

Wprowadzenie do Aplikacji Internetowych (Webowych) wykład 2

dr inż. Grzegorz Rogus
rogus@agh.edu.pl

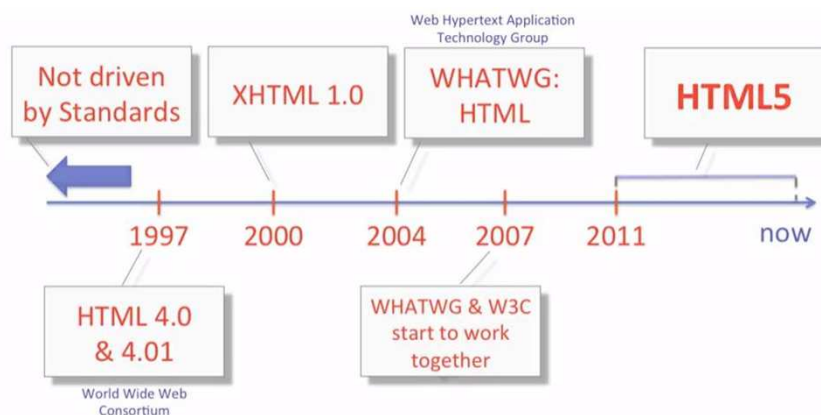
JĘZYK ZNACZNIKÓW HTML

JĘZYK HTML

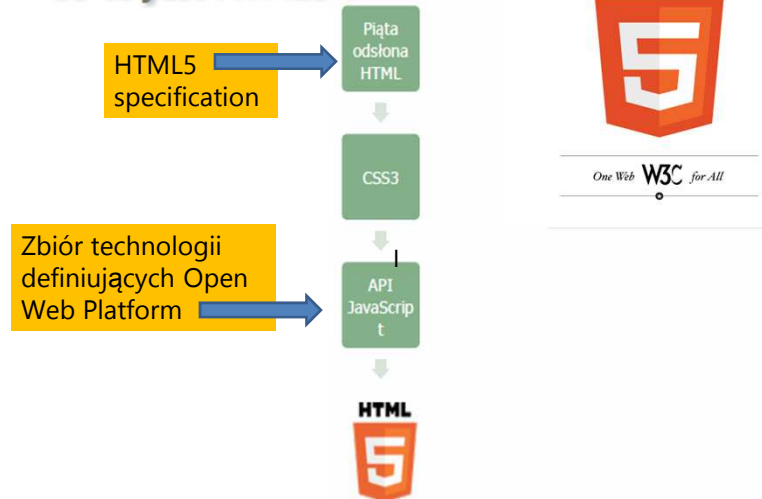
(HyperText Markup Language, hipertekstowy język znaczników)

- Język używany do tworzenia stron internetowych
- Język znaczników, które umożliwiają nam określenie ról naszych elementów na stronie. Możemy za jego pomocą przekazać, że dany element będzie nagłówkiem, akapitem bądź obrazkiem.
- Jest niezależny od systemu operacyjnego i parametrów sprzętowych komputera na którym strona jest wyświetlana
- Właściwie możemy myśleć o naszych elementach strony jak o *klockach lego*, z których będziemy budować. Każdy taki "klocek" musi być zapisany w specjalny sposób - za pomocą **tagów HTML**.

Ewolucja



Co to jest HTML5 ?



tworzenie serwisów i aplikacji które do tej pory były zarezerwowane tylko dla aplikacji desktopowych

HTML 5 — czego dotyczy

- 🔍 Przechowywanie danych
- ≡ Semantyka HTML 5
- 📁 Dostęp do systemu plików urządzeń
- 🔗 Łączność
- ⚙️ Wydajność i integracja
- 🎮 3d, grafika, efekty
- 📺 Multimedia
- 🎨 CSS 3

Teraz kilka słów o składni i semantyce HTML

Na początek powrót do postaci klasycznej

KOMPONENTY JĘZYKA HTML

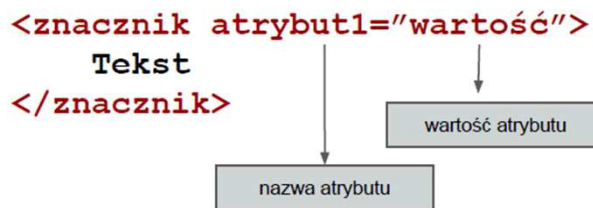
Język HTML składa się z kilku kluczowych komponentów:

- deklaracji typu dokumentu,
- znaczników,
- atrybutów,
- typów danych,
- referencji znakowych i odwołań w postaci encji.

