  <caption>Wyniki drużynowego konkursu skoku w dal</caption>
  <thead>
    <tr>
      <th rowspan="2">Drużyna</th>
      <th rowspan="2">Imię i nazwisko członka drużyny</th>
      <th colspan="2">Wyniki</th>
    </tr>
    <tr>
      <th>cząstkowy</th>
      <th>średnia</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td rowspan="2">1</td>
      <td>Marcin Janowski</td>
      <td>3.5m</td>
      <td rowspan="2">3m</td>
    </tr>
    <tr>
      <td>Wojtek Wojtek</td>
      <td>2.5m</td>
    </tr>
    <tr>
      <td rowspan="2">2</td>
      <td>Kacper Wioślarz</td>
      <td>1.5m</td>
      <td rowspan="2">1.5m</td>
    </tr>
    <tr>
      <td>Wojtek Krzesiwo</td>
      <td>1.5m</td>
    </tr>
    <tr>
      <td colspan="3">Średni wynik na drużynę</td>
      <td>2.25m</td>
    </tr>
  </tbody>
</table>
```

Wyniki drużynowego konkursu skoku w dal

Drużyna	Imię i nazwisko członka drużyny	Wyniki	
		cząstkowy	średnia
1	Marcin Janowski	3.5m	3m
	Wojtek Wojtek	2.5m	
2	Kacper Wioślarz	1.5m	1.5m
	Wojtek Krzesiwo	1.5m	
Średni wynik na drużynę			2.25m



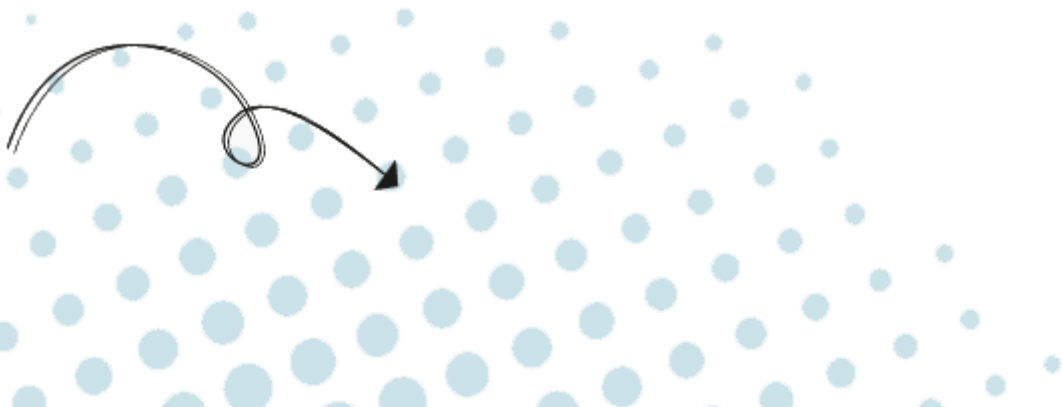
# HTML – STRUKTURA DOKUMENTU – ZADANIE DO WYKONANIA

## Zadanie domowe

Korzystając z poznanych znaczników dla tabeli i ich atrybutów, znacznika small, oraz b, stwórz kod poniższej tabeli.

Remanent sklepu "Wujka Czesława"

Lp	Nazwa towaru	Symbol	Ilość (szt.)	Cena			Wartość	
				netto (zł)	vat (%)	brutto (zł)	netto (zł)	brutto (zł)
1	Trampki Adibos	TAdibo	3	10.00	23%	12.30	30.00	36.90
1	Trampki Lukasty	TLukas	2	10.00	23%	12.30	20.00	24.60
Suma wartości towarów							50.00	61.50





# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

## **<img>**

Znacznik służący do osadzania obrazów w dokumencie HTML.

Znacznik img jest pojedynczy, a nie podwójny, gdyż obraz jest obiektem - nie trzeba określać gdzie się zaczyna, a gdzie kończy, ponieważ o tym decyduje rozmiar źródłowej grafiki lub ewentualnie określone przez nas właściwości CSS ew. atrybut width i height.

Znacznik posiada 2 wymagane atrybuty

- **src** = "link\_do\_źródła\_obrazu" – określający ścieżkę dostępu do źródłowej grafiki
- **alt** = "tekst\_alternatywny" - określający tekstowy opis opisujący czym jest (co przedstawia) wstawiany obraz. Tekst ten wyświetlany jest jedynie wtedy, gdy grafika nie może zostać wczytana np. źródło podane w src nie istnieje.

```

```



# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

Dla znacznika obrazu możemy użyć także atrybutu width oraz height do określenia odpowiednio szerokości i wysokości obrazu.

Nie jest to jednak zalecane, gdy nie jest konieczne. Jeżeli już musisz określić rozmiar obszaru zrób to elementem otaczającym i/lub skorzystaj z CSS!

```
<!-- przykład użycia width / height-->  

```

Formaty obrazów jakie możemy osadzić za pomocą img to:

- [APNG \(Animated Portable Network Graphics\)](#) - animacje
- [AVIF \(AV1 Image File Format\)](#) – animacje , zdjęcia
- [GIF \(Graphics Interchange Format\)](#) – proste animacje o obniżonej jakości
- [JPEG \(Joint Photographic Expert Group image\)](#) – jeden z najbardziej znanych formatów grafiki rastrowej, nieruchomych obrazów, daje się dobrze (stratnie jakościowo) kompresować
- [PNG \(Portable Network Graphics\)](#) – obrazy obsługujące przeźroczystość, dobry wybór dla bezstratnej kompresji zdjęć
- [SVG \(Scalable Vector Graphics\)](#) - Format obrazu wektorowego. Używaj w przypadku obrazów, które muszą być narysowane dokładnie w różnych rozmiarach np. logotyp.
- [WebP \(Web Picture format\)](#) – najlepszy format (wybór) dla animowanych i nieanimowanych obrazów, pozwala znacznie obniżyć wielkość plików graficznych (kompresja stratna lub bezstratna konwertowanych obrazów)

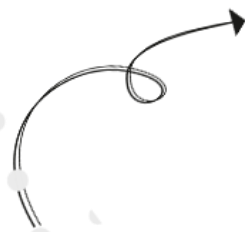




# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

## Pamiętaj!

- zawsze zmniejszaj rozmiary grafik przynajmniej do potencjalnie najmniejszego największego rozmiaru! Lub stosuj różne grafiki w zależności od wielkości ekranu urządzenia, na którym wyświetlana jest strona,
- poddawaj używane pliki kompresji np. z wykorzystaniem [tinypng.com](https://tinypng.com),
- dostosuj format pliku do potrzeb i rodzaju prezentowanej treści,
- optymalizuj, optymalizuj, optymalizuj, obrazy to najczęściej najwięcej „ważąca” część serwisu internetowego.





# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

## `<figure></figure>`

Jeżeli chcesz nadać swojemu obrazowi znaczenia opisz go tagiem `<figure>` oraz `<figcaption>` ma to znaczenie nie tylko semantyczne, ale i użytkowe.

- `<img>` - tag obrazu użyj jak zwykle
- `<figcaption>` - użyj aby dodać opis zawartości zdjęcia

```
<!-- figure jest elementem blokowym! -->
<figure>
  
  <figcaption>Mount Blanc pokryta śniegiem odbijającym się w wodzie wieczorem w chamonix we francji.</figcaption>
</figure>
```



# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

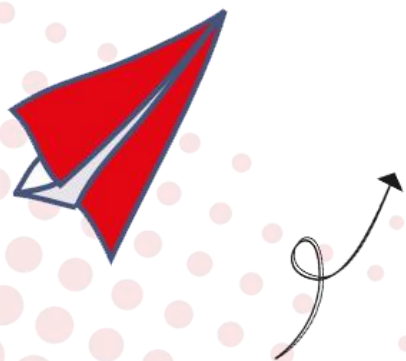
**<picture></picture>**

Innym sposobem osadzania obrazów w serwisach WWW, coraz częściej stosowanym, jest użycie tagu <picture>, głównie z uwagi na elastyczność w określaniu źródła zasobu obrazu.

Tag ten bowiem umożliwia podanie kilku źródeł obrazu dla różnych rozdzielczości ekranu.

Szczegóły tagu media poznamy podczas zajęć z CSS.

Zatem najczęstszym zastosowaniem elementu <picture> będzie kierowanie grafiką w responsywnych projektach. Zamiast jednego obrazu, który jest skalowany w górę lub w dół w zależności od szerokości widocznego obszaru, można zaprojektować wiele obrazów, aby ładniej wypełniały widoczny obszar przeglądarki.







# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

