

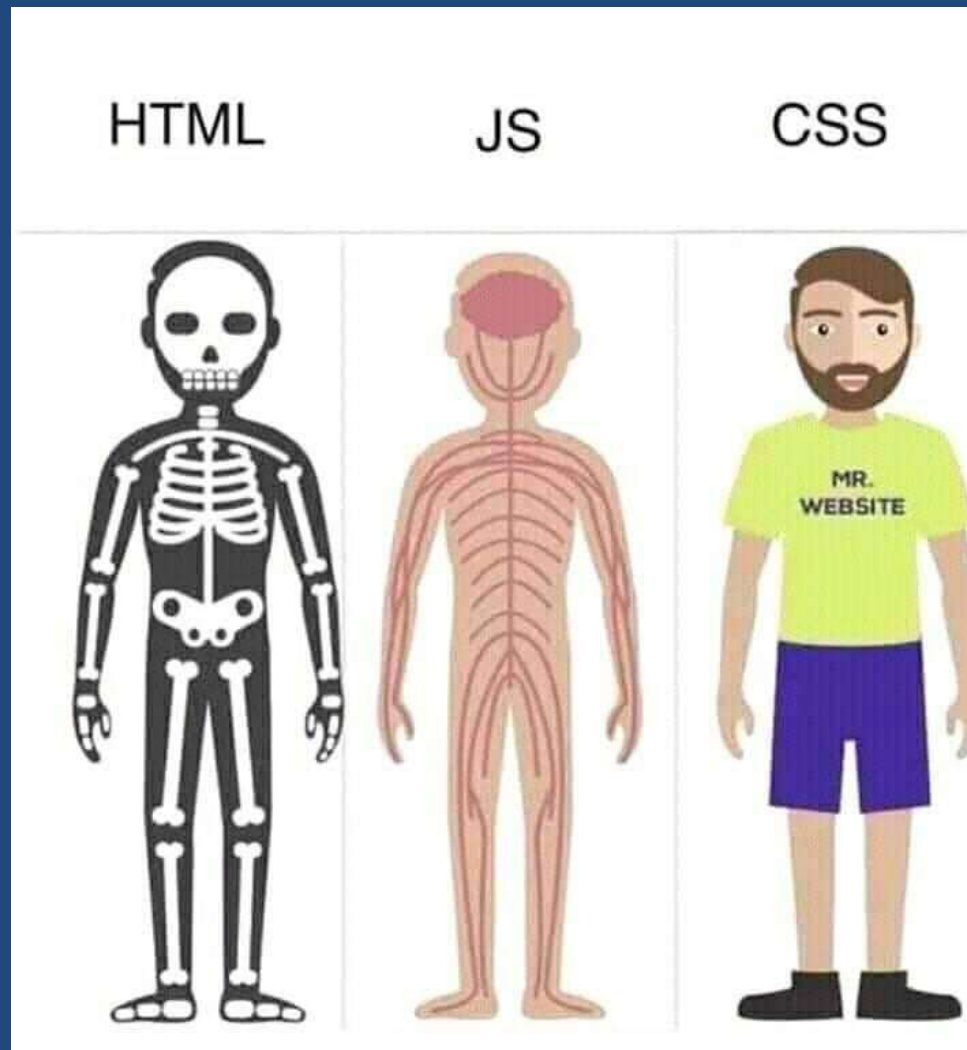


# Cascading Style Sheets

Czyli po naszymu

Kaskadowe Arkusze Styli

# CSS to układ strony i jej wygląd



# Krótką (?) Historia CSS

Historia CSS (*ang. Cascading Style Sheets*) zaczyna się w listopadzie 1994 roku, gdzie Hakon Wium Lie na konferencji dotyczącej technologii internetowych w Chicago zaprezentował pierwszy szkic projektu kaskadowych arkuszy stylów zwany Cascading HTML Style Sheets. Niestety, przedstawiony przez niego projekt wywołał wiele burzliwych dyskusji. Rozwiązanie było bardzo uproszczone i wielu przeciwników zarzucało projektowi zbyt dużą prostotę i małą użyteczność. Oponenci poszukiwali języka w pełni programowalnego, a nie prostego arkusza stylów.

W sierpniu 1996 roku kaskadowe arkusze stylów CSS (nie będąc jeszcze rekomendowanym produktem przez organizację W3C – *World Wide Web Consortium*) zostały pierwszy raz zaimplementowane w komercyjnej przeglądarce Microsoft Internet Explorer 3. Cztery miesiące później, w grudniu 1996 roku, CSS1 został oficjalnie zatwierdzonym i rekomendowanym standardem. Uzyskana rekomendacja poskutkowała znacznym i gwałtownym rozwojem technologii. Nie minęło zaledwie półtora roku, a w maju 1998 roku została opublikowana i ponownie oficjalnie zarekomendowana przez W3C kolejna wersja tego języka o nazwie kodowej CSS2.