Podstawy składni ZNACZNIKI (TAGI)

<znacznik>

Znacznik poza słowem kluczowym definiującym jego cechę, może zawierać różne atrybuty modyfikujące jego podstawowe własności. Atrybuty są definiowane w znaczniku otwierającym w następujący sposób:



Uwaga:

- nazwy elementów (znaczników) oraz atrybutów według standardu XHTML powinny być pisane małymi literami.

ATRYBUTY SPECJALNE

Atrybutu **title** używa się w celu dodania do elementu informacji tekstowej.

```
<span title="Hypertext Markup Language">HTML</span>
```

Atrybut **id** nadaje elementowi unikalny identyfikator.

```
<div id="site-header">...</div>
```

Atrybut **class** służy do sklasyfikowania elementów dla celów prezentacyjnych.

```
<span class="title">Tytuł w tekście</span>
```

TYPY DANYCH

Typy danych wprowadzane jako wartości atrybutów:

- skrypty (script data) `<script type="text/javascript">`
- dane arkuszy stylów ``
- identyfikatory i nazwy `<div id="siteId"><form name="kontakt">`
- adresy URI ``
- liczby i jednostki `<div width="200"> <div width="200px">`
- języki i kodowanie znaków `<html lang="en"><meta charset="utf-8">`
- deskryptory mediów `<object data="helloworld.swf">`
- kolory ``

ZNACZNIKI (TAGI)

`<p />`



`<p></p>`



Niektóre elementy nie posiadają zawartości, a także znacznika zamykającego:

`
`

``

`<hr>`

`<input type="text" name="imie" id="imie" value="Jan">`

TYPY ZNACZNIKÓW HTML

- **Znaczniki strukturalne** – opisują logiczną strukturę dokumentu,
- ~~**Znaczniki prezentacyjne** – opisują wygląd poszczególnych elementów (stopniowo usuwane),~~
- **Znaczniki hipertekstowe** – zawierają linki do innych dokumentów lub innych części tego samego dokumentu.

Znaczniki prezentacyjne

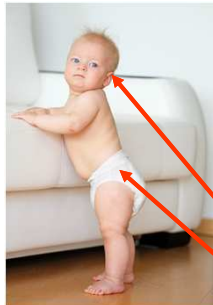
Semantyka: Redukcja znaczników

- ~~<applet>~~
- ~~<big>~~
- ~~<center>~~
- ~~~~
- ~~<frame>~~
- ~~<frameset>~~
- ~~<strike>~~
- ...

Zaczynamy projekt strony



Strona internetowa jest jak dziecko: trzeba ją zaplanować i zrobić.



U dziecka są raczki, nóżki, głowa, tułów itp.

Strona potrzebuje tylko głowy (head) oraz ciała (body)

DOKUMENT HTML

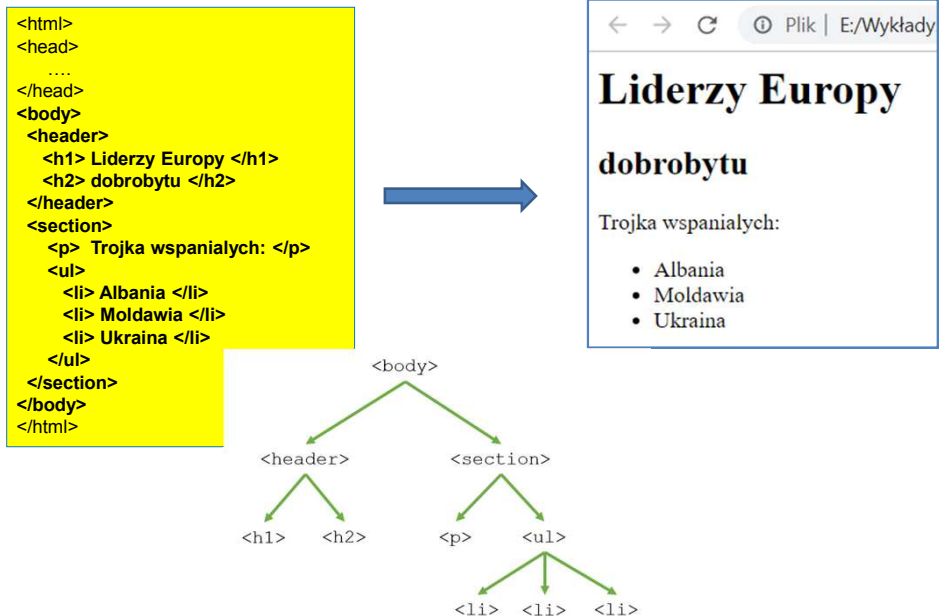
Prawidłowo skonstruowany dokument HTML składa się z dwóch części:

- **sekcja head** - zawiera podstawowe informacje o dokumencie, w tym znaczniki META, służące do identyfikacji witryny internetowej przez przeglądarki oraz wyszukiwarki; tu znajduje się informacja o kodowaniu, o zewnętrznych plikach i bibliotekach, które chcemy na stronie wykorzystać.
- **sekcja body** – zawartość dokumentu HTML wyświetlana w oknie przeglądarki internetowej. To, co widzi odwiedzający witrynę internetową.

Podstawowa struktura dokumentu

<code><!DOCTYPE html></code>	deklaracja typu dokumentu
<code><html></code>	początek dokumentu hipertekstowego
<code><head></code>	sekcja nagłówkowa
<code><title></code> Tytuł strony <code></title></code>	tytuł strony
<code></head></code>	
<code><body></code>	
<code><p>Treść strony</p></code> <code></body></code>	sekcja zawartości strony
<code></html></code>	koniec dokumentu hipertekstowego

Anatomia dokumentu HTML



DEKLARACJA TYPU DOKUMENTU (DOCTYPE)

Służy do celów poprawnej walidacji utworzonej strony.

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

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

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

<!DOCTYPE html>

Deklaracja DOCTYPE mówi nam o wersji i typie dokumentu HTML.

Nie ma znaczenia z punktu widzenia treści,
jednak jest bardzo ważna dla przeglądarki
interpretującej nasz kod

Nagłówek <head> ... </head>

Znacznik **head** otacza nagłówek dokumentu **HTML**. Zawiera informacje o tytule dokumentu, meta-informacje, a w większości przypadków także dołączone skrypty. W ramach tego znacznika należy zagnieździć znacznik **<title>**, który zawiera tekst, wyświetlany przez przeglądarkę jako tytuł strony. Następujące znaczniki mogą znaleźć się w sekcji nagłówka:

<title>, **<style>**, **<base>**, **<link>**, **<meta>**, **<script>**, **<noscript>**

obowiązkowy w sekcji head

znaczniki opcjonalne

Przykład:

