

# Technologie tworzenia stron i aplikacji WWW

## wykład 2 – HTML + CSS 3

dr inż. Grzegorz Rogus

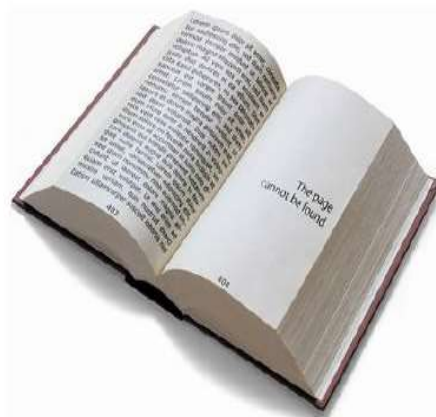


CSS Kaskadowe Arkusze Styli

1

## Budowanie struktury strony

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



2

## Lista znaczników

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

3

## Lista znaczników

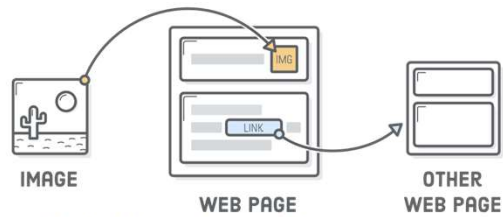
<u>&lt;select&gt;</u>	Lista wybieralna
<u>&lt;small&gt;</u>	Pomniejszony tekst
<u>&lt;span&gt;</u>	Sekcja w dokumencie
<u>&lt;strong&gt;</u>	Tekst silnie wyróżniony
<u>&lt;style&gt;</u>	Definicja arkusza stylu CSS
<u>&lt;sub&gt;</u>	Tekst w dolnym indeksie
<u>&lt;sup&gt;</u>	Tekst w górnym indeksie
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

*i inne ...*

Więcej -> [www.w3schools.com/tags](http://www.w3schools.com/tags)

4

## ZNACZNIKI HIPERTEKSTOWE - uwagi



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

```
<h2 id="sekcja2">First topic</h2>
```

.....

```
<a href="#sekcja2">Idz do sekcji 2 </a>
```

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

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

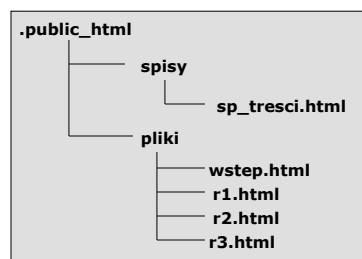
Zamieszczanie wokół `target="_blank"`.

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

5

## 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

6

# Typy elementów HTML

Dowolny element HTML należy do jednej z kategorii:

## 1. **block**: duże bloki zawartości

- Zajmują zawsze całą szerokość viewport (**przepływ wertykalny - top to bottom**)
- Mają wysokość i szerokość height width
- Mogą zawierać inne elementy (block lub inline) jako elementy zagnieżdżone (children)

Przykłady:

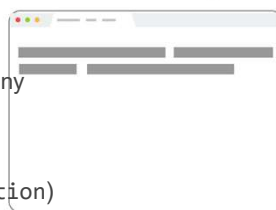
`<p>`, `<h1>`, `<ol>`, `<ul>`, `<table>`, `<div>`

## 2. **inline**: małe ilości treści

Przykłady:

`<a>`, `<em>`, `<strong>`, `<br>`

- Szerokość ustalana według potrzeb (przepływ horyzontalny (from left to right))
- **nie ma** szerokości ani długości height width
- **nie może mieć** elementów zagnieżdżonych ( children)
- **nie może** być pozycjonowany ( np. przez float lub position)



