

# Aplikacje webowe



## CSS3 – krótkie wprowadzenie

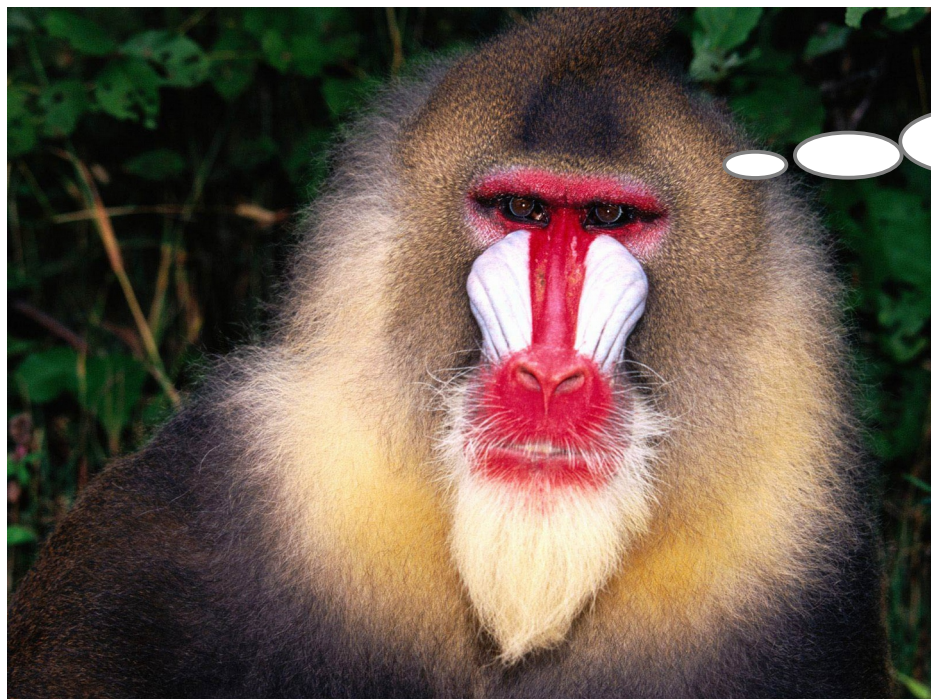
Aleksander Lamża  
ZKSB · Instytut Informatyki  
Uniwersytet Śląski w Katowicach

[aleksander.lamza@us.edu.pl](mailto:aleksander.lamza@us.edu.pl)

- Czym jest i do czego służy CSS?
- Reguły = właściwości + wartości
- Dołączanie stylów do dokumentu
- Selektory
- Dziedziczenie i kaskadowość
- Przegląd właściwości CSS

# Czym jest i do czego służy CSS?

Żeby się dowiedzieć, czym jest CSS i dlaczego trzeba go znać, najlepiej na własnej skórze odczuć, czym byłby świat bez CSS-a...



Jestem przedstawicielem  
Ministerstwa Informatyzacji  
Republiki Pawianów.  
Szukam kogoś, kto przygotuje  
nam stronę.

Nie ma sprawy.  
HTML-a nauczyłem się kilkanaście lat temu,  
więc spokojnie mogę się tym zająć.