```
<head>
  <link rel="stylesheet" type="text/css" href="styles.css" />
  <title>Tytuł dokumentu</title>
  <meta name="description" content="Przykładowa strona" />
  <meta name="keywords" content="przykład, nagłówek, strona" />
  <script type="text/javascript" src="skrypt.js">
</script>
</head>
```

Dodawanie arkusza stylu i skryptów

Style CSS:

```
<head>
  <link href="styl.css" rel="stylesheet">
</head>
```

W porównaniu z
HTML4 i XHTML
nie ma atrybutu
`type="text/css"`

Uwaga:
Ten znacznik musi
być zapisany
dokładnie w ten
sposób, a nie
`<script/>`

Skrypty:

```
<head>
  <script src="skrypt.js"></script>
</head>
```

Kodowanie znaków i język strony

W HTML5 kodowanie znaków jest bardzo ułatwione i sprowadza się do jednej dyrektywy meta w części head naszej strony internetowej:

Kodowanie:

```
<head>
  <meta charset="utf-8">
  <title> Gzegzółka </title>
</head>
```

Dodatkowo dobrą praktyką i oznaką profesjonalizmu jest wskazanie języka strony, co pozwala np. wyszukiwarką na odfiltrowywanie treści pasujących tylko do języka wyszukiującego. Atrybut wskazujący na język można dodać do dowolnego znacznika jednak najczęściej umieszcza się go w znaczniku html na początku strony:

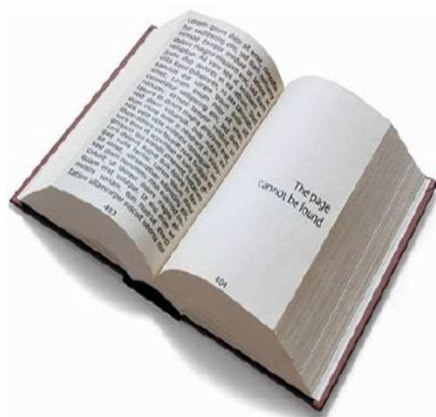
```
<html lang="pl">
```

Minimalistyczna wersja szablonu strony w HTML5