7

## 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.

8

## ELEMENTY BLOKOWE

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

9

## Czym jest formularz

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

10

## 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

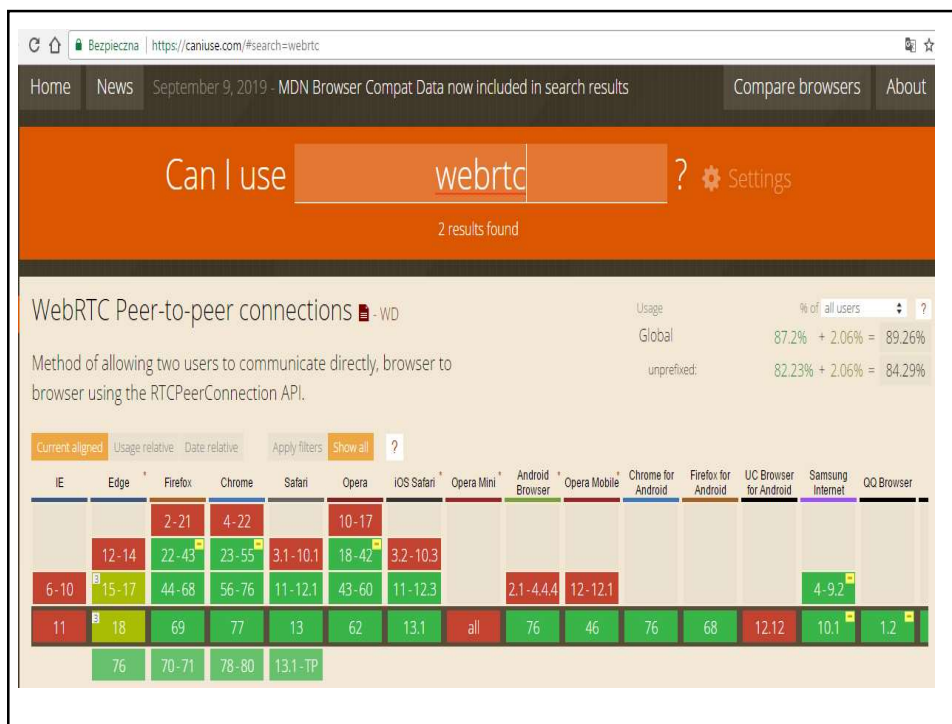
### Dane użytkownika

Imię i Nazwisko	<input type="text" value="Jan Malinowski"/>
Hasło	<input type="password" value="password"/>
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"/> <input type="button" value="v"/>
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"/>	

11



12



13

## 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

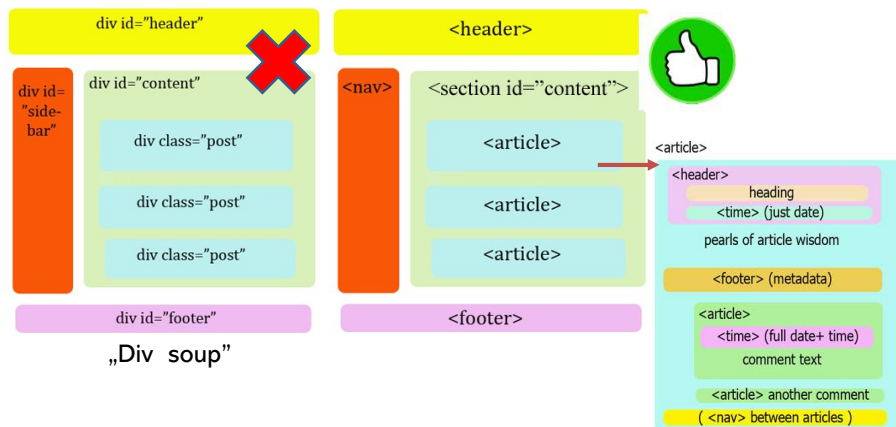
```
function setDuration() {
    var duration = document.getElementById("player").duration;
    document.getElementById("durInfo").innerHTML = "Całość: " + duration.toFixed(2);
    document.getElementById("bar").max = duration;
}
function showInfo() {
    var player = document.getElementById("player");
    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>
```