# Krótką (?) Historia CSS...cd

Drugą przeglądarką (po Microsoft Internet Explorer 3), która zdecydowała się zaimplementować rozwiązania CSS do swojego projektu była Netscape's Navigator 4.0. Niestety, stworzony produkt stał się bardzo niedoskonały. Programiści nie poświęcili wystarczająco dużo czasu na przetestowanie swojego produktu, czego konsekwencją była długa lista nieobsługiwanych właściwości CSS oraz znaczna liczba niezadowolonych użytkowników.

Trzecią przeglądarką wspierającą technologię CSS została w listopadzie 1998 roku Opera. Nowa wersja przeglądarki, w porównaniu do Netscape's Navigator 4.0 wspierała znaczną większość dostępnych funkcji technologii CSS1.

Kiedy w maju 1998 roku została opublikowana nowa wersja języka CSS o nazwie CSS2, twórcy technologii od razu przeszli do pracy nad jej jeszcze ulepszoną wersją CSS3 oraz w międzyczasie zaczęli opracowywać poprawioną wersję języka CSS2 o nazwie kodowej CSS2.1. Natomiast programiści przeglądarek internetowych ponownie rozpoczęli żmudny proces implementacji nowego standardu w swoich produktach.

# Krótką (?) Historia CSS...end

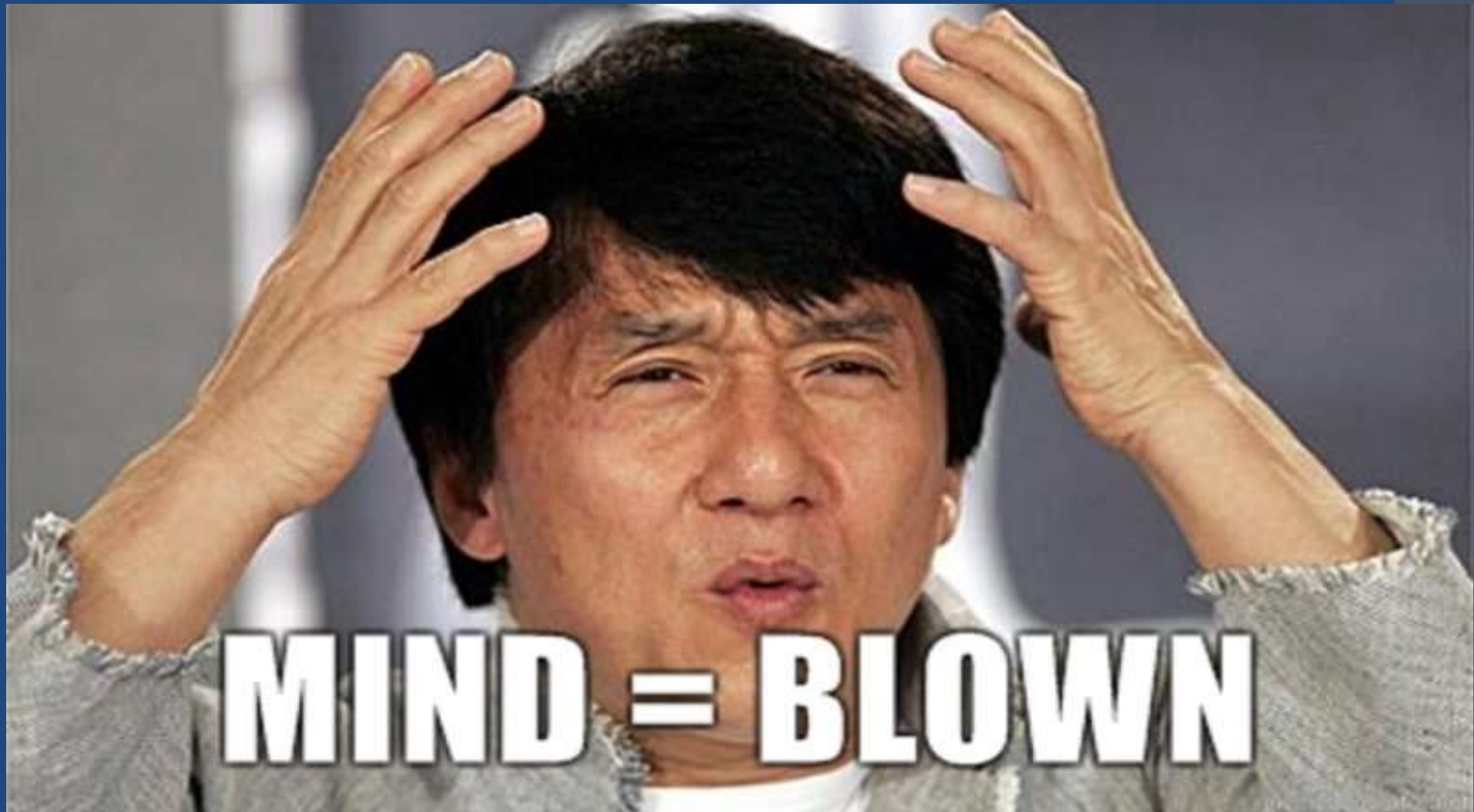
Nowa wersja CSS3 była systematycznie publikowana jako seria teoretycznych, autonomicznych modułów, które razem tworzyły jedną monolityczną specyfikację technologii. W 2012 roku, trzy główne moduły języka CSS (Color Level 3, CSS Namespaces, Selectors Level 3), połączone wraz z poprzednimi modułami CSS1 i CSS2.1 osiągnęły status rekomendowanej przez konsorcjum W3C technologii. Język CSS3 (będący w rzeczywistości wieloma różnymi modułami) nigdy nie uzyskał własnej (połączonej, jednolitej) specyfikacji, jak to było w przypadku jego poprzednich wersji.