```
<!DOCTYPE html>
<html lang="pl">
  <head>
    <meta charset="utf-8">
    <link href="styl.css" rel="stylesheet">
    <title>Tytuł strony</title>
    <script src="skrypt.js"></script>
  </head>
  <body>
    <p>Treść strony</p>
  </body>
</html>
```

Budowanie struktury strony

Na początek warto wyobrazić sobie stronę internetową jak książkę z rozdziałami, podrozdziałami, pod-pod-rozdziałami, akapitami, listami, ilustracjami



Lista znaczników

<!--...-->	Komentarz	<head>	Definicja informacji o dokumencie
<!DOCTYPE>	Definicja typu dokumentu	<hr>	Pozioma linia
<a>	Odnośnik (kotwica)	<iframe>	Ramka pływająca (iframe)
<blockquote>	Komentarz blokowy		Obrazek
<body>	Ciało dokumentu HTML	<input>	Pole formularza (opis ogólny)
 	Przejdźcie do nowej linii	<ins>	Tekst wstawiony
<button>	Przycisk	<label>	Etykieta dla kontrolki formularza
<caption>	Tytuł tabeli	<legend>	Tytuł dla grupy pól
<center>	Centrowanie tekstu (niezalecany)		Element listy
<cite>	Cytat	<link>	Odnośnik do zasobu
<code>	Fragment kodu programu	<map>	Mapa odsyłaczy graficznych
<dd>	Opis definicji	<meta>	Informacje meta
	Tekst skasowany	<noscript>	Sekcja noscript
<dfn>	Definiowane słowo lub wyrażenie	<object>	Osadzony obiekt
<div>	Sekcja w dokumencie		Lista numerowana
<fieldset>	Grupa pól	<option>	Opcja w liście rozwijalnej
<form>	Formularz	<p>	Paragraf
<frame>	Definicja zawartości ramki	<param>	Parametr dla obiektu
<frameset>	Grupa ramek na stronie z ramkami	<pre>	Tekst preformatowany
<h1>	Nagłówek poziomu 1	<samp>	Przykład kodu komputerowego
<h6>	Nagłówek poziomu 6	<script>	Skrypt

Lista znaczników

<select>	Lista wybieralna
<small>	Pomniejszony tekst
	Sekcja w dokumencie
	Tekst silnie wyróżniony
<style>	Definicja arkusza stylu CSS
<sub>	Tekst w dolnym indeksie
<sup>	Tekst w górnym indeksie
<table>	Tabela
<tbody>	Ciało tabeli
<td>	Komórka tabeli
<textarea>	Pole do wprowadzania wielu linii
<tfoot>	Stopka tabeli
<th>	Komórka nagłówka tabeli
<thead>	Nagłówek tabeli
<title>	Tytuł dokumentu HTML
<tr>	Wiersz tabeli
<tt>	Tekst "maszynowy"
	Lista wypunktowana

i inne ...

Więcej -> www.w3schools.com/tags

ZNACZNIKI HIPERTEKSTOWE - uwagi



Zawierają linki do innych dokumentów lub innych części tego samego dokumentu.

```
<a href="http://szkola.pl" title="Strona szkoły">Szkoła </a>
```

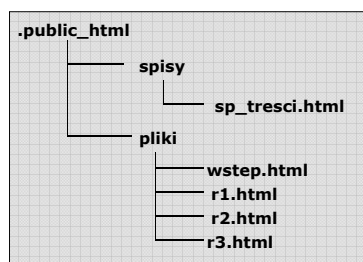
```
<a href="index.html#ofirmie" title="Opis firmy">O nas</a>
```

Zamieszczanie wokół `target="_blank"`.

```
<a href="http://szkola.pl" title="Strona szkoły" target="_blank"> Szkoła </a>
```

Odsyłacze do plików lokalnych