14



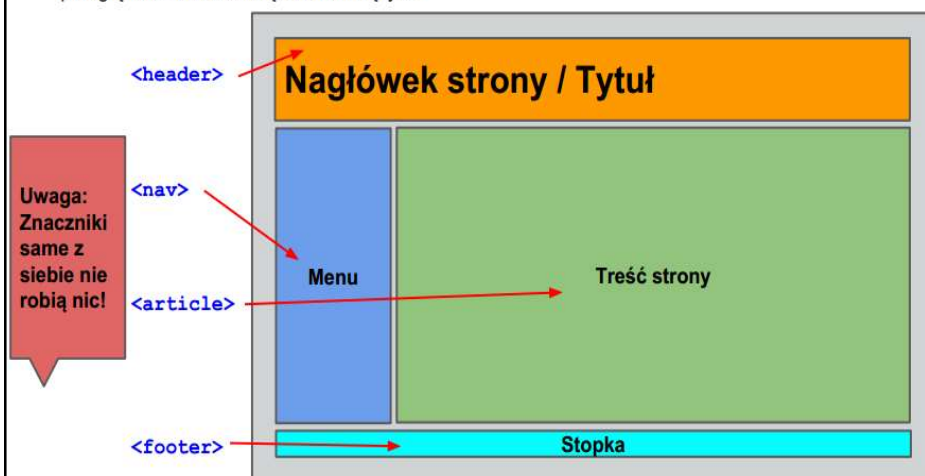
## Struktura strony internetowej



15

## 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 czytelników ekranów ułatwiających przeglądanie stron osobą niedowidzącym.



16



## PRZYKŁADOWE NOWE ELEMENTY W 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



17

## Co nam daje semantyka?

- łatwiej nam znaleźć odpowiedni blok kodu, niż przeszukiwać ścianę divów i spanów
- technologia asystująca lepiej przetworzy structure naszej strony (accessibility)
- SEO - wyszukiwarki lepiej zindeksują zawartość, wiedząc co jest czym

### ZASADA OGÓLNA:

- Znaczniki semantyczne nadają przeznaczenie wybranym treściom na stronie
- Nie należy w nich nadużywać stylów CSS
- Div dalej służą i pozwalają rozmieszczać, układać i stylizować treść witryny


18

## 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.



19

## Nowe znaczniki semantyczne — details i summary

```
...
<section>
  <details>
    <summary>Czytaj więcej...</summary>
    <p>Znaczniki <em>details</em> i <em>summary</em>.</p>
    <p>Aktualnie zaimplementowane jedynie w Chrome.</p>
  </details>
</section>
...
```

Na razie tylko Chrome

► Czytaj więcej...

▼ Czytaj więcej...

Znaczniki `<details>` i `<summary>`.

Aktualnie zaimplementowane jedynie w Chrome.

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>`.

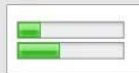
20

## 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

21

## 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



22

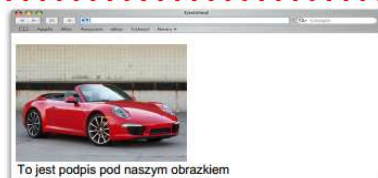
## Nowe znaczniki semantyczne: figure i figurecaption

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.

23

## <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>
```



24

## Nowa semantyka w Web page - podsumowanie



25

## 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

26

## 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

27

## 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.



28



## 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.



29

## <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>
```



30



## Czym jest CSS

- Odpowiada za układ i wygląd strony

- CSS – Kaskadowe arkusze stylów

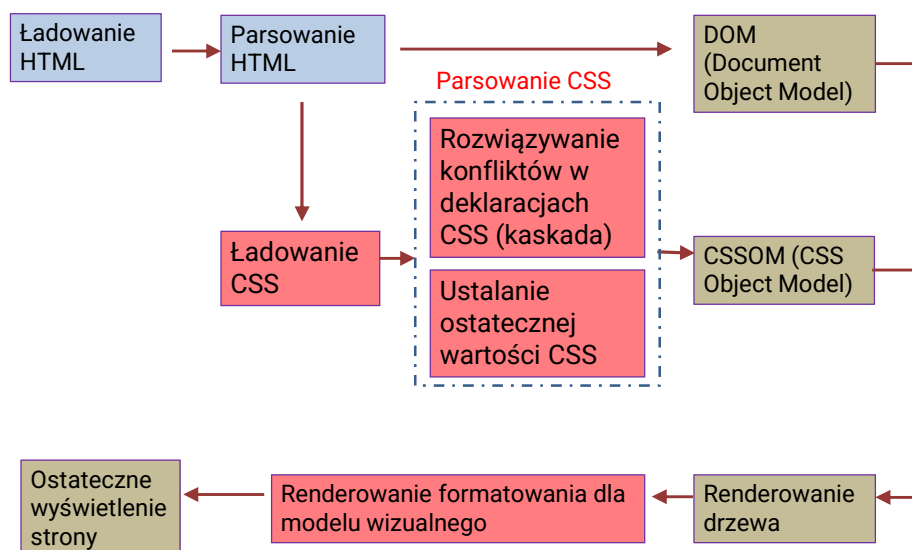
**Kaskadowe** – zawierające hierarchie ważności i definiujące zasady stosowania reguł CSS do danego elementu