```
<picture class="full-width">
  <source media="(min-width:1500px)" srcset="assets/images/polska-kula-ziemiska-2023.jpg">
  <source media="(min-width:1024px)" srcset="assets/images/polska-kula-ziemiska-2023.png">
  <source media="(min-width:680px)" srcset="assets/images/polska-kula-ziemiska-2023_680.png">
  <source media="(min-width:465px)" srcset="assets/images/polska-kula-ziemiska-2023_480.png">
  
</picture>
```

Element **<picture>** zawiera dwa znaczniki: jeden lub więcej znaczników **<source>** i jeden znacznik **<img>**. Przeglądarka wyszuka pierwszy element **<source>**, w którym zapytanie o media pasuje do aktualnej szerokości rzutni (viewport), a następnie wyświetli odpowiedni obraz (określony w atrybucie **srcset**). Element **<img>** jest wymagany jako ostatni element potomny elementu **<picture>**, jako opcja awaryjna, jeśli żaden ze znaczników źródłowych nie pasuje.



# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

**<svg></svg>**

Znacznik definiujący kontener grafiki SVG. W nim możemy „rysować” kształty za pomocą kodu (XML)



```
<svg width="100" height="100">  
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />  
</svg>
```

Aby osadzić gotowy plik SVG otwórz go za pomocą edytora np. Sublime, skopuj kod <svg>.... </svg> (pomiń nagłówek <xml > i wklej do Twojego pliku HTML.





# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI



**<video></video>**

Znacznik <video> służy do osadzania treści wideo w dokumencie, na przykład klipu filmowego lub innych strumieni wideo.

```
<video width="320" height="240" controls poster="assets/images/polska-kula-ziemski-2023_480.png">  
  <source src="assets/videos/file_example_OGG_640_2_7mg.ogg" type="video/ogg">  
  <source src="assets/videos/magic.mp4" type="video/mp4">  
  Wygląda na to, że twoja przeglądarka nie obsługuje plików wideo.  
</video>
```

Tag <video> zawiera jeden lub więcej tagów <source> z różnymi źródłami wideo. Przeglądarka wybierze pierwsze obsługiwane źródło.

Tekst między tagami <video> i </video> będzie wyświetlany tylko w przeglądarkach, które nie obsługują elementu <video>. Istnieją trzy obsługiwane formaty wideo w HTML: MP4, WebM i OGG.





# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI



**<audio></audio>**

Znacznik <audio> służy do osadzania treści audio w dokumencie, na przykład muzyki lub innych strumieni dźwiękowych.

```
<audio controls>
  <source src="assets/sounds/mixkit-fast-rocket-whoosh-1714.wav" type="audio/wav">
  <source src="assets/sounds/deep-house-pluck-25180.mp3" type="audio/mp3">
  Wygląda na to, że twoja przeglądarka nie obsługuje plików audio.
</audio>
```

Znacznik <audio> zawiera jeden lub więcej znaczników <source> z różnymi źródłami dźwięku. Przeglądarka wybierze pierwsze obsługiwane źródło (liczy się kolejność).

Tekst między tagami <audio> i </audio> będzie wyświetlany tylko w przeglądarkach, które nie obsługują elementu <audio>.

Istnieją trzy obsługiwane formaty audio w HTML: MP3, WAV i OGG.





# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

`<object></object>`



Znacznik służący do osadzania obiektów zewnętrznych. Domyślnie powstał do ładowania pluginów wyszukiwarek. Na siłę można nim osadzić obraz, video, audio, lecz mamy do tego semantyczne i dedykowane znaczniki `<img>`, `<video>`, `<audio>`.

Co więc osadzić za pomocą tego znacznika? Np. obiekt zewnętrzny taki jak plik PDF nie w formie linku, a jako dokument.

```
<object type="application/pdf" data="/assets/other/Poradnik_ABC_cyberbezpieczenstwa.pdf" width="800" height="500"></object>
```

Aby poprawnie osadzić obiekt dobrze jest określić jego wysokość, szerokość oraz typ danych (choć ten często przeglądarki potrafią odczytać samodzielnie). Należy także wskazać dane obiektu określając parametr **data**.



# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

**<embed/>**



To również znacznik służący do osadzania obiektów zewnętrznych.

Na siłę można nim osadzić obraz, video, audio, lecz mamy do tego semantyczne i dedykowane znaczniki <img>, <video>, <audio>

Co więc osadzić za pomocą tego znacznika? Np. obiekt zewnętrzny taki jak plik PDF nie w formie linku, a jako dokument.

```
<embed src="/assets/other/Poradnik_ABC_cyberbezpieczenstwa.pdf" width="400" height="300">
```

Aby poprawnie osadzić element zewnętrzny należy określić opcjonalnie jego: wysokość, szerokość oraz typ danych (choć ten często przeglądarki potrafią odczytać samodzielnie). Należy także obowiązkowo określić źródło elementu osadzanego przez atrybut **src**.

Zarówno <object> jak i <embed> nie są zalecane do osadzania plików HTML, do tego lepiej użyć....





# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI

## <iframe></iframe>



Znacznik określa ramkę w wierszu (inline).

Ramka wbudowana służy do osadzania innego dokumentu w bieżącym dokumencie HTML (niezależnie).

Jest to najlepszy znacznik do osadzania jednego dokumentu w drugim.

Dlaczego? Posiada najwięcej opcji w formie atrybutów i daje się dobrze stylować za pomocą CSS.

```
<!-- przykład pliki lokalne -->
<iframe width="800" height="500" src="/index.html" title="Podgląd dokumentu index.html"></iframe>
<iframe width="800" height="500" src="/assets/other/Poradnik_ABC_cyberbezpieczenstwa.pdf" title="Podgląd dokumentu PDF"></iframe>

<!-- przykład ładowania innych stron -->
<iframe width="800" height="500" src="https://teb.edu.pl" title="Podgląd teb.edu.pl" ></iframe>
<iframe width="800" height="500" src="https://www.w3schools.com/" title="Podgląd dokumentu index.html"></iframe>

<!-- przykład ładowania mapy -->
<iframe width="425" height="350" frameborder="0" scrolling="no" marginheight="0" marginwidth="0" src="https://www.openstreetmap.org/export/
embed.html?bbox=18.568428754806522%2C49.94716746725411%2C18.582590818405155%2C49.95367744127047&layer=mapnik" style="border: 1px solid 
black"></iframe><br/><small><a href="https://www.openstreetmap.org/#map=17/49.95042/18.57551">Wyświetl większą mapę</a></small>

<!-- przykład ładowania filmu z YT -->
<iframe width="560" height="315" src="https://www.youtube.com/embed/6nUjeVw0HN8" title="YouTube video player" frameborder="0"
allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share" allowfullscreen></iframe>
```

Wskazówka: dobrą praktyką jest zawsze umieszczanie atrybutu tytułu dla elementu <iframe>. Jest to używane przez czytniki ekranu do odczytywania zawartości elementu <iframe>.



# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI - FORMULARZ

**`<form></form>`**



Znacznik `<form>` definiuje w HTML formularz, a konkretniej opisuje gdzie formularz zaczyna się i kończy. Obszar pomiędzy znacznikami `form` wypełniamy dowolnymi kontrolkami - polami formularza (o tych za chwilę).