```
<h2>Spis treści:</h2>
<ol>
<li><a href="../pliki/wstep.html">Wstęp</a>
<li><a href="../pliki/r1.html">Rozdział 1</a>
</ol>
<li><a href="../pliki/r1.html#zal">Założenia wstępne</a>
<li><a href="../pliki/r1.html#teza">Teza pracy</a>
</ol>
<li><a href="../pliki/r2.html">Rozdział 2</a>
<li><a href="../pliki/r3.html">Podsumowanie</a>
</ol>
```



Spis treści:

1. Wstęp
2. Rozdział 1
 1. Założenia wstępne
 2. Teza pracy
3. Rozdział 2
4. Podsumowanie

Element Liniowy, a Element Blokowy



- Elementy liniowe zajmą tylko tyle miejsca ile same potrzebują, pozostawiając miejsce na tej samej linii dla kolejnych elementów liniowych.
- Natomiast elementy blokowe zawsze wyświetlane są na nowej linii i zajmują jej całą powierzchnię, nakazując kolejnym elementom ukazanie się pod spodem, na kolejnej linii.

ELEMENTY BLOKOWE

- Tekst preformatowany `<pre>`
- Oddzielanie elementów za pomocą linii `<hr>`
- Listy uporządkowane i nie ``, ``
- Tabele `<table>`
- Akapity `<p>`
- Nagłówki sekcji `<h1><h2><h3><h4><h5><h6>`
- Sekcje `<div>`
- Sekcje `<section>`
- Nagłówki `<header>`
- Stopki `<footer>`
- Nawigacje `<nav>`

Czym jest formularz

- Służy do umieszczania na stronie www elementów interaktywnych:
 - Pola tekstowe
 - Przyciski
 - Listy i listy rozwijalne
 - Checkboxy i Radiobuttony
 - Okno dialogowe „plik”

Typy pól formularza

- INPUT
 - Text – zwykłe pole tekstowe
 - Hidden – pole ukryte dla użytkownika (widoczne w kodzie)
 - Password – pole ukrywające wprowadzane znaki
 - Checkbox – pole wyboru
 - Radio – przycisk radiowy (jeden z wielu)
 - Submit – przycisk wysyłający dane z formularza
 - Reset – przycisk przywracający wartości formularza
 - Image – przycisk w formie obrazka działający jak Submit
 - Button – dowolny przycisk bez szczególnego znaczenia dla formularza
- SELECT – lista rozwijalna
- TEXTAREA – pole tekstowe

Dane użytkownika

Imię i Nazwisko	<input type="text" value="Jan Malinowski"/>
Hasło	<input type="password" value="haslo12345"/>
Wykształcenie	<input checked="" type="radio"/> wyższe (mgr) <input type="radio"/> wyższe (dr) <input type="radio"/> średnie <input type="radio"/> podstawowe
Miejscowość	<input type="text" value="Sopot"/>
Opis	<input type="text" value="To jest miejsce na dłuższy opis..."/>
Języki	<input checked="" type="checkbox"/> Angielski <input checked="" type="checkbox"/> Polski <input type="checkbox"/> Rosyjski <input checked="" type="checkbox"/> Inny
<input type="button" value="Zapisz"/> <input type="button" value="Kasuj"/> <input type="button" value="Lista użytkowników"/>	



Can I use ? Settings

2 results found

WebRTC Peer-to-peer connections

Method of allowing two users to communicate directly, browser to browser using the RTCPeerConnection API.

Usage: Global 87.2% + 2.06% = 89.26%
unprefixed: 82.23% + 2.06% = 84.29%

Current aligned	Usage relative	Date relative	Apply filters	Show all	?									
IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Opera Mobile	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser
		2-21	4-22		10-17									
	12-14	22-43	23-55	3.1-10.1	18-42	3.2-10.3								
6-10	15-17	44-68	56-76	11-12.1	43-60	11-12.3		2.1-4.44	12-12.1				4-9.2	
11	18	69	77	13	62	13.1	all	76	46	76	68	12.12	10.1	1.2
	76	70-71	78-80	13.1-TP										

HTML5 a JavaScript

- Rozwój HTML5 i CSS3 zmienił wzajemne relacje pomiędzy HTML a językiem JavaScript.
- **Z jednej strony:** zminimalizowana została rola JS jako narzędzia upiększania i dynamizowania wizualnego zamieszczanych treści
- **Z drugiej strony:** kreatywne użycie nowych elementów HTML5 (np. canvas, video, audio) nie jest możliwe bez wykorzystania JS i DOM