**Arkusze** – grupowanie reguł w obrębie dokumentów

**Styl** definiuje wygląd elementów dokumentu HTML, kontrolując ich wizualne cechy w odseparowaniu od kodu HTML

31

## Proces ładowania strony



32

# Jak łączyć?

## Zewnętrzny arkusz stylów

Przykład wykorzystania stylów osadzonych, dokument html:

```
<html>
<head>
  <link rel="stylesheet" type="text/css" href="style.css" />
  . . .
</head>

<body>
  . . .
</body>
</html>
```

Plik definicji stylów style.css:

```
h1 { color: red }
p  { color: navy }
```

33

## By biegle korzystać CSS trzeba...

1. Poznać zasady tworzenie selektorów -> sposobu wskazywania elementów w strukturze HTML

`div.first p` -> **SELEKTOR** – **GDZIE ZASTOSOWAC**

2. Poznać właściwości i wartości, które możemy użyć

`{ font-size: 20px; }` -> **DEKLARACJA WŁAŚCIWOSCI** – **CO ZROBIC**

czyli:

`div.first p { font-size: 20px; }` - **REGUŁA (STYL) CSS**

Szybkość i efektywność przyjdą z praktyka

34

## KASKADOWE ARKUSZE STYLÓW

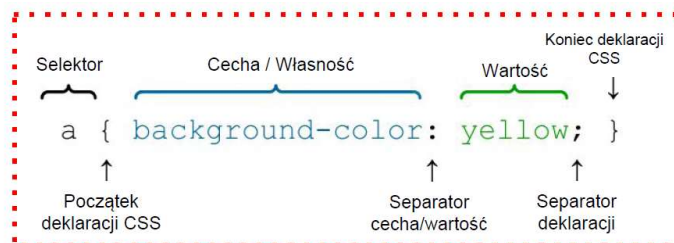
Można w ten sposób opisać wszystkie cechy odpowiedzialne za prezentację elementów dokumentów internetowych, takie jak:

- rodzina czcionek,
- kolor tekstu,
- marginesy,
- odstęp międzywierszowy,
- kolor tła,
- pozycja danego elementu względem innych,
- animacje i przejścia.

**CSS3**

35

## SKŁADNIA ARKUSZY CSS



Przykładowa deklaracja powoduje ustawienie koloru tła dla wszystkich hiperłączy na kolor żółty.



```
<a href="http://www.onet.pl">Onet</a>
```

36

## SKŁADNIA ARKUSZY CSS

**selektor { właściwość: wartość }**

p {color:red;}

ul.nav-tab {display:block; margin:0px;}

Możliwe jest grupowanie selektorów i deklaracji:

**selektor1, selektor2 {**

**właściwość1: wartość1;**

**właściwość2: wartość2;**

**}**

a, a:hover {color:#FA4567; text-decoration:underline }

37

## SKŁADNIA ARKUSZY CSS

Oprócz korzystania bezpośrednio z nazw znaczników HTML, w regułach CSS jako selektorów można używać atrybutów identyfikujących znaczniki HTML.

- Poprzez atrybut **class**:

<h1 **class="text-primary"**>Nagłówek</h1>

<p **class="text-primary"**>Podświetlony akapit</p>

**.text-primary { color: navy; }**

- Poprzez atrybut **id**:

<header **id="site-top"**></header>

**#site-top { display:block; }**

38

## Klasy i identyfikatory

Jak klasy i identyfikatory pomagają w definicji stylu strony ?