```
<form>
  •
  elementy formularza (pola, przyciski)
  •
</form>
```

Formularze służą użytkownikom do wprowadzania danych wejściowych, które następnie są przetwarzane przez serwer czyli np. dodawane do bazy, używane w skrypcie jako parametry np. listy produktów, stosowane do logowania użytkownika itd.






# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI - FORMULARZ



Atrybuty formularza, które powinny być zawsze uzupełnione lecz nie są wymagane:

- **action = "plik\_akcji.php"** – określający ścieżkę do pliku, który będzie przetwarzał dane z formularza, w tym przypadku plik\_akcji.php. Jeżeli pominiesz ten atrybut formularz prześle dane do pliku (ścieżki), w której się znajduje, czyli wyświetlanej podstrony.
- **method = "post"** – określający metodę przetwarzania formularza. Wśród najczęściej używanych metod formularzy są: GET i POST. Domyślną metodą jest GET i to ona właśnie zostanie użyta gdy pominiesz w formularzu określenie tego atrybutu. Metody POST używaj zawsze gdy w formularzu przesyłasz dane „wrażliwe” np. hasła, pesel lub plik. Dane w metodzie POST są przesyłane „niejawnie” i nie są cache-owane w przeglądarce (przy użyciu metody GET jest odwrotnie)

Podstawowe różnice pomiędzy metodami, oraz inne metody 

- **enctype="multipart/form-data"** - określa sposób kodowania danych formularza podczas przesyłania ich na serwer, używany jedynie z metodą POST. Użyj tego kodowania jeżeli przesyłasz dane binarne np. pliki. 

```
<!-- dla metody GET -->  
<form action="plik_akcji.php" method="get">  
<!-- dla metody POST -->  
<form action="plik_akcji.php" method="post" enctype="multipart/form-data">
```





# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI - FORMULARZ



→ Wszystkie atrybuty znacznika 

Attribute	Description
<u>accept-charset</u>	Specifies the character encodings used for form submission
<u>action</u>	Specifies where to send the form-data when a form is submitted
<u>autocomplete</u>	Specifies whether a form should have autocomplete on or off
<u>enctype</u>	Specifies how the form-data should be encoded when submitting it to the server (only for method="post")
<u>method</u>	Specifies the HTTP method to use when sending form-data
<u>name</u>	Specifies the name of the form
<u>novalidate</u>	Specifies that the form should not be validated when submitted
<u>rel</u>	Specifies the relationship between a linked resource and the current document
<u>target</u>	Specifies where to display the response that is received after submitting the form



# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI - FORMULARZ



Możliwe elementy formularza: 

<b>&lt;input&gt;</b>	znacznik pojedynczego (jednoliniowego) pola wsadowego różnego typu (numeryczny, e-mail, text ... itd)
<b>&lt;label&gt;</b>	znacznik etykiety pola, szczególnie przydatny dla użytkowników czytników ekranów, utrzymania porządku pól, oraz osób mających problem w kliknięciu w małe elementy (radiobox, checkbox)
<b>&lt;select&gt;</b>	znacznik określający pole wyboru z listy rozwijanej
<b>&lt;option&gt;</b>	znacznik określający opcje wyboru listy rozwijanej lub listy opcji predefiniowanych
<b>&lt;optgroup&gt;</b>	znacznik określający grupy opcji wyboru listy rozwijanej
<b>&lt;textarea&gt;</b>	znacznik wieloliniowego wsadowego pola tekstowego
<b>&lt;button&gt;</b>	znacznik definiujący klikalny przycisk
<b>&lt;fieldset&gt;</b>	znacznik grupujący pola formularza w logiczny blok
<b>&lt;legend&gt;</b>	znacznik etykiety grupy pól określonego przez <fieldset>
<b>&lt;datalist&gt;</b>	znacznik definiujący listę pól predefiniowanych dla pola zdefiniowanego przez <input>
<b>&lt;output&gt;</b>	znacznik definiujący wynik działania skryptu



# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI - FORMULARZ

`<input/>`



Znacznik pojedynczy `<input>` określa pole wejściowe (jednoliniowe), w którym użytkownik może wprowadzać dane.

Znacznik ten posiada obowiązkowy atrybut **type** określający jakiego rodzaju jest to pole.

Innymi słowy określa jakie dane chcemy od użytkownika dostać w tym danym konkretnym polu formularza.

W HTML5 posiadamy naprawdę spory zasób typów pól, za chwilę przyjrzymy się wybranym z bliska 😊.

`<input type="button">`

`<input type="checkbox">`

`<input type="color">`

`<input type="date">`

`<input type="datetime-local">`

`<input type="email">`

`<input type="file">`

`<input type="hidden">`

`<input type="image">`

`<input type="month">`

`<input type="number">`

`<input type="password">`

`<input type="radio">`

`<input type="range">`

`<input type="reset">`

`<input type="search">`

`<input type="submit">`

`<input type="tel">`

**`<input type="text">`**

`<input type="time">`

`<input type="url">`

`<input type="week">`





# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI - FORMULARZ





`<input/>`



Przyjrzyjmy się jednak innym ważnym atrybutom znacznika input, których poznanie ułatwi nam interpretację dalszych elementów:

- **name = "nazwa\_pola"** – nadając polu atrybut nazwa, określamy również nazwę, którą przyjmie ona w tablicy przestanych danych do pliku akcji. Nazwa nie może zaczynać się od liczby, posiadać spacji oraz znaków specjalnych poza „-” „\_”, „[]”. Dobrym zwyczajem jest nadawanie polom nazw odzwierciedlających ich zawartość czyli np. dla pola zawierającego email name="e-mail",
- **id = "identyfikator"** – czyli atrybut globalny, niezwykle ważny w polach formularzy, warto go nadawać aby móc odwoływać się do konkretnych pól np. nadając im etykiety.
- **value = "wartosc\_atrybutu"** – atrybut ustawiający wartość pola już w momencie jego inicjacji

I bardzo funkcjonalnym 😊, które wejdą w krew w trakcie stosowania praktycznego formularzy:

- **required** – bez wartości, oznaczający dane pole jako wymagane,
- **readonly** – bez wartości, oznaczający pole tylko do odczytu (jego edycja jest niemożliwa) ,
- **disabled** - bez wartości, oznaczający pole wyłączone z możliwości wprowadzenia,
- **placeholder = "opis\_pomocniczy"** - pokazuje opis pomocniczy (podpowiedź) do czasu, aż pole nie zostanie wypełnione,
- atrybut **autocomplete** – przyjmujący wartości **on** lub **off** – odpowiednio pozwalające (lub nie) na autouzupełnianie pola przez przeglądarkę na podstawie zapamiętanych danych
- **pattern = "wyrażenie\_regularne"** – atrybut służący walidacji wprowadzonych danych wg. wymogu (pattern-u)  
- O innych ciekawych atrybutach wspomnimy przy okazji omawiania różnych typów pól.  
Obecnie zapamiętaj te, jako podstawę dla znacznika `<input>` 😊



# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI - FORMULARZ

`<label></label>`



```
<label for="lastname">Nazwisko:</label>  
<input type="text" id="lastname" name="nazwisko" placeholder="Wpisz nazwisko">
```

Znacznik etykiety, który może być zastosowany do kilkunastu elementów html, w tym wybranych elementów formularza.

**Aby przypisać etykietę do konkretnego pola należy ustawić wartość atrybutu for identyczną jak wartość atrybutu id pola, z którym chcemy powiązać etykietę.**





# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI - FORMULARZ

**<select></select>** - znacznik określający pole wyboru z listy rozwijanej



```
<select name="kraj" id="country">
  <option value="pl">Polska</option>
  <option value="es" selected>Hiszpania</option>
  <option value="de">Niemcy</option>
</select>
```

**<option></option>** znacznik definiujący pojedyncze opcje wyboru listy rozwijanej lub listy definicji, które mogą być wybrane.

Domyślnie zaznaczonym elementem na liście rozwijanej jest pierwszy element `<option>`. Jeżeli chcemy domyślnie zaznaczyć inny element należy w odpowiednim elemencie `<option>` wprowadzić atrybut **selected** (bezwartościowy).



# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI - FORMULARZ

`<select></select>` - użyteczne atrybty znacznika select

→ **size** = "ilość widocznych wierszy do wyboru"