Aktualnie trwają prace nad kolejną wersją języka o sugerowanej nazwie kodowej CSS4, która ponownie jak jej poprzedniczka zbudowana będzie z różnych modułów takich, jak już opublikowana specyfikacja Selectors Level 4.

Źródło: <https://danielpietrasik.pl/historia-css/>



Ufff ... za dużo „Lorem Ipsum”  
przejdźmy do konkretów



# Kaskadowość w CSS ..co to takiego?

- Kaskadowość określa, iż w pierwszej kolejności brane są pod uwagę style pochodzące z arkusza zewnętrznego( oddzielnego pliku o rozszerzeniu **.css**). Te z kolei mogą zostać nadpisane przez style zdefiniowane w arkuszu zagnieżdżonym. Style zdefiniowane 'inline' znajdują się na końcu tego "łańcuszka" jako znajdujące się najbliżej opisywanego elementu strony.

# Składnia/Reguła CSS

**selektor { właściwość : wartość; }**

Reguła CSS składa się z:

- **selektora** - służy on do wyszukania elementu bądź grupy elementów w danym dokumencie HTML, które mają otrzymać dany styl CSS, czyli dane właściwości CSS, które odpowiadają za zmianę wyglądu odszukanych elementów HTML. Przykładowo, selektor `p > span` odszuka wszystkie elementy `span` umieszczone w elemencie `p`.
- **nawiasu klamrowego { }**, który występuje zaraz po selektorze i w którym należy umieścić właściwości CSS jakie mają otrzymać dane elementy HTML, wyselekcjonowane przez dany selektor.
- **właściwości CSS** - za pomocą nich możemy określić co chcemy zmienić w wyglądzie danego elementu HTML, umieszczamy je w nawiasie klamrowym. Przykładowo, właściwość `background-color` określa kolor tła elementu HTML.
- **wartości właściwości CSS** - występują po właściwości, są podawane po dwukropku i określają w jaki sposób chcemy zmienić daną właściwość wyglądu elementu HTML. Na końcu wartości należy postawić średnik ; Przykładowo, wartość `lightblue` określa kolor jasnoniebieski.
- Ponadto, jeżeli dany styl CSS ma zawierać więcej niż jeden selektor, to kolejne selektory należy oddzielić od siebie przecinkiem ,



# Składnia/Reguła CSS...cd

jeżeli dany styl CSS ma zawierać więcej niż jeden selektor, to kolejne selektory należy oddzielić od siebie przecinkiem

**selektor, selektor**

**{**

**właściwość:wartość;**

**właściwość:wartość;**

**właściwość:wartość;**

**/\* tak oznaczamy komentarz widoczny tylko w pliku . CSS \*/**

**}**

# Metody osadzenia CSS na stronie czyli w kodzie HTML

- Metoda inline (Polega na bezpośrednim przypisaniu stylu do elementu HTML poprzez atrybut **style**)
- CSS Zagnieżdżony (Jest to arkusz umieszczony wewnątrz znacznika `style` w części nagłówkowej strony (wewnątrz znacznika `<head>`).)
- Zewnętrzny arkusz CSS (jest to plik tekstowy z rozszerzeniem `.css` znajdujący się najczęściej w tym samym folderze co plik html dołączany w nagłówku strony )