Możemy za pomocą klasy lub identyfikatora konkretnie wybrać element(y) dla którego mają być nadane specyficzne własności i cechy.

Selektor klasy:

```
selektor.nazwaklasy { cecha1: wartość;  
                      cecha2: wartość;  
                      ...;  
}
```

↑  
Separator klasy

Selektor identyfikatora:

```
selektor#nazwaklasy { cecha1: wartość;  
                      cecha2: wartość;  
                      ...;  
}
```

↑  
Separator identyfikatora

39

## WARTOŚCI W ARKUSZACH CSS

Właściwości selektorów mogą przyjmować różne wartości:

1. Tekstowe, np.: block, uppercase, solid;
2. Jednostki miar, np.: 2px, 2em, 1rem, 3pt;
  - px – wielkość w pikselach;
  - pt – wielkość w punktach;
  - % - w procentach;
  - em – wielkość względem wysokości czcionki elementu, w którym znajduje się obiekt;
  - rem – wielkość względem wysokości czcionki zadeklarowanej w HTML.

40

## WŁAŚCIWOŚCI SELEKTORÓW

Właściwości selektorów mogą być zapisywane na kilka sposobów:

- border: 2px dashed #ff0000;
- border-width: 2px;  
border-style:solid;  
border-color:#000000;
- border-top: 2px dashed #ff0000;
- border-top-color: #d4d4d4;

41

## CSS (Kaskadowe Arkusze Stylu)

Kaskadowe arkusze stylu dzielimy na trzy grupy ze względu na miejsce wystąpienia:

1. **Wbudowane (inline-style)** - są zapisane w miejscu ich działania, tzn. w znaczniku, któremu mają nadawać specyficzne cechy.
2. **Osadzone (embedded-style)** - są zapisane za pomocą znacznika <style> w sekcji head dokumentu hipertekstowego.
3. **Dołączone (linked-style)** - są zapisane w osobnych plikach o rozszerzeniu .css.

Która z metod zapisywania CSS ma największy priorytet ?

**KASKADOWOŚĆ**

obowiązujący jest styl położony najbliżej elementu

42

## Kaskadowość – metoda rozwiązywania problem redundancji reguł

- Proces łączenia różnych arkuszy stylów i rozwiązywania konfliktów między różnymi regułami CSS i deklaracjami, kiedy więcej niż jedna reguła ma zastosowanie do określonego elementu.