```
<select name="kraj" id="country" size="3">
  <option value="pl">Polska</option>
  <option value="es" selected>Hiszpania</option>
  <option value="de">Niemcy</option>
  <option value="sw">Austria</option>
  <option value="as">Australia</option>
</select>
```

→ **multiple** (nie przyjmujący wartości) – umożliwiający wybranie kilku wartości z listy

```
<select name="kraj2" id="country2" multiple>
  <option value="pl">Polska</option>
  <option value="es">Hiszpania</option>
  <option value="de">Niemcy</option>
</select>
```





# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI - FORMULARZ

**<optgroup>****</optgroup>** - znacznik grupujący logicznie lub funkcjonalnie pola wyboru listy rozwijanej



```
<select name="kraj" id="country">
  <optgroup label="Kraje UE">
    <option value="pl">Polska</option>
    <option value="es" selected>Hiszpania</option>
    <option value="de">Niemcy</option>
  </optgroup>
  <optgroup label="Kraje spoza UE">
    <option value="ch">Chiny</option>
    <option value="ua" selected>Ukraina</option>
    <option value="ru">Rosja</option>
  </optgroup>
</select>
```

**<optgroup>** powinien posiadać atrybut **label**, którego wartość określa nazwę grupy.

Opcjonalnie może także posiadać atrybut **disabled** (bez wartości), który spowoduje wyłączenie z wyboru wszystkich opcji w grupie, którą dany znacznik opisuje.



# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI - FORMULARZ

**<textarea>****</textarea>** - znacznik definiujący wielowierszowe pole tekstowe (obszar tekstowy)



```
<textarea name="textarea" id="message" cols="30" rows="10">
| Lorem ipsum dolor sit amet consectetur adipiscing elit. Hic vero perspiciatis, architecto corrupti
</textarea>
```

**<textarea>** m.in. może posiadać atrybuty **rows** i **cols**, których wartości numeryczne będą określały odpowiednio wysokość (w wierszach tekstu) i szerokość pola tekstowego (w ilości znaków w wierszu). Oczywiście wygodniej stylować jest wygląd pola za pomocą sytlów w CSS.

Jeżeli chcemy ustawić predefiniowaną wartość pola textarea, możemy to zrobić umieszczając ją pomiędzy znacznikami pola. Nie używamy atrybutu value, który jest niepoprawny dla tego znacznika.



# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI - FORMULARZ



`<fieldset></fieldset>` i `<legend></legend>` znaczniki używane do grupowania pól w logiczne lub funkcjonalne elementy

```
<fieldset>
  <legend>Personalia:</legend>
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"><br><br>
  <label for="email">Email:</label>
  <input type="email" id="email" name="email"><br><br>
  <label for="birthday">Birthday:</label>
  <input type="date" id="birthday" name="birthday"><br><br>
</fieldset>
```



# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI - FORMULARZ

**<button></button>** - znacznik definiujący klikalny przycisk



```
<button type="reset">Resetuj</button>

<button type="submit"></button>

<button type="button">Nic nie robię</button>
<button type="button" onclick="alert('Pokaż mnie!');">Wyświetl alert</button>
```

**<button>** podobnie jak input wykonawczy może posiadać 3 typy: **submit**, **reset** oraz **button**.

Buttons, wykonują dokładnie taką samą akcję jak wykonawcze inputy o odpowiednich typach. Czym zatem się różnią? Przede wszystkim są znacznikami podwójnymi, czyli możemy w nich umieścić inne znaczniki (poza formularzowymi). Do tego dają się dobrze stylować 😊

Default Button

CSS Button



# HTML – STRUKTURA DOKUMENTU – PODSTAWOWE ZNACZNIKI - FORMULARZ

**<datalist></datalist>** - znacznik definiujący listę predefiniowanych wyborów/opcji dla elementu <input>.  
Uwaga nie mylić z <select>.