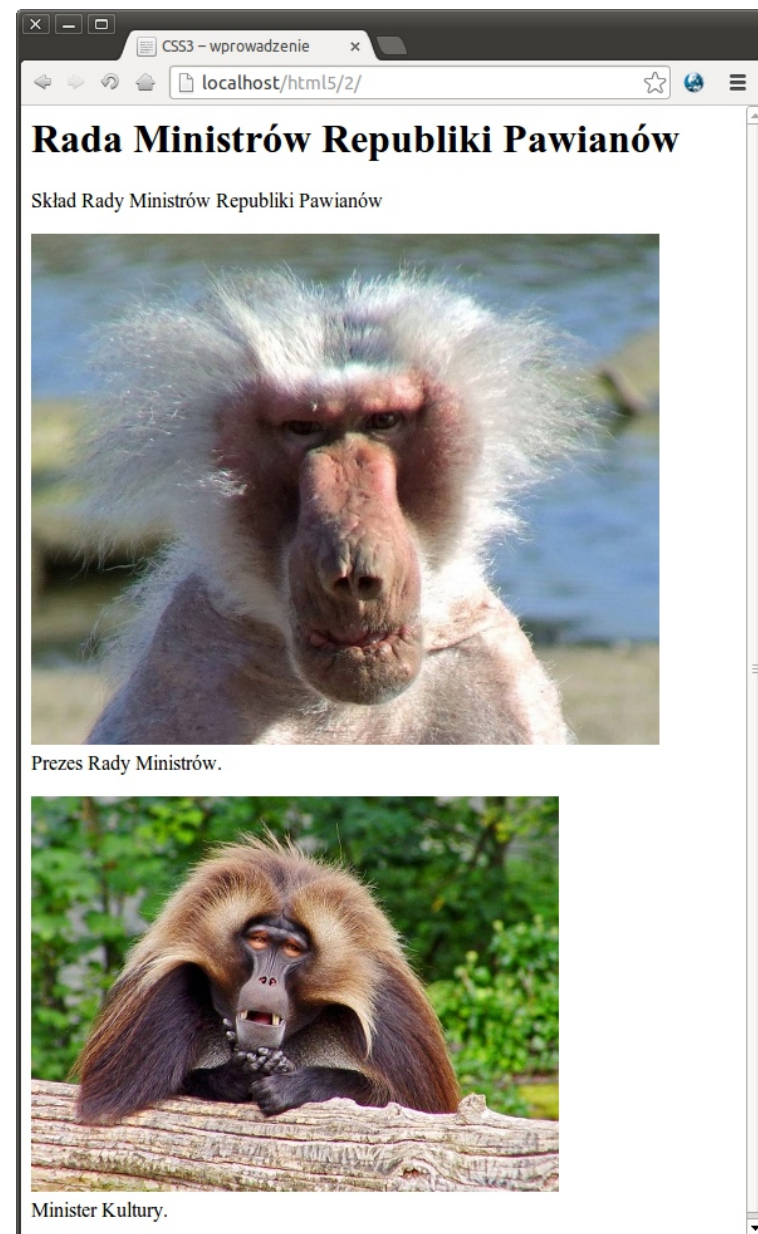


# Czym jest i do czego służy CSS?



Przygotowałem już stronę  
ze składem Rady Ministrów...

```
<body>
  <h1>Rada Ministrów Republiki Pawianów</h1>
  <p>Skład Rady Ministrów Republiki Pawianów:</p>
  <p>
    <br>Prezes Rady Ministrów
  </p>
  <p>
    <br>Prezes Rady Ministrów
  </p>
</body>
```





# Czym jest i do czego służy CSS?



Co to za #\*\$%@?!

←  
Kierownik Pionu Kontroli Jakości  
w Ministerstwie Informatyzacji  
Republiki Pawianów

# Czym jest i do czego służy CSS?



Uff... ale nerwowe te pawiany...  
Wystarczyło powiedzieć, że ma wyglądać inaczej  
- przecież nie ma problemu.

O gustach się nie dyskutuje... a już zwłaszcza z pawianami ;)



# Czym jest i do czego służy CSS?

A jak wygląda kod HTML takiej ślicznej strony?

```
<body bgcolor="green">
  <h1><font color="yellow" size="5" face="Verdana, sans-serif">Rada Ministrów Republiki
Pawianów</font></h1>
  <p><font color="yellow" size="3" face="Times, serif">Skład Rady Ministrów Republiki
Pawianów:</font></p>
  <p align="center">
    <br><font color="white" size="2">Prezes Rady Ministrów</font>
  </p>
  <p align="center">
    <br><font color="white" size="2">Minister Kultury</font>
  </p>
</body>
```

W początkach HTML-a wygląd dokumentu definiowało się w kodzie za pomocą specjalnych elementów i atrybutów.

Ciekawe czy ten kod się spodoba Kierownikowi Pionu Jakości...



# Czym jest i do czego służy CSS?



Co to za #\*\$%@?!



# Czym jest i do czego służy CSS?

## Dlaczego ten kod jest zły?

- Informacje o wyglądzie są **wymieszane** z treścią

```
<h1><font color="yellow" size="5" face="Verdana, sans-serif">Rada  
Ministrów Republiki Pawianów</font></h1>
```

- Jeżeli trzeba będzie wprowadzić zmiany w wyglądzie, trzeba przejrzeć **cały dokument** i wprowadzić te same zmiany w **wielu miejscach**

```
<p align="center">  
  <br><font color="white" size="23">Prezes Rady Ministrów</font>  
</p>  
<p align="center">  
  <br><font color="white" size="23">Minister Kultury</font>  
</p>  
...
```

Jeżeli witryna ma więcej stron (a zwykle tak jest), zmiany trzeba będzie wprowadzić **we wszystkich** dokumentach.