ISTOTNOŚĆ

>

SPECYFICZNOŚĆ

>

KOLEJNOŚĆ

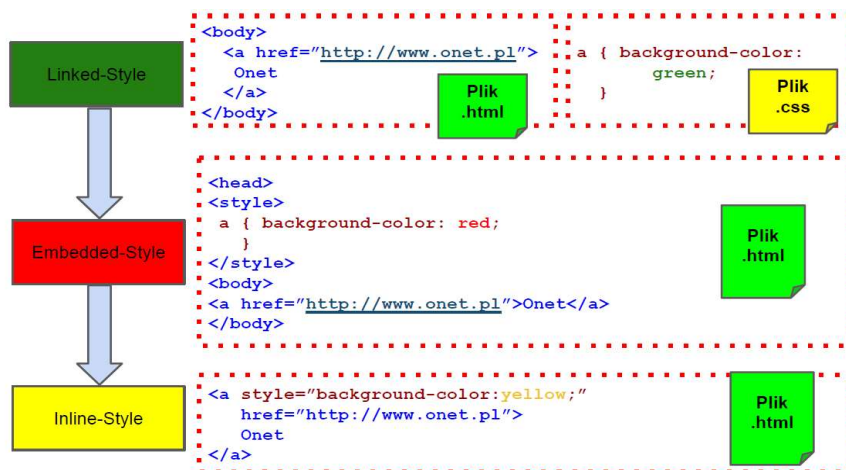
priorytet  
według źródeł  
stylów

Kaskadowość w obrębie  
dokumentu

43

## CSS (Kaskadowe Arkusze Stylu)

priorytet według źródeł stylów



**Uwaga na !important** - łamie zasady kaskadowości

a {background-color: white !important;}

44



## Reguły Kaskady

### ISTOTNOŚĆ



Ten sam dokument

### SPECYFICZNOŚĆ



Ten sam typ selektora

### KOLEJNOŚĆ

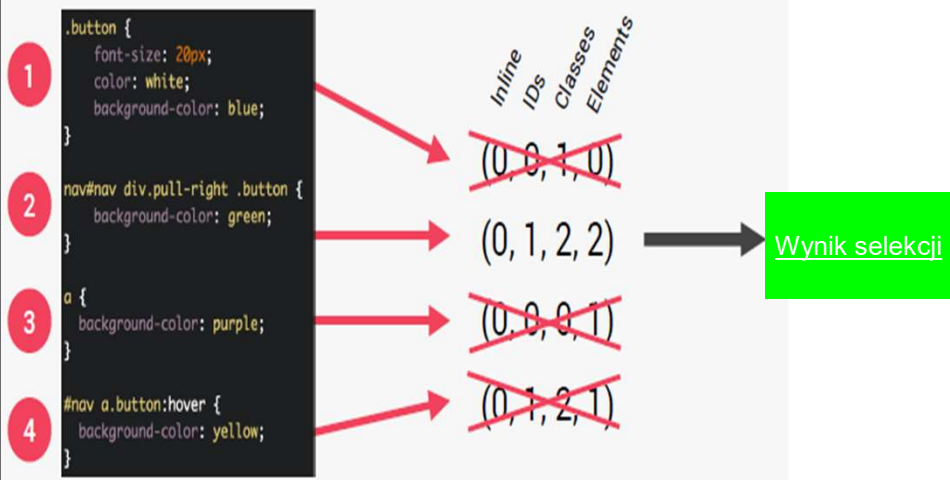
1. deklaracja **!important**
2. Lokalne style inline
3. Wewnętrzny arkusz stylów
4. Importowane style do wewnętrznego arkusza CSS
5. Deklaracja domyślna przeglądarki

A	id selectors	#foo
B	class selectors	bar
	attribute selectors	[title], [colspan="2"]
	pseudo-classes	:hover, :nth-child(2)
C	type selectors	div, li
	pseudo-elements	::before, ::first-letter

Ostatnia deklaracja w kodzie będzie przesłaniała wszystkie inne deklaracje i to ona zostanie zastosowany.

45

## Kaskadowość w obrębie dokumentu



Selektor zawierający 1 identyfikator jest bardziej szczegółowy niż selektor zawierający 1000 klas;  
Selektor zawierający 1 klasę jest bardziej szczegółowy niż selektor zawierający 1000 elementów;  
Selektor uniwersalny \* nie ma wartości specyficzności (0, 0, 0, 0);

46

## Przy okazji – król znaczników div

- Bloki wydzielamy za pomocą znaczników `<div> ... </div>`. Fragment dokumentu objęty blokiem można w swobodny sposób formatować za pomocą stylów.
- Wydzielone bloki mogą zawierać w sobie dowolne inne elementy.
- Dzięki temu nadają się do wydzielania większych, logicznych fragmentów dokumentu i nadawania im specyficznego formatowania za pomocą stylów

### Identyfikacja bloku – specyfikacja klasy bloku

```
.komentarz
{
  font-family: verdana, arial, sans-serif;
  font-weight: bold;
  color: blue;
}

<div class="komentarz">
  <h1>Uwaga</h1>
  <p>
    To jest tekst komentarza
  </p>
</div>
```

### Uwaga

To jest tekst komentarza

47

	Własności		Własności
<b>Fonty</b>	font-family font-size font-weight font-style font-variant text-decoration @font-face	<b>Barwy</b>	color background-color
<b>Wielkość</b>	width height	<b>Przerzeń i odstępy</b>	margin padding margin-left, margin-right, margin-top, margin-bottom padding-left, padding-right, padding-top, padding-bottom
<b>Układ</b>	position left, right float, clear	<b>Obramowanie</b>	border-width border-style border-color border (ustawia grubość, styl i barwę za jednym zamachem)
<b>Grafika</b>	background-image background-repeat background-position	<b>Wyrównanie tekstu</b>	text-align text-indent word-spacing letter-spacing line-height white-space