```
function setDuration() {  
    var duration = document.getElementById("player").duration;  
    document.getElementById("durInfo").innerHTML = "Całość: " + duration.toFixed(2);  
    document.getElementById("bar").max = duration;  
}  
function showInfo() {  
    var player = document.getElementById("player");  
    var info = document.getElementById("currInfo");  
    var bar = document.getElementById("bar");  
    info.innerHTML = "Aktualnie: " + player.currentTime.toFixed(2);  
    bar.value = player.currentTime;  
}  
...  
<video id="player" src = "wildlife.ogg" controls  
    ontimeupdate="showInfo()" ondurationchange="setDuration()">  
    Ta przeglądarka nie odtwarza wideo.  
</video>
```



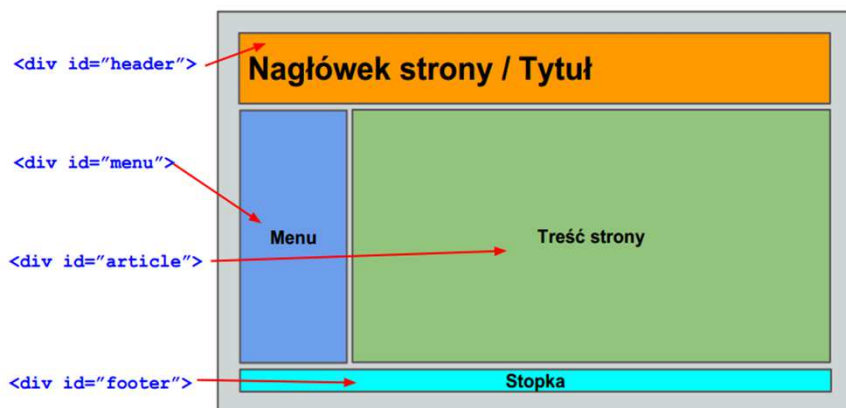
PRZYKŁADOWE ELEMENTY HTML5

- Header `<header></header>`
- Footer `<footer></footer>`
- Nav `<nav></nav>`
- Main `<main></main>`
- Section `<section></section>`
- Article `<article></article>`
- `<audio>` i `<video>` - odtwarzacz muzyki i odtwarzacz filmów
- `<canvas>` - przestrzeń do rysowania w 2D i 3D



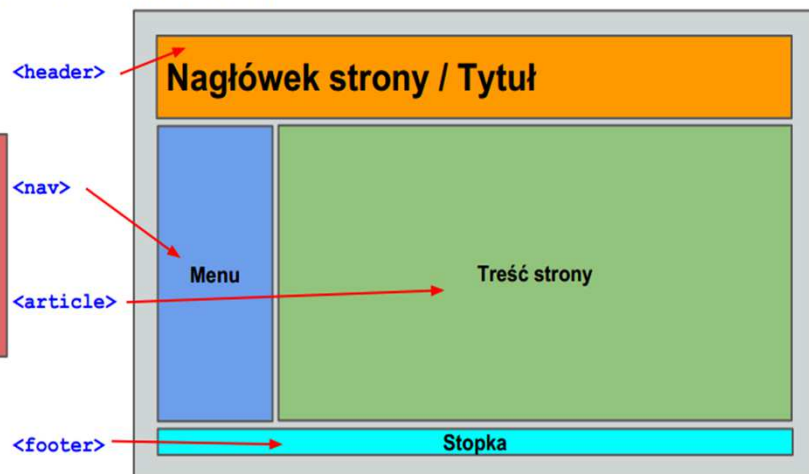
Budowanie struktury strony

Standard HTML5 wprowadza kilkanaście nowych znaczników oraz nowe i lepsze metody organizacji treści na stronie internetowej. Do tej pory strony były zbudowane najczęściej w oparciu o elementy podziału (elementy blokowe) `<div>`, które umożliwiały w miarę logiczne rozdzielanie treści na stronie.



Budowanie struktury strony

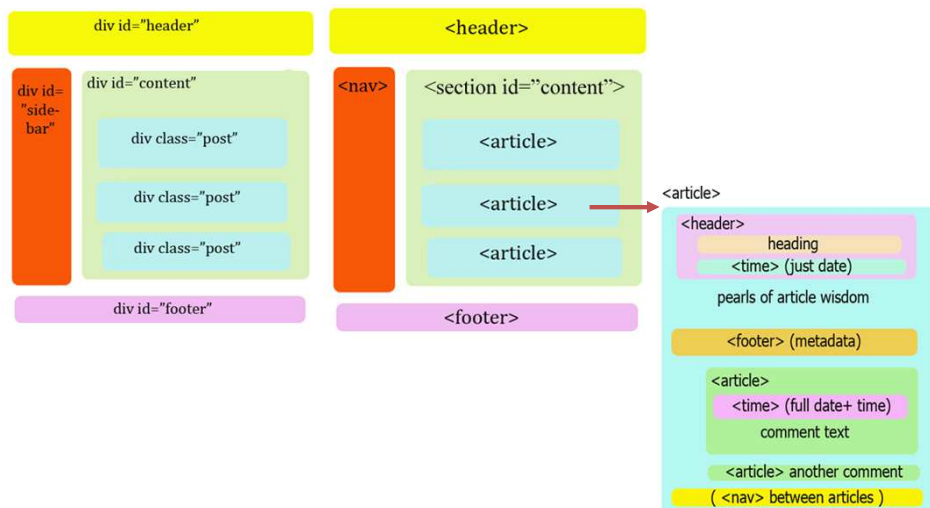
Dzięki HTML5 mamy nowe znaczniki które organizują treści na stronie w sposób logiczny oraz dzięki temu są bardziej zoptymalizowane pod kątem wyszukiwarek. Dodatkowo pozwalają na przyjazną organizację strony i czynią stronę bardziej dostępną np. dla czytników ekranów ułatwiających przeglądanie stron osobą niedowidzącym.



Uwaga:
Znaczniki
same z
siebie nie
robią nic!

Struktura Web page

Projektowanie z html5



Nowe znaczniki semantyczne — mark