- Uniemożliwia **zróżnicowanie wyglądu** w zależności od medium (ekran monitora, wydruk, mały ekran smartfona itp.)

# Czym jest i do czego służy CSS?

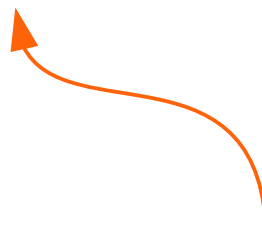
Co zrobić, żeby było dobrze?

Kod **HTML** powinien zawierać tylko to,  
co jest związane z **zawartością (treścią) dokumentu**.

Wszystko to, co jest związane w **wyglądem (prezentacją)**,  
musi trafić gdzieś indziej.

Gdzie?


Do **arkusza stylów**.



CSS (Cascading Style Sheet)

# Czym jest i jak wygląda CSS?

**CSS** jest językiem służącym do opisu prezentacji (wyglądu) dokumentu.



Pierwsza oficjalna rekomendacja W3C dla języka CSS (CSS level 1) została opublikowana w 1996 roku.

Kilka luźnych uwag na temat CSS:

Arkusz stylów składa się z zestawu reguł opisujących wygląd elementów dokumentu.

Arkusz stylów można umieścić w dokumencie HTML lub osobnym pliku.

Reguły stylów można również przypisać do dowolnego elementu HTML (służy do tego atrybut `style`).



# Reguły = właściwości + wartości

Podstawą CSS-a są **reguły** stylów

służy do wybrania elementu  
(elementów) dokumentu HTML

```
selektor {  
    właściwość: wartość;  
    ...  
}
```

określa wybrany aspekt wyglądu

wartość wybranego aspektu wyglądu

HTML

```
<h1>Rada Ministrów Republiki Pawianów</h1>
```

CSS

```
h1 {  
    color: yellow;  
    font-size: 1.8em;  
    font-family: Verdana, sans-serif;  
}
```

# Dołączanie stylów do dokumentu HTML

Style można dołączyć na kilka sposobów

```
<p style="...reguły...">...</p>
```

Reguły można zastosować bezpośrednio do wybranego elementu znajdującego się w dokumencie. W tym przypadku nie używa się selektora.

```
<head>  
  <style>  
    ...reguły...  
  </style>  
</head>
```

Arkusz stylów można umieścić w sekcji head dokumentu. Użycie selektorów jest konieczne.

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

Arkusz stylów można umieścić w osobnym pliku (.css) i dołączyć do dokumentu.

Selektory służą do wybrania pożądaných elementów dokumentu

## Selektory elementów

<b>*</b>	←	wszystkie elementy
<b>E</b>	←	element typu E
<b>E F</b>	←	element F będący potomkiem elementu E
	↕ ?	
<b>E &gt; F</b>	←	element F będący dzieckiem elementu E
<b>E + F</b>	←	element F poprzedzany (bezpośrednio) przez element E
<b>E ~ F</b>	←	element F poprzedzany przez element E

### Przykłady

```
p {...}  
ol li {...}  
ol > li {...}
```



## Selektory atrybutów

**E[a]**

← element z ustawionym atrybutem a

**E[a="w"]**

← element z atrybutem a ustawionym na wartość w

**E[a~="w"]**

← element, w którym atrybut a zawiera wartość w  
(dotyczy listy wartości oddzielonych przecinkami)

**E[lang|="w"]**

← element, w którym wartość atrybutu lang rozpoczyna się od w

**E[a^="w"]**

← element, w którym wartość atrybutu a rozpoczyna się od w

**E[a\$="w"]**

← element, w którym wartość atrybutu a kończy się na w

**E[a\*="w"]**

← element, w którym wartość atrybutu a zawiera w

## Dwa szczególne selektory

Selektor klasy

**E.klasa**

to samo co E[class~="klasa"]

Selektor identyfikatora

**E#id**

to samo co E[class="id"]

