

Witryny Internetowe

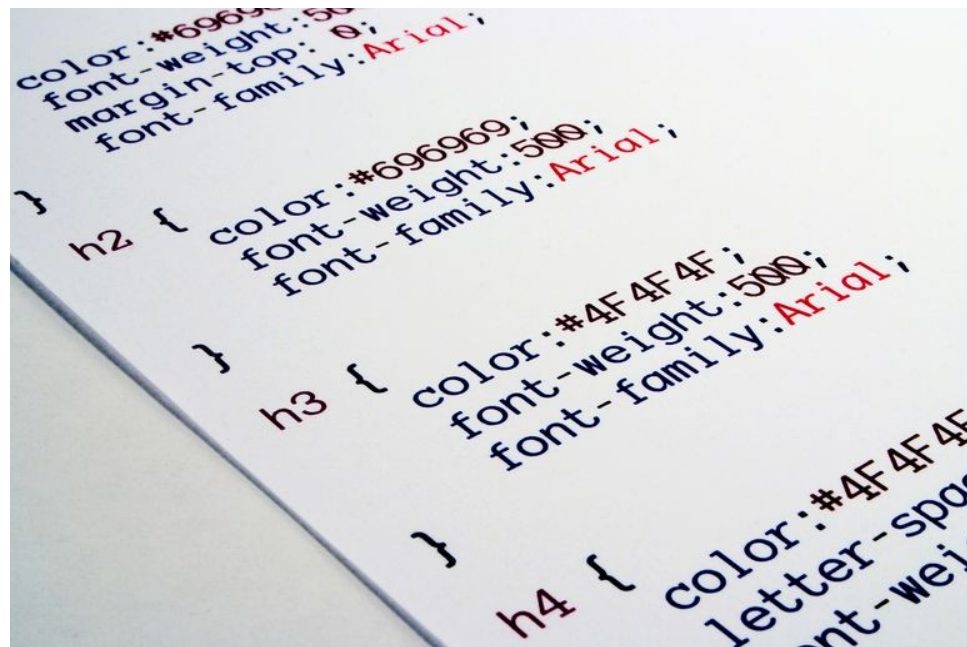
▶ Podstawy języka CSS

- Selektory i pseudoselektory
- Media print
- Responsywność
- Animacje

Opracował: inż. Grzegorz Petri

Przegląd zagadnień

- Powstanie języka CSS
- Struktura stylów
- Osadzanie stylów
 - w dokumencie
 - w pliku zewnętrznym
- Jednostki miary
 - względne
 - bezwzględne
- Podział oraz Opis reguł CSS

A close-up photograph of a piece of paper with CSS code written on it. The code is color-coded: selectors like 'h2', 'h3', and 'h4' are in red, and property names like 'color', 'font-weight', 'margin-top', and 'font-family' are in blue. The values are in black. The code defines styles for heading elements, including color (hex codes), font weight (500), margin-top (0), and font family (Arial).

```
color: #696969;  
font-weight: 500;  
margin-top: 0;  
font-family: Arial;  
  
h2 {  
  color: #696969;  
  font-weight: 500;  
  font-family: Arial;  
}  
  
h3 {  
  color: #4F4F4F;  
  font-weight: 500;  
  font-family: Arial;  
}  
  
h4 {  
  color: #4F4F4F;  
  letter-spacing: 0.1em;  
  font-weight: 500;  
}
```