```

...
<section>
  <article>
    <header>
      <h2>Tytuł nagłówka artykułu</h2>
      <p>Jakaś dodatkowa treść i np. link: <a href="#">więcej...</a></p>
    </header>
    <p>Znacznik <mark>article</mark> reprezentuje pewną zamkniętą całość,
      która może być traktowana niezależnie od całej strony, np. wpis
      blogowy, artykuł, komentarz.</p>
  </article>
</section>
...

```

Przy okazji — znacznik `<mark>` wyróżnienia fragmentu tekstu.

Tytuł nagłówka artykułu

Jakaś dodatkowa treść i np. link: [więcej...](#)

Znacznik `<article>` reprezentuje pewną zamkniętą całość, która może być traktowana niezależnie od całej strony, np. wpis blogowy, artykuł, komentarz.



Nowe znaczniki semantyczne — details i summary



Znacznik `<details>` określa dodatkową informację, którą użytkownik może zobaczyć lub ukryć na życzenie.

Sterowanie pokazywaniem/ukrywaniem zawartości `<details>` steruje atrybut boolowski `open`.

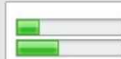
Znacznik `<summary>` określa widoczny nagłówek sekcji `<details>`.

Nowe znaczniki semantyczne — meter i progress

```
<meter value="3" min="1" max="10">3 w skali od 1 do 10</meter><br />  
<meter value="0.4">40%</meter>
```

Gdy brak implementacji

3 w skali od 1 do 10
40%

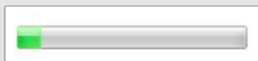


Opera, Chrom

Znacznik `<meter>` definiuje wskaźnik, np. zużycie przestrzeni dyskowej, page rank, wynik ankiety. Uwaga — nie używa się jako paska postępu, do tego jest `<progress>`.

```
<progress value="11" max="100">Ładowanie: 11%</progress>
```

Ładowanie: 11%



Gdy brak implementacji

Opera, Chrom, Firefox

Sterowanie elementem progress – prosty przykład

```
...  
var timeoutObject;  
var sek = 0;  
function display()  
{  
    document.getElementById( "pb" ).value = sek++;  
    if( sek > 100 ) zeruj();  
}  
function start()  
{  
    display();  
    window.timeoutObject = setInterval( "display()", 100 )  
}  
function stop()  
{  
    window.clearInterval( timeoutObject );  
}  
function zeruj()  
{  
    document.getElementById( "pb" ).value = sek = 0;  
}  
...  
<body>  
<h1>Progress bar</h1>  
<progress id="pb" value="0" max="100"></progress>  
<input type="button" value="Start" onclick="start()" />  
<input type="button" value="Stop" onclick="stop()" />  
<input type="button" value="Zeruj" onclick="zeruj()" />  
</body>  
</html>
```

Progress bar



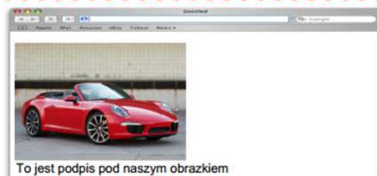
Nowe znaczniki semantyczne: figure i figurecaption

HTML5 wprowadza również pewną nowość ułatwiającą osadzanie ilustracji. Wcześniej aby osadzić na stronie obraz wraz z podpisem musieliśmy:

```
<div>  
    
  <p>To jest podpis pod naszym obrazem</p>  
</div>
```

Standard HTML oferuje nam rozwiązanie semantyczne <figure> dzięki któremu możemy w sposób zorganizowany w ramach jednego bloku osadzać obrazy:

```
<figure>  
    
  <figcaption>To jest podpis pod naszym obrazem</figcaption>  
</figure>
```



W obu przypadkach efekt jest ten sam.

<CANVAS>

```
<canvas id="myCanvas">
```

Your browser does not support the HTML5 canvas tag.

```
</canvas>
```

```
<script>
```

```
var c = document.getElementById("myCanvas");
```

```
var ctx = c.getContext("2d");
```


```
ctx.fillStyle = "#FF0000";
```

```
ctx.fillRect(0, 0, 80, 100);
```

```
</script>
```



Nowa semantyka w Web page - podsumowanie



```
<header>
```

```
<nav>
```

```
<article>
```

```
<section>
```

```
<aside>
```

```
<footer>
```

- > <header>
- > <nav>
- > <article>
- > <section>
- > <aside>
- > <footer>

Inne znaczniki

- > <canvas>
- > <hgroup>
- > <address>
- > <figure>
- > <meter>
- > <progress>
- > <time>

HTML5 Forms

Nowe input types (nowych pola formularza) w HTML5

„Stare” pola formularza w HTML

■ button ■ checkbox ■ file ■ hidden ■ image ■ password ■ radio ■ reset ■ submit ■ text

HTML5 dodaje **13** nowych typów pól:

- color
- date
- datetime
- datetime-local
- email
- month
- number
- range
- search
- tel
- time
- url
- week

Nowe atrybuty:

- autocomplete
- autofocus
- form
- formaction
- formenctype
- formmethod
- formnovalidate
- formtarget
- height and width
- list
- min and max
- multiple
- pattern (regexp)
- placeholder
- required
- step

Forms

<input type= email

<input type= url

<input type= tel