## Selektory pseudoelementów

**E:first-line**

pierwszy wiersz tekstu w elemencie E

**E:first-letter**

pierwsza litera tekstu w elemencie E

**E:before**

wstawia zawartość właściwości content przed zawartością elementu E

**E:after**

wstawia zawartość właściwości content po zawartości elementu E



## Wybrane selektory pseudoklas

**E:link**  
**E:hover**  
**E:visited**

**E:enabled**  
**E:disabled**

**E:checked**

**E:nth-child(n)**  
**E:nth-last-child(n)**

**E:nth-of-type(n)**  
**E:nth-last-of-type(n)**

**E:first-child**  
**E:last-child**  
**E:only-child**

**E:not(selektor)**

# Wracamy do pawianów

## HTML

```
<body>
  <h1>Rada Ministrów Republiki Pawianów</h1>
  <p>Skład Rady Ministrów Republiki
Pawianów:</p>
  <p class="image">
    <br>Prezes Rady Ministrów
  </p>
  <p class="image">
    <br>Prezes Rady Ministrów
  </p>
</body>
```

Dokument HTML jest przejrzysty  
(zawiera tylko treść).

Akapitom zawierającym rysunki została przypisana  
klasa „image”, aby je odróżnić od zwykłych akapitów.

(Uwaga: w tym przypadku lepiej byłoby zastosować nowe  
elementy figure i figcaption.)

## CSS

```
body {
  background-color: green;
}

h1 {
  color: yellow;
  font-size: 1.8em;
  font-family: Verdana, sans-serif;
}

p {
  color: yellow;
  font-size: 1em;
  font-family: Verdana, sans-serif;
}

p.image {
  color: white;
  font-size: 0.8em;
  font-family: Times, serif;
  text-align: center;
}

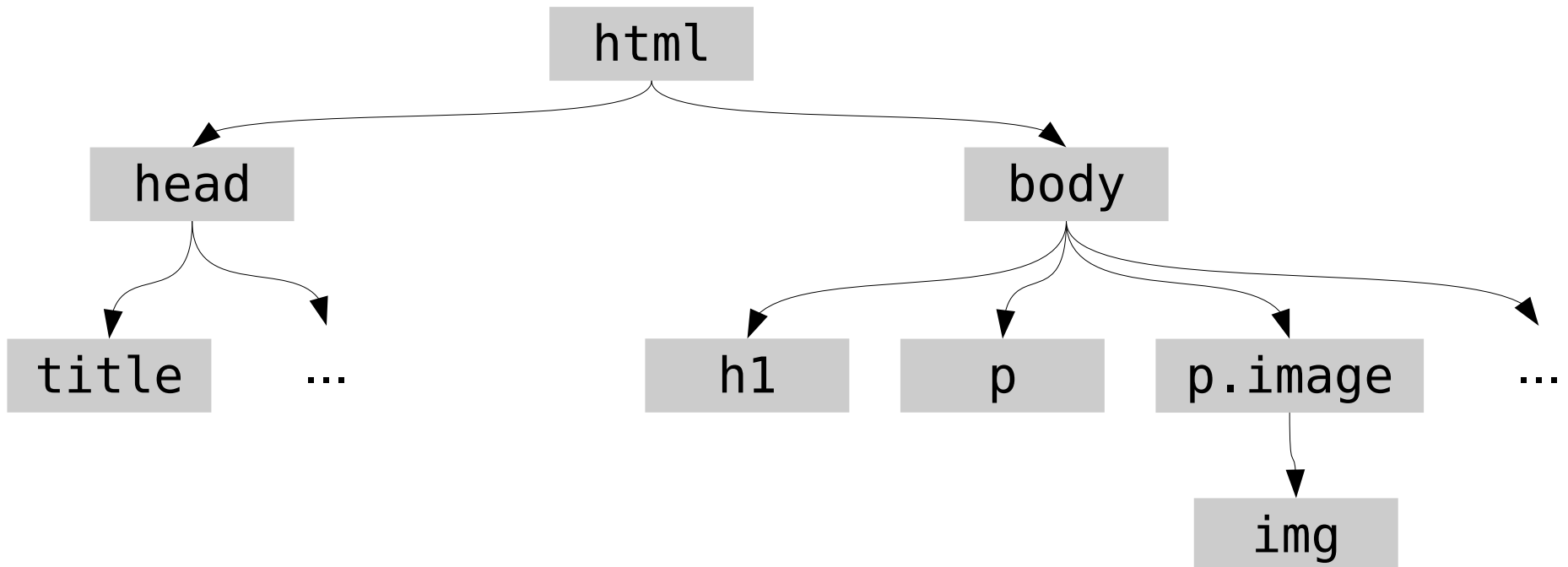
img {
  width: 300px;
  border: 10px solid black;
}
```

Wszystko co jest związane z wyglądem  
trafiło w jedno miejsce.