```
<datalist id="browsers">
  <option value="Internet Explorer">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>
```

Znacznik <datalist> służy do udostępniania funkcji „autouzupełniania” dla elementów <input>.

Podczas wprowadzania danych użytkownicy zobaczą rozwijaną listę predefiniowanych opcji.  
Atrybut **id** elementu <datalist> musi być równy atrybutowi **list** elementu <input> (to łączy je ze sobą logicznie).



# HTML – STRUKTURA DOKUMENTU – ZNACZNIKI BLOKOWE / LINIOWE

Każdy znacznik HTML ma domyślną wartość wyświetlania: blokową (**block**) lub liniową (**inline**). Właściwość tą można zmienić dla każdego elementu odpowiednim wpisem w stylach elementu (CSS).

## BLOCK:



## INLINE:





# HTML – STRUKTURA DOKUMENTU – ZNACZNIKI BLOKOWE / LINIOWE

## Elementy domyślnie blokowe

<address> <article> <aside> <blockquote> <canvas> <dd> **<div>** <dl> <dt> <fieldset> <figcaption>  
<figure> <footer> <form> <h1>-<h6> <header> <hr> <li> <main> <nav> <noscript> <ol> **<p>** <pre>  
<section> <table> <tfoot> <ul> <video>

## Elementy domyślnie liniowe

<a> <abbr> <acronym> **<b>** <bdo> <big> **<br>** <button> <cite> <code> <dfn> **<em>** <i> <img> <input>  
<kbd> **<label>** <map> <object> <output> <q> <samp> **<script>** **<select>** **<small>** **<span>** **<strong>** **<sub>**  
**<textarea>** <time> <tt>





# HTML – STRUKTURA DOKUMENTU – ZNACZNIKI BLOKOWE / LINIOWE

## Elementy blokowe

- Element blokowy zawsze zaczyna się od nowej linii i zajmuje całą dostępną szerokość (rozciąga się w lewo i prawo tak daleko, jak to możliwe)

## Elementy liniowe

- Element liniowy jest przeciwieństwem elementu na poziomie bloku.
- Jak sama nazwa wskazuje, element Inline nie zaczyna się od nowej linii, ale zostanie osadzony w istniejącej linii.
- Element **inline** zajmuje tylko tyle szerokości, ile jest to konieczne
- Element ignoruje atrybuty width i height

## Elementy liniowo-blokowe

- Element liniowy przejmujący właściwości blokowe tj. słuchający się atrybutów width i height





# HTML – STRUKTURA DOKUMENTU – ZNACZNIKI BLOKOWE / LINIOWE

JULIA EVANS  
@b0rk

## inline vs block

HTML elements default  
to **inline** or **block**

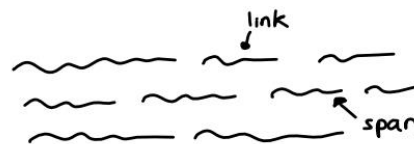
example block  
elements

`<p>` `<div>`  
`<ol>` `<ul>` `<li>`  
`<h1>` - `<h6>`  
`<table>` `<form>`  
`<article>` `<nav>`

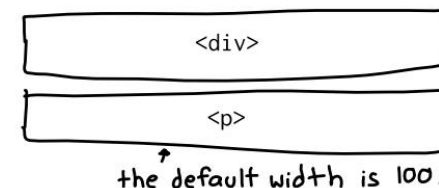
example inline  
elements

`<a>` `<span>`  
`<strong>` `<i>`  
`<button>` `<input>`  
`<small>` `<abbr>`  
`<textarea>`

**inline** elements are  
laid out horizontally



**block** elements are laid  
out vertically by default



**inline elements**  
ignore width & height

Setting the width is impossible,  
but you can use line-height  
to change the height

also, inline elements ignore  
the vertical padding of other  
inline elements

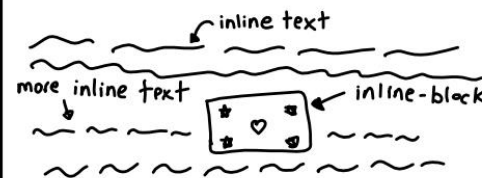
**display can force an  
element to be inline or block**

display determines 2 things:

- ① whether the element itself is  
inline, block, inline-block, etc
- ② how child elements are laid out  
(grid, flex, table, default, etc)

`display: inline-block;`

inline-block makes a block  
element that's laid out  
horizontally like an inline element



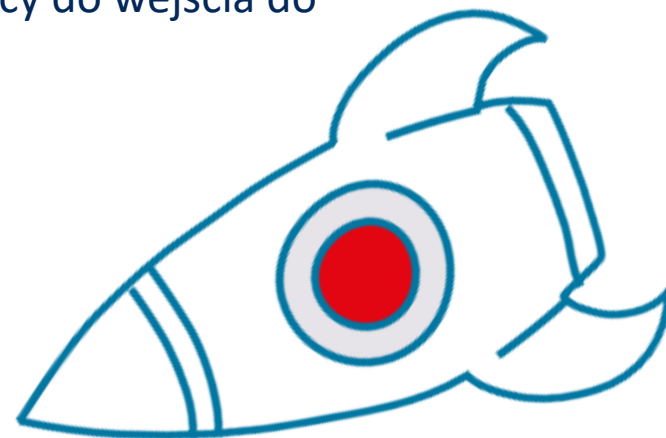