```
<input id="email" type="email"/>
```

```
<input id="website" type="url"/>
```

```
<input id="phone" type="tel"/>
```

Email Address

Web Site

Phone Number

Two mobile form interfaces are shown. The left one has red boxes around the 'Email Address', 'Web Site', and 'Phone' input fields, indicating validation errors. The right one shows the same form with green boxes around the 'Email Address' and 'Web Site' fields, indicating successful validation. Below the forms are two mobile keyboards: a standard QWERTY keyboard and a numeric keypad.

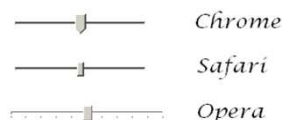
Two mobile form interfaces are shown. The left one has green boxes around the 'Email Address' and 'Web Site' input fields, indicating successful validation. The right one shows the same form with a red box around the 'Phone' input field, indicating a validation error. Below the forms are two mobile keyboards: a standard QWERTY keyboard and a numeric keypad.

Forms

<input type="range">

Wyświetla suwak, dzięki któremu możemy określić w łatwy sposób wartość. dozwolone atrybuty:

- max - określa maksymalną wartość
- min - określa wartość minimalną
- step - określa co ile wartość suwaka ma się przesuwać
- value - definiuje wartość domyślną



<label for="rating">Ocen w skali 1 do 10, znajomosc HTML5:</label>

<input type="range" min="1" max="10" name="rating" type="range">



Ocen w skali 1 do 10, znajomosc HTML5:

Forms

<input type="number">

Określa pole na liczbę, wyświetlając w prawej części strzałki góra/dół do zmiany wartości.

dozwolone atrybuty:

- max - określa maksymalną wartość
- min - określa wartość minimalną
- step - określa co ile wartość suwaka ma się p
- value - definiuje wartość domyślną

Chrome

How often you watch movie in week :

Opera

How often you watch movie in week :

<input type="number" name="ocena" min="1" max="5">

Number of user

license :

Gdyby była potrzeba zmiany wartości za pomocą javascript to funkcje

- ocena.stepUp(x) - x wielkość o jaką zwiększa się wartość pola
- ocena.stepDown(x) - x wielkość o jaką zmniejsza się wartość pola.
- ocena.valueAsNumber – zwraca wartość pola w formie floating zamiast string.

Forms

`<input type="search">`

search : Chrome
search : Safari

Pole na wpisanie frazy do wyszukiwarki. Search Google: `<input type="search" name="googlesearch" />`
Search type -> wsparcie w Chrome, Opera | Safari.

`<input type="color">`

`<input type="color" name="favcolor" />`

Określa pole na wartość koloru. Po kliknięciu pojawia nam się paleta kolorów z wyborem.



Forms

`<input type="date">`

`<input type="month">`

X ▲ ▼

`<input type="week">`

`<input type="time">`

Dodatkowe atrybuty formularza

Regular Expressions

```
• <form action="" method="post">
  <label for="username">Create a Username: </label>
  <input type="text"
    name="username"
    id="username"
    placeholder="4 <> 10"
    pattern="[A-Za-z]{4,10}"
    autofocus
    required>
  <button type="submit">Go </button>
</form>
```

Nowe znaczniki semantyczne — audio

```
<audio src="traktor.mp3" controls autoplay loop preload="none">
  Ta przeglądarka nie odtwarza dźwięku.
</audio>
```

▶ *controls* — wyświetlanie panelu sterowania odtwarzaniem,

▶ *autoplay* — automatyczne odtwarzanie od razu, gdy to możliwe,

▶ *loop* — powtarzanie odtwarzania.

▶ *preload* — wskazówka dla przeglądarki: jak postępować z dźwiękiem w trakcie ładowania strony: *auto* — ładowanie pliku w trakcie ładowania strony; *metadata* — ładowanie metadanych w trakcie ładowania strony; *none* — plik nie powinien być ładowany w trakcie ładowania strony. Dla *autoplay = true* atrybut *preload* jest ignorowany.



<VIDEO>

```
<video width="200" height="150" autoplay controls poster="vid_poster.jpg">  
  <source src="wildlife.mp4" type="video/mp4" />  
  <source src="wildlife.ogg" type="video/ogg" />  
  Ta przeglądarka nie odtwarza wideo.  
</video>
```

```
<video width="320" height="240" autoplay controls poster="vid_poster.jpg">  
  <source src="wildlife.mp4" type="video/mp4" />  
  <source src="wildlife.ogg" type="video/ogg" />  
  Ta przeglądarka nie odtwarza wideo.  
</video>
```