# Dziedziczenie i kaskadowość

Style zdefiniowane dla danego elementu są  
**dziedziczone**  
przez elementy potomne.

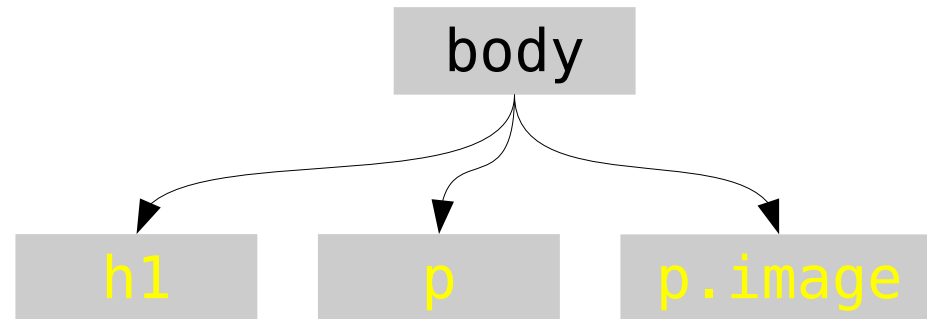
Strukturę naszej strony można przedstawić w taki sposób:





# Dziedziczenie i kaskadowość

Skupmy się na ciele dokumentu oraz ustawieniach **koloru i kroju pisma**

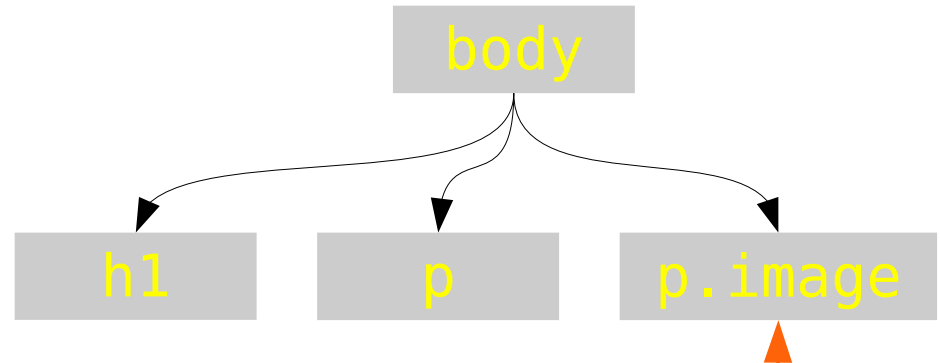


```
body {  
  background-color: green;  
}  
  
h1 {  
  color: yellow;  
  font-size: 1.8em;  
  font-family: Verdana, sans-serif;  
}  
  
p {  
  color: yellow;  
  font-size: 1em;  
  font-family: Verdana, sans-serif;  
}  
  
...
```

Reguły są jednakowe, więc można je przerzucić do nadrzędnego elementu, czyli body.

# Dziedziczenie i kaskadowość

```
body {  
  background-color: green;  
  color: yellow;  
  font-family: Verdana, sans-serif;  
}  
  
h1 {  
  color: yellow;  
  font-size: 1.8em;  
  font-family: Verdana, sans-serif;  
}  
  
p {  
  color: yellow;  
  font-size: 1em;  
  font-family: Verdana, sans-serif;  
}
```



No dobrze, a co z elementem `p.image`?  
Przecież ma być biały, a nie żółty...

# Dziedziczenie i kaskadowość

Dziedziczenie nieodłącznie wiąże się z **przesłanianiem** (override), czyli nadpisywaniem wartości właściwości stylów z wyższych poziomów.

Sprawa się komplikuje, ponieważ do danego elementu dokumentu można zastosować wiele definicji stylów.