48

## SELEKTORY CSS – IDENTYFIKACJA

**\*** - spełniony przez każdy element

**E** - spełniony przez każdy element E

```
body { font-family:Verdana; font-size:11px; }
```

**E.value** - spełniony przez te elementy E, które posiadają atrybut CLASS o wartości value

```
p.title1 { font-size:18px; }
```

**E#value** - spełniony przez te elementy E, które posiadają atrybut ID o wartości value

```
img#news { border:2px double #FFFFFF; }
```

49

## SELEKTORY kontekstowe CSS – ZAGNIEŹDŻANIE (selektory potomków)

**E F** - spełniony przez każdy element F zagnieżdżony wewnątrz elementu E

```
p b { color:#FFFFFF; }
```

**E > F** - spełniony przez każdy element F zagnieżdżony bezpośrednio w E

```
ul > li { font-family:Verdana; font-size:11px; }
```

**E + F** - spełniony przez każdy element F, który następuje bezpośrednio za elementem E

```
i + b { color:#000000; }
```

50

## Wykorzystanie selektorów potomków (selektory kontekstowe)

```
div ul { color: blue }

<div>
  <p>Moje ulubione sporty to:</p>
  <ul>
    <li>narciarstwo,</li>
    <li>kolarstwo,</li>
    <li>plywanie.</li>
  </ul>
</div>

<p>Moje ulubione sporty to:</p>
<ul>
  <li>narciarstwo,</li>
  <li>kolarstwo,</li>
  <li>plywanie.</li>
</ul>
```

Moje ulubione sporty to:

- narciarstwo,
- kolarstwo,
- plywanie.

Moje ulubione sporty to:

- narciarstwo,
- kolarstwo,
- plywanie.

```
div.figure p
{
  text-align: center;
  font-size: smaller;
  text-indent: 0;
}

<div class="figure">
  <p></p>
  <p>Kask Breeze firmy Mango</p>
</div>
```



Kask Breeze firmy Mango

51

## Wykorzystanie selektorów potomków (selektory kontekstowe)

### Wykorzystanie selektorów „dzieci”

```
body > p { color: blue }

<body>
  <p>To jest dziecko body</p>
  <div>
    <p>To jest potomek body</p>
  </div>
</body>
```

To jest dziecko body

To jest potomek body

### Wykorzystanie selektorów „braci”

```
p + p
{
  text-indent: 0.7em;
  margin-top : 0
}

<p>Pierwszy paragraf w tym tekście nie musi
posiadać wcięcia akapitowego</p>

<p>Drugi paragraf w tym tekście już powinien
takie wcięcia posiadać</p>



<p>Za wyjątkiem tych paragrafów które następują
po rysunkach</p>
```

Pierwszy paragraf w tym tekście nie musi posiadać wcięcia akapitowego

Drugi paragraf w tym tekście już powinien takie wcięcia posiadać



Za wyjątkiem tych paragrafów które następują po rysunkach

**Tytuł**

52

## SELEKTORY CSS – PSEUDOKLASY

**A:link** - spełniony przez każdy link, który nie został jeszcze odwiedzony

```
a:link { text-decoration:none; }
```

**A:visited** - spełniony przez każdy link, który został odwiedzony

```
a:visited { text-decoration:underline; }
```

**E:hover** - spełniony przez każdy element E, nad którym właśnie znajduje się wskaźnik

```
tr:hover { text-decoration:underline; }
```