# HTML – STRUKTURA DOKUMENTU – UKŁADANIE STRUKTURY

Istnieją dwie uznane metody układania bloków obok siebie. Obydwie realizowane są poprzez wykorzystanie stylu elementu.

Jedna wymaga użycia w stylach tzw. opływania float:left połączonego z jego wyczyszczeniem opływania poprzez użycie clear:both, zaś w drugiej używamy zapisu display: inline-block.

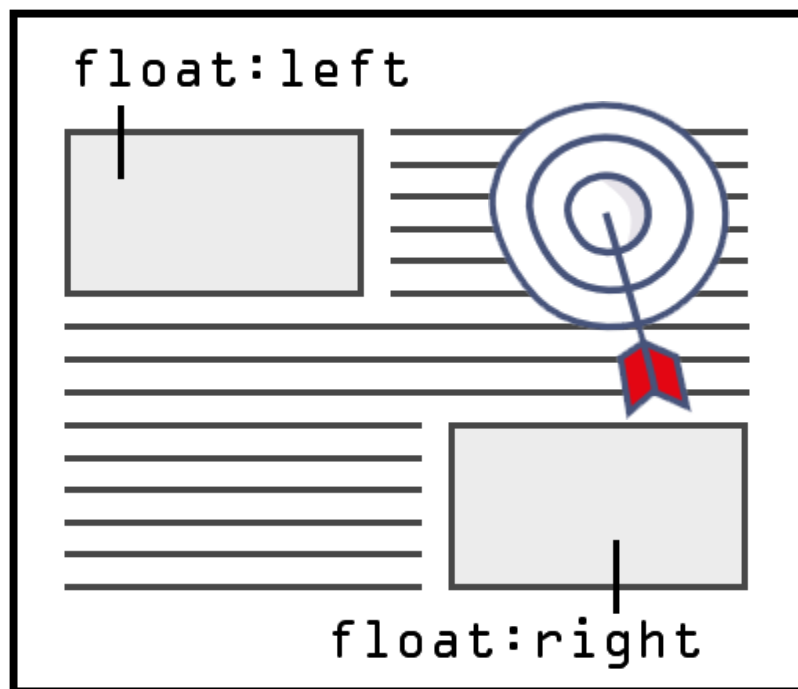
Istnieją także rozwiązania frameworkowe (np. bootstrap) oraz flexbox, kandydujący do wejścia do standardów W3C.





# HTML – STRUKTURA DOKUMENTU – UKŁADANIE STRUKTURY – FLOAT

Float to realizacja koncepcji tzw. opływania elementu - znamy tę sztuczkę m.in. z Worda czy z innego edytora treści, w których można było ustawić opływanie wstawionych obrazów tekstem

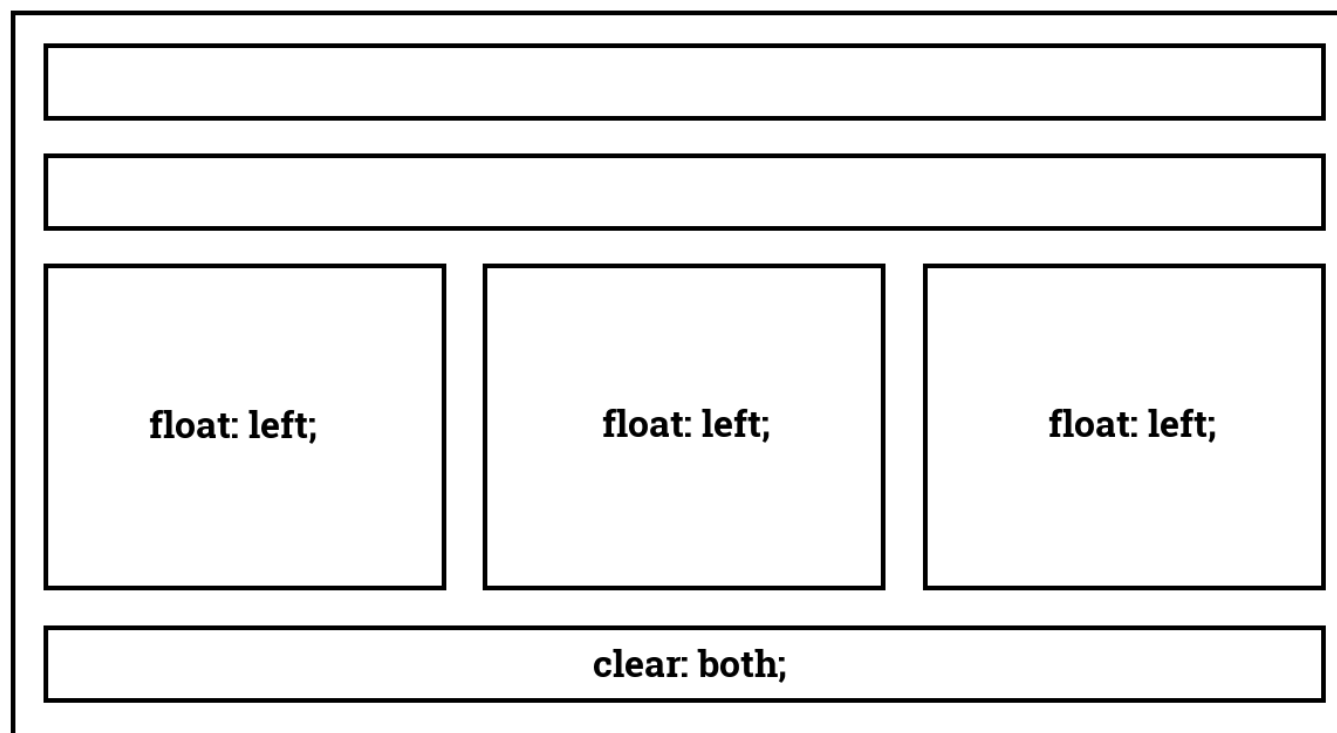


<https://pasja-informatyki.pl/>



# HTML – STRUKTURA DOKUMENTU – UKŁADANIE STRUKTURY – FLOAT

W praktyce metoda ta polega na tym, iż wszystkie bloki, które mają być ułożone obok siebie, będą posiadały w swoich klasach CSS dodatkowy atrybut `float:left`. A następny blok, który już nie ma się układać poziomo, ma posiadać zapis czyszczący opływanie: `clear:both`;



<https://pasja-informatyki.pl/>

Atrybut `float:left` włączył opływanie do lewej strony dla każdego kolejnego bloku, stąd na końcu tego „piętra” trzeba nam jeszcze to opływanie wyłączyć – i dlatego też w stopce pojawił się zapis: `clear:both`. Oto cała filozofia – to wszystko co trzeba wiedzieć, aby opływania poprawnie użyć.



# HTML – STRUKTURA DOKUMENTU – UKŁADANIE STRUKTURY – FLOAT

## Zalety podejścia float:left

- Opływanie poprawnie zadziała we wszystkich przeglądarkach internetowych, nawet w IE 6/7 (wspierała kompatybilność wsteczna).
- Brak jakichkolwiek problemów ze znakami białymi, które to problemy występują w konkurencyjnym podejściu display: inline-block (niewymagane są obejścia bądź hacki likwidujące problem znaków białych).

## Wady podejścia float:left

- Zachodzi konieczność wyczyszczenia opływania zapisem clear:both w dodatkowym elemencie blokowym (choć można zapis clear:both umieścić w bloku przewidzianym w layoucie).
- Elementy blokowe z włączonym opływaniem zostają „wyjęte” z naturalnego flow witryny (i dlatego zachodzi konieczność wyłączenia opływania clear: both).
- Rozwiązanie to zostało opracowane na potrzeby opływania obrazów tekstem, i dopiero później zaadoptowane do układania z jego pomocą bloków strukturalnych.

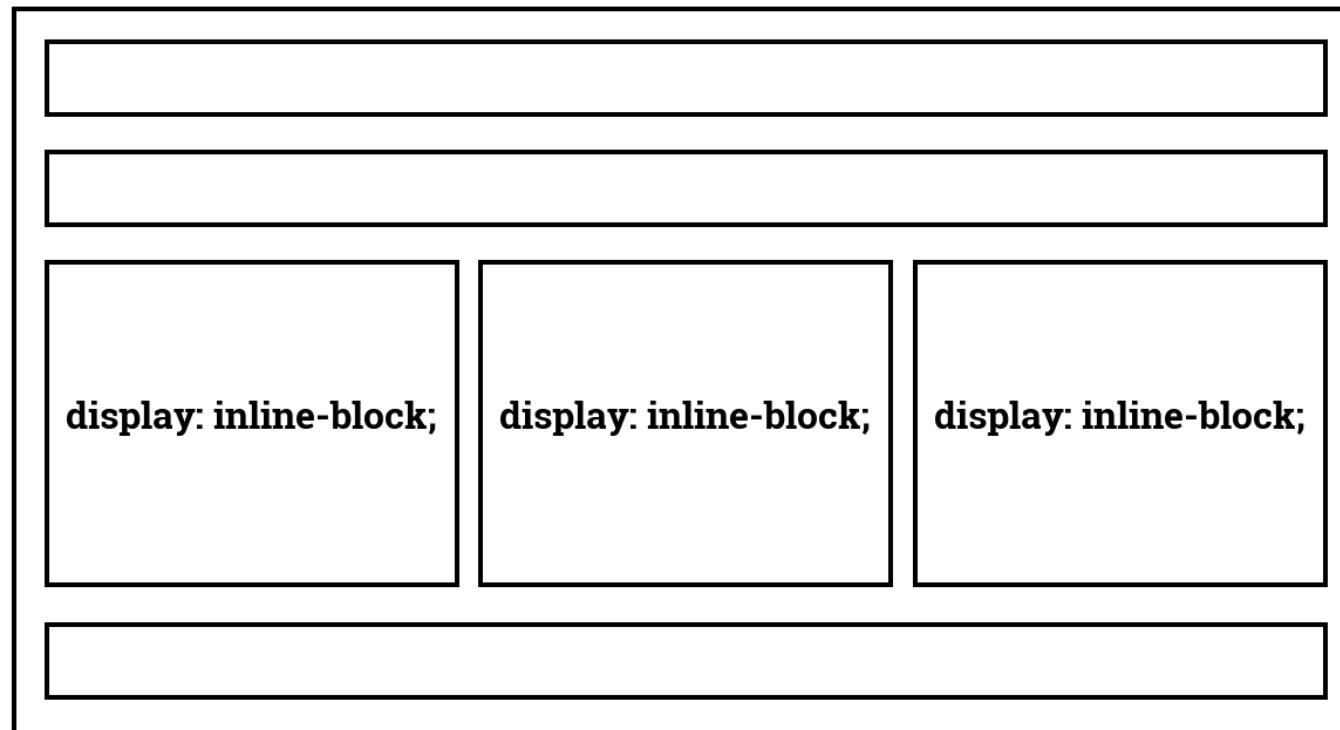




# HTML – STRUKTURA DOKUMENTU – UKŁADANIE STRUKTURY – DISPLAY

## Metoda `display:inline-block;`

Atrybut `display:inline-block` wstawiamy do każdego bloku, który ma być wyświetlony w linii – oto cała filozofia tego rozwiązania. Nie zachodzi tutaj w ogóle potrzeba stosowania zapisu `clear: both`, gdyż nie „wrywamy” bloków z domyślnego flow witryny





# HTML – STRUKTURA DOKUMENTU – UKŁADANIE STRUKTURY – DISPLAY

Niestety w metodzie tej występują problemy z występowaniem znaków białych (odstępów poziomych) pomiędzy blokami ustawionymi w linii. Niewielkie pionowe odstępy (choć występują) nie stanowią taki wielkiego problemu jak „spacje” poziome.

## Zalety podejścia

- Nie występuje potrzeba czyszczenia float:leftu z użyciem clear:both w dodatkowym elemencie blokowym
- Metoda nowsza, pomyślana jako alternatywa do „wyrzucającego” bloki z flow witryny opływania