HTML

```
...  
<p class="important" style="background-color: red;">UWAGA!</p>  
...
```

Definicje stylów z elementu <style>

```
<head>  
...  
<style>  
  p {  
    color: white;  
  }  
  
  .important {  
    background-color: yellow;  
  }  
</style>  
</head>
```

Zewnętrzny plik CSS

```
p {  
  color: black;  
}
```

Jak będzie wyglądał napis „UWAGA!”

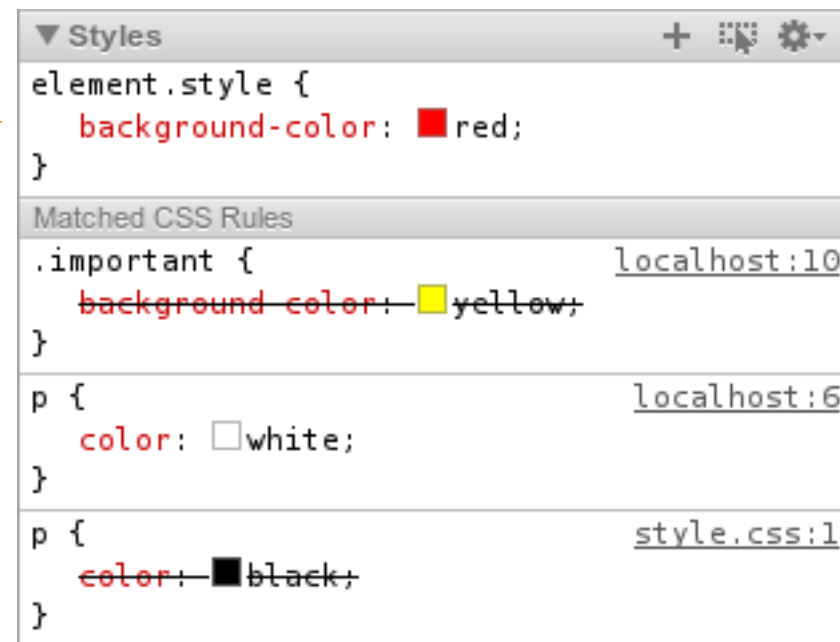


# UWAGA!

Styl „inline” (atrybut style)

Arkusz stylów w sekcji head  
(element <style>)

Zewnętrzny arkusz stylów



„Narzędzia dla programistów” w przeglądarce Chrome



# Dziedziczenie i kaskadowość

W naszym przykładzie, dzięki przesłanianiu, akapit `p.image` będzie taki jak trzeba (biały i mniejszy).