Kaskadowe arkusze stylów

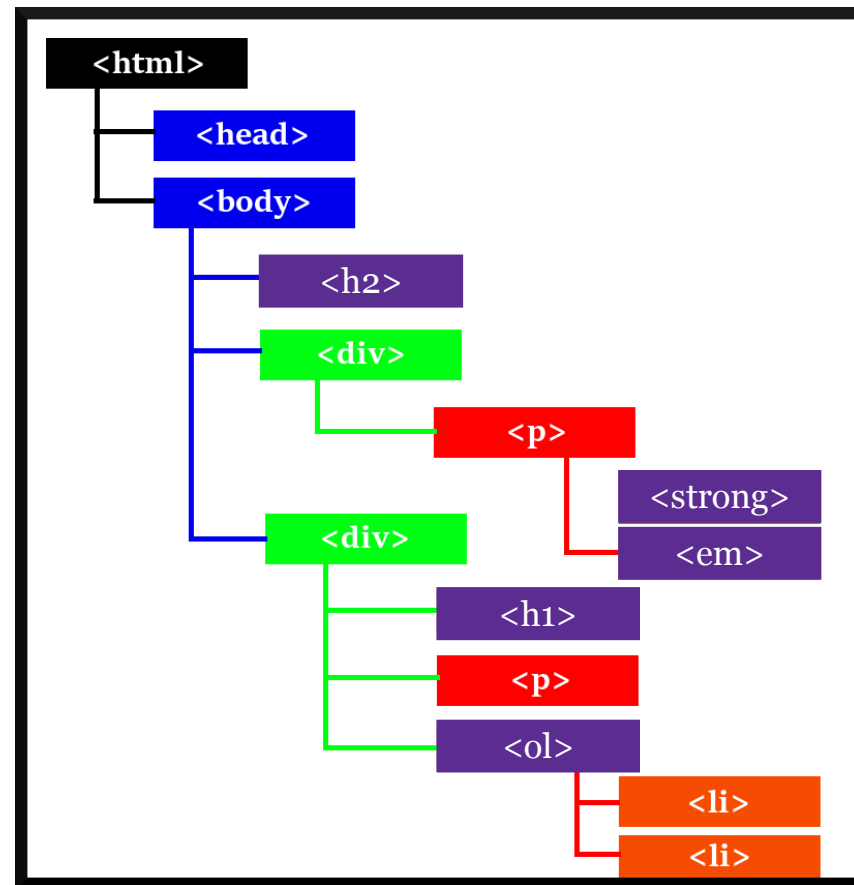
Krótką historia Cascading Style Sheets

- ❖ Język służący do opisu **formy prezentacji** (wyświetlania) stron WWW
- ❖ Pierwszy szkic CSS-a zaproponował Håkon Wium Lie w 1994, który to zakładał współpracę tylko z HTML, skąd wczesna nazwa CHSS
- ❖ Arkusze stylów były odpowiedzią na **niekompatybilność** wyświetlanych w różnych przeglądarkach dokumentów HTML (**własnościowe tagi HTML**)
- ❖ Został opracowany przez organizację **W3C** jako **standard** w 1996 roku
- ❖ **Uniwersalność** zastosowania stylów CHSS dla tagów nie związanych z językiem HTML umożliwiła zastosowanie ich do innych języków opartych o tagi, np. **XML, XSL, czy XSLT** (lub dowolny Markup Language, dla którego renderer może zastosować CSS)

Właściwości języka CSS

Co sprawia, że CSS jest potężnym narzędziem

- Kaskadowość, czyli możliwość dziedziczenia ustawień elementów podrzędnych (dzieci: **LI**) z elementów nadrzędnych (rodziców: **BODY, OL**)
- Selektywność stosowania stylów do **wybranych elementów** lub grup tych samych elementów (**DIV, P, LI**)
- Uniwersalność (HTML, XML, XSLT) i powtórne użycie w innych plikach



Struktura i budowa reguł

Anatomia arkusza i reguł

- Arkusz CSS podzielony jest na fragmenty złożone z **selektorów** oraz bloku **deklaracji** ograniczonej nawiasami klamrowymi (znaki { oraz }).
- Selektor to nazwa tagu HTML (**BODY**), Klasy (**.pen**) lub Identyfikatora (**#myid**).
- Blok deklaracji składa się z **reguł** oddzielonych średnikami (znak ;)
- Pojedyncza reguła składa się z dwóch części: **właściwość (color)** i **wartość (red)**.

```
SELEKTOR { DEKLARACJA }
```

```
SELEKTOR { reguła1; reguła2; regułaN; }
```

```
SELEKTOR { właściwość: wartość; }
```

```
BODY { background-color: yellow; }
```

```
.pen { color: red; }
```

Osadzanie stylów

Czyli co, gdzie, jak...