Enough talk ...let's do smtn



# Metoda inline:

```
<p style="color:blue;"> Hello world </p>
```

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
<meta charset="UTF-8">  
<meta name="viewport" content="width=device-width, initial-scale=1.0">  
<meta http-equiv="X-UA-Compatible" content="ie=edge">  
<title>Document</title>  
</head>  
<body>
```

```
<p style="color:red; font-weight:bold;text-align: center; ">  
Hello world </p>
```

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

# Css zagnieżdżony:

```
<head>
```

```
  <style type="text/css">
```

```
    p {
```

```
      font-weight:bold;
```

```
      color:blue;
```

```
    }
```

```
  </style>
```

```
</head>
```

# Css zagnieżdżony:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>

  <style type="text/css"> <!-- początek stylu zagnieżdżonego... -->
    p {
      font-weight:bold;
      color:blue;
    }
  </style> <!-- ...a tu jego koniec -->

</head>

<body>
  <p style="color:red;text-align: center; "> Hello world </p>
</body>
</html>
```



# Zewnętrzny arkusz CSS

Najbardziej zalecany.!!

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

Polega na umieszczeniu definicji stylów w oddzielnym pliku o rozszerzeniu **.css**, a w dokumencie go używającym umieszczeniu instrukcji importującej w części nagłówkowej strony **<head>**

W HTML arkusz zewnętrzny jest dołączany przez element **<link>**. Posiada on trzy atrybuty: **rel** informuje przeglądarkę o rodzaju linkowanego zasobu, **type** wskazuje rodzaj arkusza stylu. Atrybut **href** zawiera adres URL arkusza.

# Zewnętrzny arkusz CSS

```
<head>
```

```
<title>Kochamy CSS</title>
```

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

```
</head>
```

# Selektory css

- ⦿ Selektor **id elementu** html
- ⦿ Selektor **klasy** elementu html

# Klasy css

Klasa to zbiór cech które możemy wykorzystywać wielokrotnie dla różnych elementów html

```
<div class="blue">
```

Odwołanie do klasy w css stosujemy za pomocą kropki

```
.blue { background-color: blue; }
```

# Identyfikator id

Do identyfikacji poszczególnych elementów używamy tzw. identyfikatorów w skrócie id. Id to atrybuty elementów html, które pozwalają je odróżniać. Zapisujemy je skrótem id i przypisujemy do nich wartości tak jak do innych atrybutów, używając do tego znaku = oraz " ".

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

Selektory **id** uzyskujemy poprzez dodanie znaku # przed nazwą id, np. (#header). Jeśli chcielibyśmy dodać kolor tła do naszej sekcji header moglibyśmy tak tego dokonać.

```
#header { background-color: blue; }
```

# Łączenie

Oczywiście nic nie stoi na przeszkodzie  
żeby łączyć obie te rzeczy w kodzie html

```
<div id="header" class="blue"></div>
```

Albo

```
<div id="header" class="blue klasa1 klasa2"  
></div>
```



Enough talk ...let's do smtn



# Ale najpierw IDE

- **Zintegrowane środowisko programistyczne, IDE** (od ang. *integrated development environment*) – program lub zespół programów (środowisko) służących do tworzenia, modyfikowania, testowania i konserwacji oprogramowania....czyli u nas to Visual Studio Code

# Ale najpierw .... EMMET!

**Emmet czyli jak  
przyspieszyć  
pisanie kodu HTML**

*zadanie #06, dzień 7*



P R O D U K T Y W N O Ś Ć

# Co to emmet i co nam daje

- ◎ <https://czterytygodnie.pl/emmet-czyli-jak-przyspieszyc-pisanie-kodu-html/>
- ◎ <https://docs.emmet.io/cheat-sheet/>

Enough talk ...let's do smtn



# Projekt pierwszej strony





# Szkielet html

```
<body>
```

```
  <div id="container">
```

```
    <div id="logo"></div>
```

```
    <div id="menu"></div>
```

```
    <div id="topbar"></div>
```

```
    <div id="sidebar"></div>
```

```
    <div id="content"></div>
```

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

```
  </div>
```

```
</body>
```

# Czyli w emmecie było by tak

Każda instrukcja w Emmecie jest zakończona tabulatorem... jeśli coś nie zadziało tak jak byśmy oczekiwali sprawdzmy składnię instrukcję

- ⦿ `div#container(tabulator)`
- ⦿ `div#container>div#logo+div#menu...ect(  
tab)`

# Logo:

- ⦿ `<div id="logo">Poznaj Linuxa</div>`