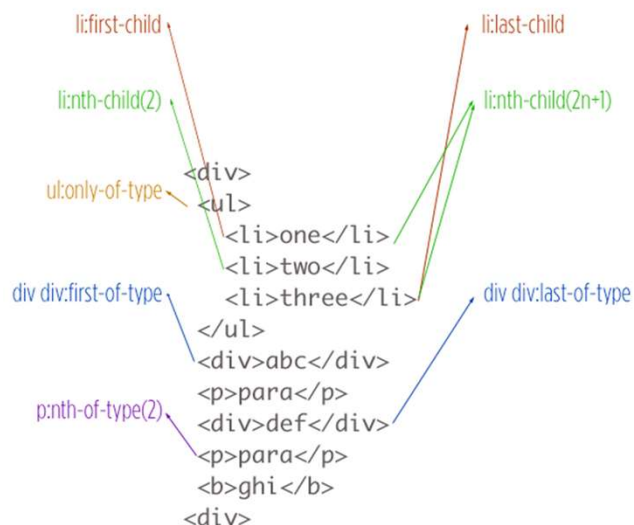
**E:focus** - spełniony przez każdy element E, na którym właśnie znajduje się focus dokumentu

```
input:focus { border-radius:3px; }
```

DEMO -> linki.html

53

## SELEKTORY CSS - PSEUDOELEMENTY



54

## SELEKTORY CSS – KOLEJNOŚĆ

**E:first-child** – spełniony przez pierwszy element E

```
ul li:first-child {color: #f46c00;}
```

**E:last-child** – spełniony przez ostatni element E

```
ul li:last-child {color: #000000;}
```

**E:nth-child(n)** – spełniony przez n-ty element E

```
ul li:nth-child(2) {text-decoration: underline;}
```

**E:nth-child(even)** – spełniony przez wszystkie nieparzyste elementy E

```
ul li:nth-child(even) {text-decoration: underline;}
```

**E:nth-child(odd)** – spełniony przez wszystkie parzyste elementy E

```
ul li:nth-child(odd) {text-decoration: none;}
```

55

## SELEKTORY CSS - PSEUDOELEMENTY

**E::after** - wstawia coś po zawartości elementu E

```
p::after {content: url(smiley.gif) }
```

**E::before** - wstawia coś przed zawartością elementu E

```
p::before {content: url(smiley.gif) }
```

**E::first-letter** - spełniony przez pierwszą literę elementu E

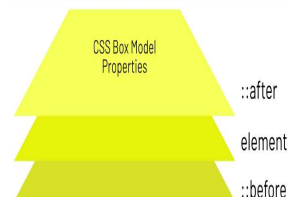
```
p::first-letter {font-size: 2em; }
```

**E::first-line** - spełniony przez pierwszą linię elementu E

```
p::first-line {font-style: italic; }
```

**E::selection** - spełniony przez część obiektu E wybranego przez użytkownika

```
p::selection {background-color: #d4d4d4; }
```



56

## Zmienne w CSS – bez użycia preprocesora

Zmienne w CSS są definiowane za pomocą specjalnej notacji:

```
:root {  
    --primary-color: yellow;  
}
```

Dostęp do zmiennych jest możliwy za pomocą funkcji `var()` :

```
p {  
    color: var(--primary-color)  
}
```

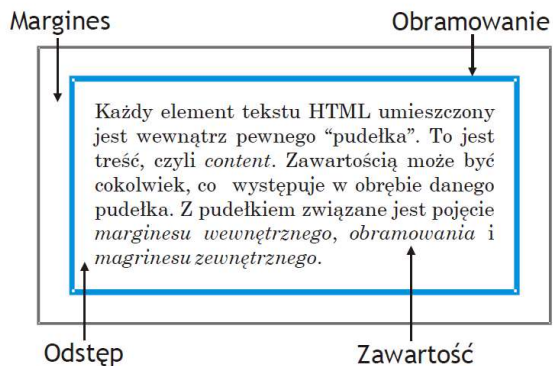
Wartością zmiennych może być dowolna wartość CSS deklarowana w dowolnym selektorze:

```
:header {  
    --default-padding: 30px 30px 20px 20px;  
    --default-color: red;  
    --default-background: #fff;  
}
```

57

## BOX MODEL

Koncepcja modelu pojemnika zakłada, że każdy element dokumentu HTML może być traktowany jako *prostokątny obszar*, którego zawartość otoczona jest *marginem wewnętrznym* (odstępem), *obramowaniem* i *marginem zewnętrznym*.



58