Style mogą być umieszczane w dokumencie na kilka sposobów:

- **In-line** - Kody formatujące są umieszczane w środku tekstu w dokumencie i dotyczą wybranego znacznika dokumentu.
- **<STYLE>** w znaczniku **<head>** - Kody formatujące są umieszczane na początku dokumentu i oddziałują na całą stronę HTML.
- **<LINK ...>** w znaczniku **<head>** - Kody formatujące są umieszczane w zewnętrznym pliku wspólnym dla wszystkich dokumentów tworzących Witrynę, do których zostaną dołączone tą dyrektywą.
- **@import** - Kody formatujące są importowane z pliku z innej strony lub innego dokumentu. Ten sposób pozwala również podzielić własne rozbudowane arkusze na mniejsze części w postaci funkcyjnych modułów

In-Line CSS

Styl dla znacznika

- Styl liniowy, o najmniejszej elastyczności oraz przenośności
- Kod CSS osadzony w dokumencie HTML w kodzie znaczników
- Nadpisuje inne reguły mogące mieć zastosowanie dla danego elementu
- Przykład:

```
<CITE style="color: red;">I've got bad feelings about this.</CITE>
```

Znacznik STYLE dla CSS

Znacznik dla kodu CSS

- Lepsze podejście niż IN-LINE - większa elastyczności i przenośności
- Kod CSS osadzony w dokumencie lecz oddzielony od kodu znaczników
- Osadzenie jak i wprowadzanie zmian w KAŻDYM dokumencie oddzielnie
- Przykład:

```
<HEAD>
  <STYLE type="text/css">
    P { color: red; }
  </STYLE>
</HEAD>
<BODY>
  <P>This PEN is RED</P>
</BODY>
```


LINK do CSS

Zewnętrzny plik arkusza

- Oddzielenie warstwy danych od warstwy prezentacji – treść opisu może być zmieniana bez modyfikacji źródłowego dokumentu HTML
- Pliki CSS mogą (powinny) być współdzielone przez kilka dokumentów
- Wprowadzanie zmian odbywa się tylko w jednym pliku
- **Przykład:**

Powiązanie pomiędzy dokumentami

Adres URL i nazwa pliku

Typ zawartości pliku

```
<HEAD>
<LINK rel="stylesheet" href="gui/arkusz.css" type="text/css">
</HEAD>
<BODY>
  <P>A ja ci powiadam, że nie rozpoznałbyś strategii nawet wtedy, gdyby wyskoczyła
  z krzaków i kopnęła cię w dupę.</P>
</BODY>
```

Znacznik **IMPORT** dla CSS

Importowanie arkusza do arkusza

- Ten sposób pozwala również podzielić własne rozbudowane arkusze na mniejsze części w postaci funkcyjnych modułów
- **Przykład:**

```
<HEAD>
  <STYLE type="text/css">
    P { color: red; }
    @import url('inny_arkusz.css');
  </STYLE>
</HEAD>
```

Jednostki miar stosowane w CSS

Miary wirtualne dla elementów –
jednostki względne:

ID	Jednostka	Opis
%	procent	Procentowy rozmiar elementu rodzica
PX	piksel	Rozmiar w PX względem dostępnej rozdzielczości
EM	proporcja	proporcje wysokości, do czcionki elementu rodzica
REM	proporcja względem HTML	proporcje wysokości, do czcionki elementu root
EX	proporcja	Proporcja do wysokości litery

Miary fizyczne (układu SI), odnoszące się do świata rzeczywistego - jednostki bezwzględne:

ID	Jednostka	Opis
PT	punkty	wywodzą się z typografii (standardowa jednostka dla czcionek).
IN	cale	jednostka wywodzi się z amerykańskiego systemu miarowego, gdzie jest głównie używana
MM	milimetry	pochodzą z systemu metrycznego
CM	centymetry	miara stosowana w większości krajów na świecie

Podział reguł CSS

Umowne pogrupowanie zastosowań reguł

- ◆ Formatowanie tekstu (*wyrównanie, kontrola odstępów, zmiany wizualne*)
- ◆ Formatowanie czcionki
- ◆ Listy wypunktowane
- ◆ Kolory i tło elementów (*wybór koloru, pozycjonowanie*)
- ◆ Marginesy i odstępy
- ◆ Elementy blokowe (*wyświetlanie pływające, rozmiar, pozycjonowanie*)