- Bloki z ustawionym inline-blockiem „słuchają” ustawienia text-align:center kontenera, w którym się znajdują

## Wady podejścia

- Metoda nie działa poprawnie w IE 6/7 (słabsza kompatybilność wsteczna).
- Problemy z występowaniem znaków białych (odstępów pomiędzy blokami) – w zależności od zastosowanej metody poradzenia sobie z tym problemem, wystąpią odmienne „skutki uboczne”.





# HTML – STRUKTURA DOKUMENTU – UKŁADANIE STRUKTURY – METODY ALTERNATYWNE



## Flexbox

To nowy, alternatywny sposób tworzenia struktury witryny, który być może stanie się w przyszłości pełnoprawnym standardem W3C. Aktualnie problemem (jak w przypadku wielu nowych rozwiązań) jest słaba kompatybilność wsteczna takiej struktury.



## Bootstrap

Bootstrap, najpopularniejszy na świecie framework do tworzenia responsywnych witryn dostosowanych do urządzeń mobilnych.





# HTML – STRUKTURA DOKUMENTU – ZNACZNIKI SEMANTYCZNE

Zawartość witryny można oczywiście wstawiać przypadkowo lub porządkowo z użyciem elementu `<div>`. Jednakże jeżeli zależy nam na dostępności do tworzonych treści należy skorzystać z semantyki, a konkretnie znaczników semantycznych.

**Semantyka w programowaniu to precyzyjnie określenie znaczenia poszczególnych symboli oraz ich funkcji w programie, w naszym ujęciu znaczenia poszczególnych elementów / treści / znaczników w stronie internetowej.**

**Znaczniki semantyczne w HTML5:**

**`<header>`**

**`<nav>`**

**`<main>`**

**`<section>`**

**`<article>`**

**`<aside>`**

**`<footer>`**

`<time>`

`<details>`

`<dialog>`

`<summary>`

`<figure>`

`<figcaption>`

`<mark>`

`<h1>-<h6>`

`<em>`

`<i>`

`<b> <strong>`

`<q>`

`<a>`

`<ol> <ul>`

`<p>`

`<hr>`

`<address>`

Semantyka treści

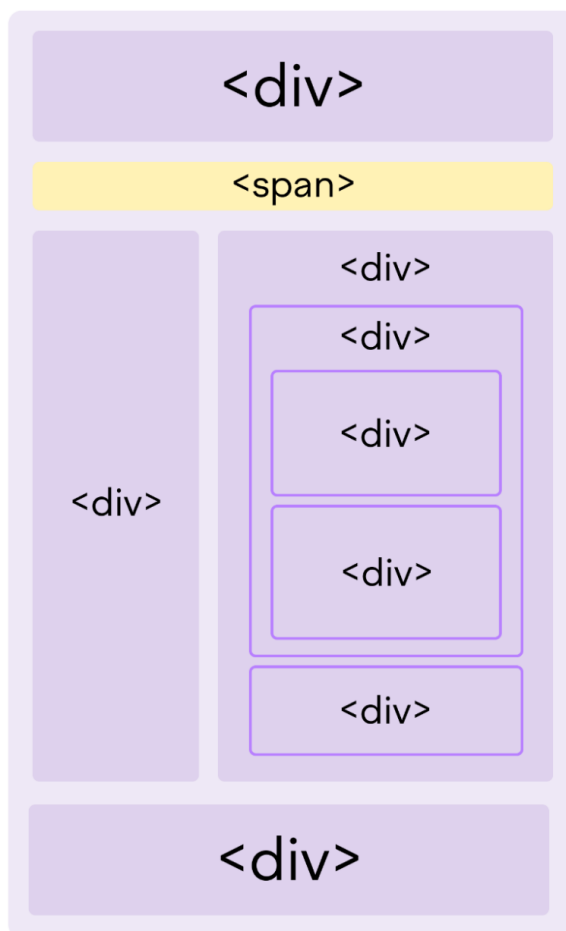
Semantyka układu elementów  
+ znaczenia treści



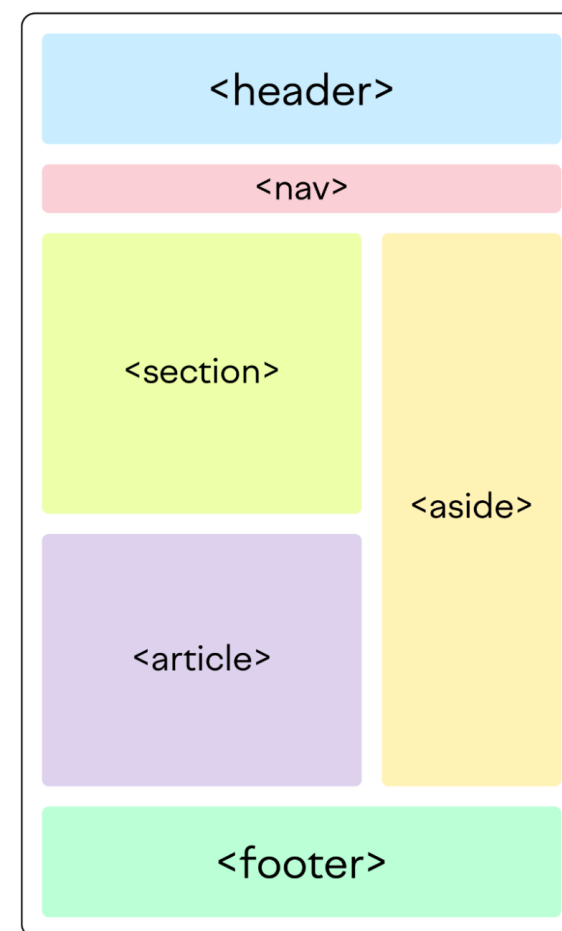
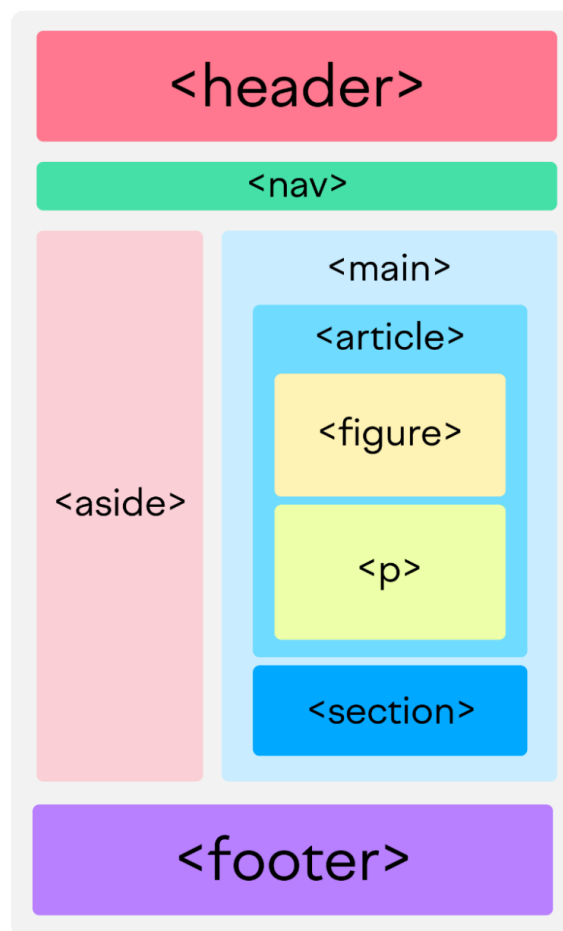


# HTML – SEMANTYKA VS BRAK SEMANTYKI – <BODY>

Non-Semantic HTML



Semantic HTML





# HTML – STRUKTURA DOKUMENTU – ZNACZNIKI SEMANTYCZNE

Element **<header>** reprezentuje kontener na treść wprowadzającą do strony lub sekcji. Może, a nawet powinien zawierać także linki nawigacyjne.

Element **<header>** zazwyczaj zawiera i/lub/oraz:

- jeden lub więcej elementów nagłówka (**<h1>** - **<h6>**)
- logo lub ikona
- treść wprowadzającą
- Nawigację

**Uwaga:** W jednym dokumencie HTML można umieścić kilka elementów **<header>**. Jednak **<header>** nie może być umieszczony w **<footer>**, **<address>** lub innym elemencie **<header>**.



# HTML – STRUKTURA DOKUMENTU – ZNACZNIKI SEMANTYCZNE

Znacznik **<nav>** definiuje zestaw linków nawigacyjnych.

Zauważ, że NIE wszystkie linki dokumentu powinny znajdować się wewnątrz elementu **<nav>**.

**Element **<nav>** jest przeznaczony tylko dla głównych bloków łączy nawigacyjnych, lecz nie jest to zasadą!**

Przeglądarki, takie jak czytniki ekranu dla niepełnosprawnych użytkowników, mogą użyć tego elementu, aby określić, czy pominąć wstępne renderowanie tej zawartości.

Jednakże znacznik ten może być zagnieżdżony nie tylko w tagu **<header>**, ale dodatkowe znaczniki nawigacji **<nav>** są również często używane w innych miejscach na stronie.



# HTML – STRUKTURA DOKUMENTU – ZNACZNIKI SEMANTYCZNE

Znacznik **<main>** określa główną treść dokumentu.