Zwróć uwagę, element/zachowanie nietypowe

Formatowanie tekstu

Wyrównanie tekstu

- Wyrównanie w poziomie (tekstu lub elementu blokowego):

`text-align: wartość;`

- Możliwe wartości:

`left | right | center | justify`

(lewo, prawo, środek, do lewej i prawej)

- Wyrównanie w pionie (tekstu w elementach blokowych):

`vertical-align: wartość;`



- Możliwe wartości:

`top | middle | bottom | sub | super`

(góra, środek, dół, indeksy: dolny, górny)

Formatowanie tekstu

Kontrola odstępów

- Wcięcie tekstu w elemencie blokowym:

`text-indent: wartość_liczbowa JEDNOSTKA;`

- Odstępy między wierszami:

`line-height: wartość_liczbowa JEDNOSTKA;`

- Odstępy między wyrazami:

`word-spacing: wartość_liczbowa JEDNOSTKA;`

- Odstępy między literami:

`letter-spacing: wartość_liczbowa JEDNOSTKA;`

- Możliwe jednostki:

`% | px | em | ex | pt | cm | mm | in`

Formatowanie tekstu

Wizualne zmiany tekstu

- Dekoracja tekstu:

`text-decoration: wartość;`

- Możliwe wartości:

`none | overline | line-through | underline`

(brak, nadkreślenie, przekreślenie, podkreślenie)

- Transformacja tekstu:

`text-transform: wartość;`

- Możliwe wartości:

`none | uppercase | lowercase | capitalize`

(bez zmian, wielkie litery, małe litery, pierwsza litera słowa duża)

- Traktowanie odstępów między wyrazami:

`white-space: normal | pre | nowrap;`

(domyślnie, tam, nie tam)

Formatowanie czcionki

Gdy Comic Sans był przestępcą

- Rodzina czcionki:

```
font-family: NAZWA_KROJU_CZCIONKI_W_SYSTEMIE;
```



- Rozmiar czcionki:

```
font-size: wartość_liczbowa JEDNOSTKA;
```

- Styl czcionki:

```
font-style: normal | italic | oblique;  
(normalna, kursywa, ?)
```

- Waga czcionki:

```
font-weight: 100 | 200 | ... | 800 | 900;
```

- Warianc czcionki:

```
font-variant: normal | small-caps;  
(normalny, kapitaliki)
```


Listy wypunktowane

Listy są przydatne

- Styl wypunktowania listy:

`list-style-type: wartość;`

- Możliwe wartości dla listy niesortowanej:

`disc | circle | square`
(pełen okrąg, okrąg, pełen kwadrat)

- Możliwe wartości dla listy sortowanej:

`none | decimal | lower-roman | upper-roman | lower-alpha | upper-alpha`
(brak; 1,2,3; i,ii,iii; I,II,III; a,b,c; A,B,C)

- Własny punktory:

`list-style-image: url („punktory.jpg”);`

- Pozycjonowanie listy względem punktora (różne przeglądarki, różny efekt):

`list-style-position: inside | outside;`
(wewnątrz, zewnątrz)

Kolory i tło elementów

Wskazanie koloru

- Kolor tekstu:

`color: wartość;`

- Kolor tła elementu:

`background-color: wartość;`

- Możliwe sposoby zapisywania wartości koloru:

- Nazwa koloru wg specyfikacji W3C: **yellow**, **black**, **red**, **green**, **blue**;
- Wartość szesnastkowa (*tu kolor czarny*): `#000000`;
- Wartość **RGB** (*tu kolor biały*): `rgb(255, 255, 255)` ;
- Wartość **RGBA** (+przezroczystość): `rgba(255, 255, 255, 0.5)` ;

- Obraz jako tło elementu:

`background-image: url("obraz.jpg");`



Kolory i tło elementów

Pozycjonowanie

- Kontrola powtarzania obrazu tła:

`background-repeat: wartość;`

- Możliwe wartości:

`no-repeat | repeat-x | repeat-y | repeat;`

(nie powtarzaj, powtarzaj poziomo, powtarzaj pionowo, powtarzaj)