Treść wewnątrz elementu `<main>` powinna być **unikalna dla dokumentu**. Nie powinien zawierać treści powtarzających się w innych elementach, takich jak paski boczne, łącza nawigacyjne, informacje o prawach autorskich, logo witryn i formularze wyszukiwania.

**Uwaga:** W dokumencie nie może być więcej niż jeden element `<main>`. Element `<main>` NIE powinien być potomkiem elementu `<article>`, `<aside>`, `<footer>`, `<header>` lub `<nav>`.



# HTML – STRUKTURA DOKUMENTU – ZNACZNIKI SEMANTYCZNE

Znacznik **<article>** określa niezależną, samodzielną treść.

Zawartość tagu <article> powinna mieć sens sama w sobie i powinna umożliwiać jej dystrybucję niezależnie od reszty strony, na której się znajduje (istnieć niezależnie od reszt treści).

Niekoniecznie oznacza to „post na blogu”. Pomyśl o tym bardziej jako o „artykule ubioru” - samodzielnym elemencie, który może być używany w różnych kontekstach.

## **Potencjalne źródła elementu <article>:**

- post na forum
- post na blogu
- historia z wiadomości
- recenencja/komentarz pod wpisem blogowym
- opis produktu
- ....



# HTML – STRUKTURA DOKUMENTU – ZNACZNIKI SEMANTYCZNE

Znacznik **<article>** określa niezależną, samodzielną treść.

Zawartość tagu `<article>` powinna mieć sens sama w sobie i powinna umożliwiać jej dystrybucję niezależnie od reszty strony, na której się znajduje (istnieć niezależnie od reszt treści).

Niekoniecznie oznacza to „post na blogu”. Pomyśl o tym bardziej jako o „artykule ubioru” - samodzielnym elemencie, który może być używany w różnych kontekstach.

## **Potencjalne źródła elementu `<article>`:**

- post na forum
- post na blogu
- historia z wiadomości
- recenencja/komentarz pod wpisem blogowym
- opis produktu
- ....



# HTML – STRUKTURA DOKUMENTU – ZNACZNIKI SEMANTYCZNE

Użycie znacznika **<section>** to sposób grupowania pobliskich treści o podobnym motywie.

Znacznik sekcji różni się od znacznika artykułu. Niekoniecznie jest samowystarczalny, ale stanowi część czegoś innego. Wg. dokumentacji tag ten definiuje po prostu „sekcję w dokumencie” :)

Przykłady sekcji: sekcja artykułów, sekcja komentarzy, sekcja opinii.







# HTML – STRUKTURA DOKUMENTU – ZNACZNIKI SEMANTYCZNE

Znacznik **<aside>** - element z samej nazwy "na boku" definiuje mniej ważną treść.

Jest często używany do pasków bocznych - obszarów, które dodają uzupełniające, ale mniej istotne informacje.

Treści poboczne **powinny być jednak pośrednio powiązane z treściami otaczającymi.**

Content w tym znaczniku może być np. zbiorem linków powiązanych, archiwum, przypisów itp.  
Często używany jednak do nawigacji pomocniczej.





# HTML – STRUKTURA DOKUMENTU – ZNACZNIKI SEMANTYCZNE

Znacznik **<footer>** definiuje stopkę dokumentu lub sekcji.

Element **<footer>** zazwyczaj zawiera:

- informacje o autorstwie
- informacje o prawach autorskich
- informacje kontaktowe
- mapę witryny
- powrót do górnych linków
- powiązane dokumenty
- link do polityki prywatności

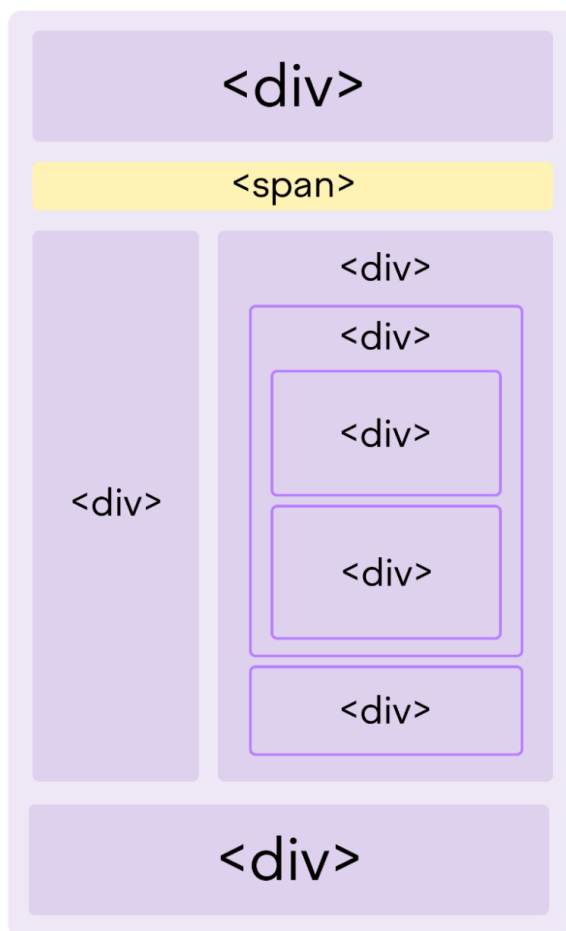
Możesz mieć kilka elementów **<footer>** w jednym dokumencie, zwykle jednak używany jest jednokrotnie.



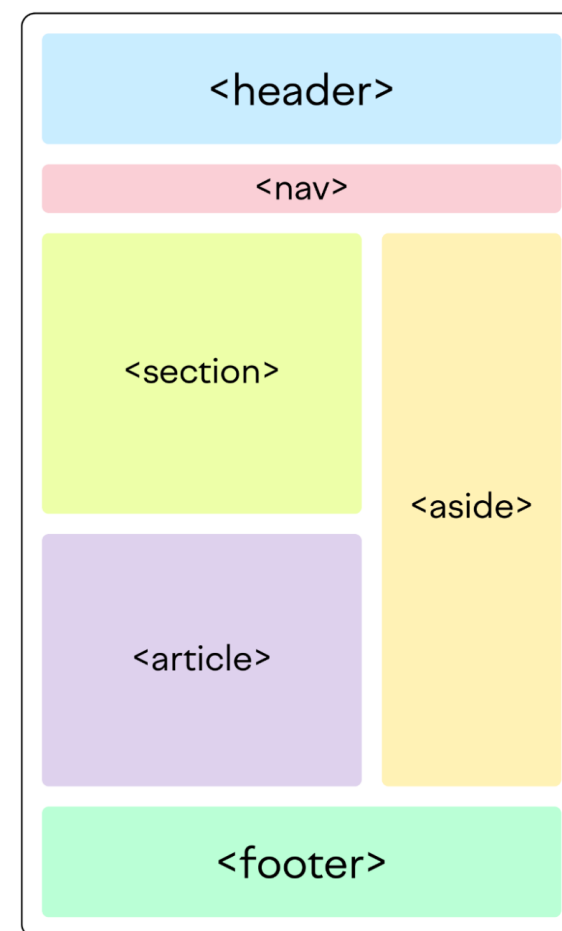
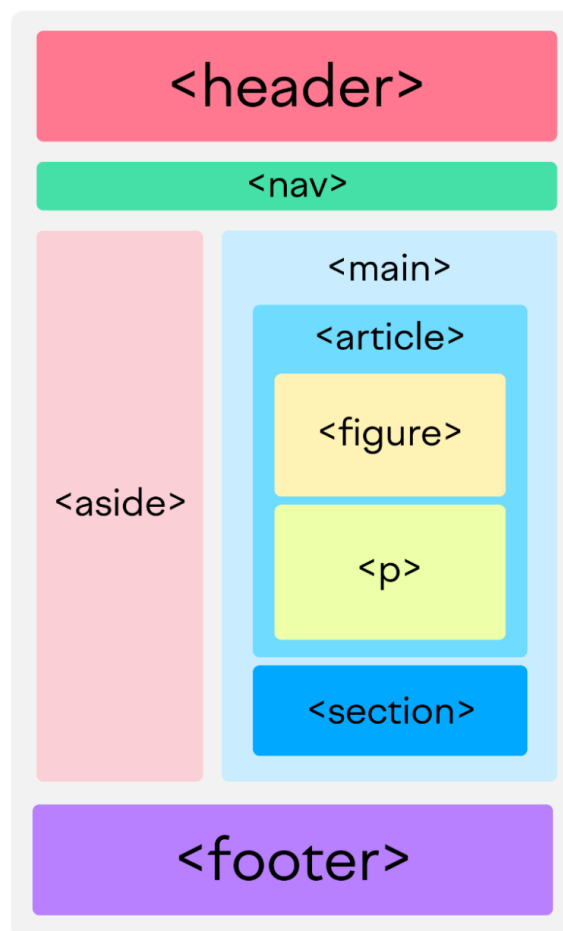


# HTML – SEMANTYKA VS BRAK SEMANTYKI – <BODY>

Non-Semantic HTML



Semantic HTML



— </HTML> —