- Kontrola pozycji obrazu tła

`background-position: wartość;`

- Możliwe wartości:

`left | right | center | top | bottom | wartość_liczbowa`

(lewo, prawo, wycentruj, góra, dół, wartość liczbowa, np. top: 10px)

- Przywiązanie obrazu tła:

`background-attachment: scroll | fixed;`

Marginesy i odstępy

Kontrola elementów blokowych lub liniowych

- Odstęp innego elementu od kontrolowanego elementu (margin):

```
margin: wartość1 wartość2 wartość3 wartość4;  
margin-top: wartość1;  
margin-right: wartość2;  
margin-bottom: wartość3;  
margin-left: wartość4;
```

- Odstęp elementu kontrolowanego od jego zawartości (padding):

```
padding: wartość1 wartość2 wartość3 wartość4;  
padding-top: wartość1;  
padding-right: wartość2;  
padding-bottom: wartość3;  
padding-left: wartość4;
```

W różnych przeglądarkach istnieją różnice w interpretacji wartości obu cech – inaczej są wyliczane procentowe zależności np. w szerokości elementu wewnętrznego względem zewnętrznego.

Elementy blokowe

Wyświetlanie i pływanie

- To elementy, które domyślnie zajmują cały dostępny obszar (szerokość) oraz których szerokość i wysokość oraz pozycjonowanie można w pełni kontrolować

- Sposób wyświetlania elementu:

`display: none | block | inline | list-item;`

(nie wyświetlaj, jako blok, w linii, jako lista)

- Umieszczenie elementu:

`float: none | left | right;`

(brak, do lewej, do prawej)

- Wyczyszczenie miejsca elementu:

`clear: none | left | right | both | all;`

(żaden, lewy, prawy, oba, wszystkie elementy)

Elementy blokowe

Rozmiar elementów

- **Wysokość elementu:**

`height: wartość_liczbowa JEDNOSTKA;`

- **Szerokość elementu:**

`width: wartość_liczbowa JEDNOSTKA;`

- **Możliwe jednostki:**

`% | px | em | ex | pt | cm | mm | in`

- **To NIE jest element blokowy:**



`Właściwości CSS
dopasowane do tekstu – width i height nie zadziała.`

Elementy blokowe

Pozycjonowanie elementów

- **Typ pozycjonowanie elementów:**

`position: absolute | relative | static | fixed;`
(absolutne wg okna, od bieżącej pozycji, statyczne, element nieruchomy - przewijana tylko reszta strony)

- **Pozycja elementu:**

`left: wartość_liczbowa JEDNOSTKA;`
`top: wartość_liczbowa JEDNOSTKA;`
`bottom: wartość_liczbowa JEDNOSTKA;`
`right: wartość_liczbowa JEDNOSTKA;`

- **Możliwe jednostki:**

`% | px | em | ex | pt | cm | mm | in`

- **Pozycja w stosie:**

`z-index: wartość_liczbowa;`

Wartość „1” (mniejsza) – na wierzchu, wartość „2” (większa) – pod spodem



Zadania dodatkowe

Podstawowe (obowiązkowe)

Zaawansowane (na ocenę)

Zadanie dodatkowe

Podstawowe

1. Użyj właściwości CSS, by sformatować wygląd dokumentu HTML
2. Przygotuj drugi szablon dla tego samego dokumentu HTML, tym razem spełniający wytyczne standardu WCAG.

Zadanie dodatkowe

Zaawansowane

1. Wypisz w tabeli jednostki miar niewyszczególnione w prezentacji.
2. Utwórz plik arkusza CSS składający się z modułów załączanych do głównego pliku arkusza poprzez mechanizm @import :
 1. Podziel moduły na: układy wyświetlania, treść, formularze, tabele
 2. Przygotuj trzy schematy kolorystyczne CSS na bazie pkt 1.1
3. Przygotuj przykładowy dokument HTML z arkuszem CSS realizujący założenia standardu WCAG poprzez skonstruowanie dokumentu używającego jednostek miar opartych na jednostce miary REM i EM.

```
CITE {  
  content: "Pytania?";  
}
```