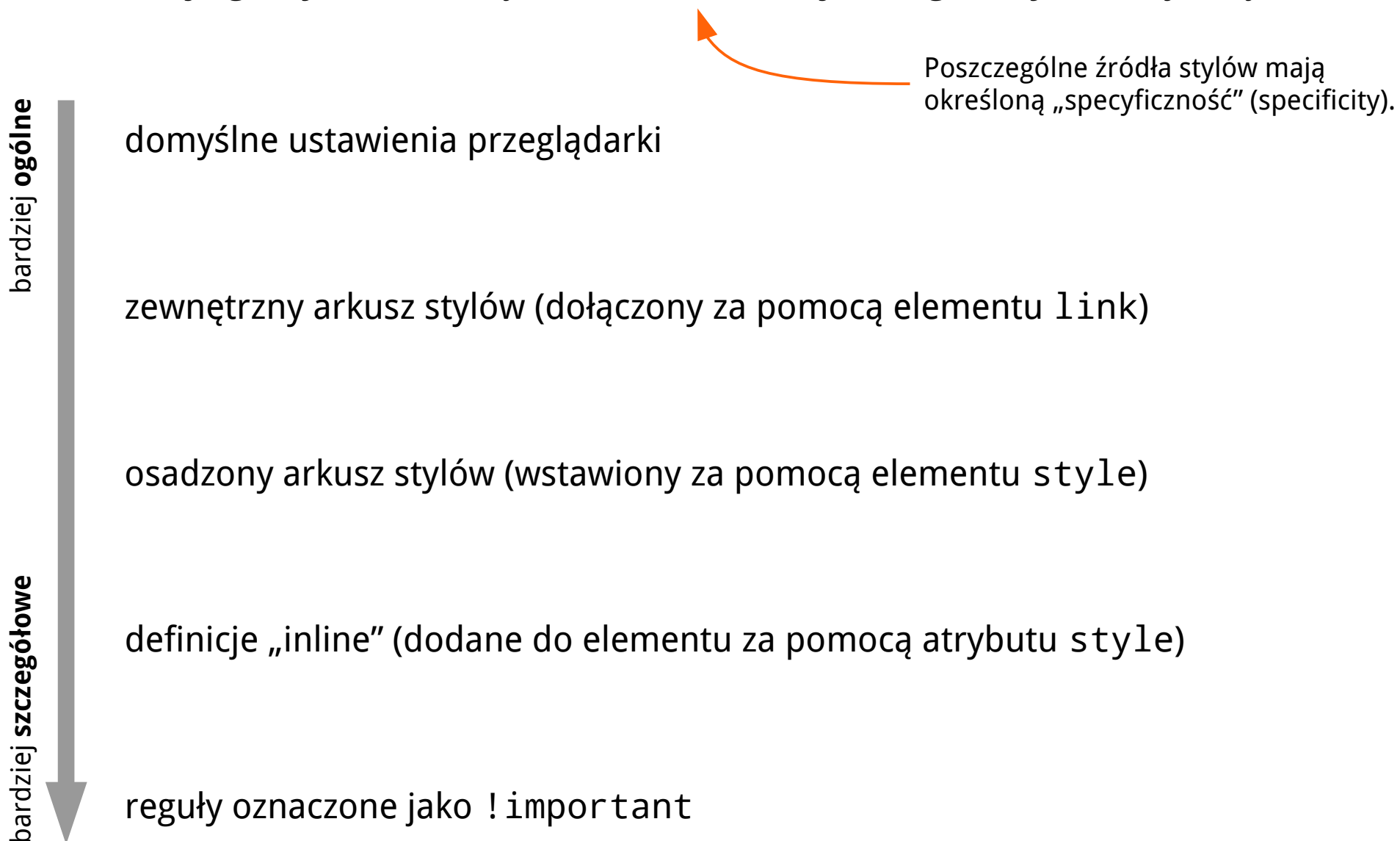
Reguły z `p.image` przesłaniają reguły z `body`.

```
body {  
  background-color: green;  
  color: yellow;  
  font-family: Verdana, sans-serif;  
}  
  
h1 {  
  font-size: 1.8em;  
}  
  
p {  
  font-size: 1em;  
}  
  
p.image {  
  color: white;  
  font-family: Times, serif;  
  font-size: 0.8em;  
  text-align: center;  
}
```

Reguła z `p.image` przesłania regułę z `p`.

# Dziedziczenie i kaskadowość

Kaskadowość polega na „przepływaniu” i przesłanianiu definicji stylów **od bardziej ogólnych** (o niższej wadze) **do bardziej szczegółowych** (o wyższej wadze).



# Dziedziczenie i kaskadowość

Wybrane reguły można oznaczyć jako `!important`, dzięki czemu blokuje się ich przesłanianie.

```
p {  
  color: white !important;  
}  
  
...  
  
<p style="color: red;">Jakiś tekst</p>
```



Tekst zostanie wyświetlony na biało

# Przegląd właściwości CSS

Pełną listę właściwości CSS można znaleźć na stronie

<http://www.w3schools.com/cssref/default.asp>

## CSS Property Groups

- [Animation](#)
- [Background](#)
- [Border and outline](#)
- [Box](#)
- [Color](#)
- [Content Paged Media](#)
- [Dimension](#)
- [Flexible Box](#)
- [Font](#)
- [Generated content](#)
- [Grid](#)
- [Hyperlink](#)
- [Linebox](#)
- [List](#)
- [Margin](#)
- [Marquee](#)
- [Multi-column](#)
- [Padding](#)
- [Paged Media](#)
- [Positioning](#)
- [Print](#)
- [Ruby](#)
- [Speech](#)
- [Table](#)
- [Text](#)
- [2D/3D Transform](#)
- [Transition](#)
- [User-interface](#)

Inne ciekawe miejsca:



<http://www.css3files.com/>



<http://www.css3.info/>

**CSS3**, please!

This element will receive instant changes as you edit the CSS rules on the left. Enjoy!

<http://css3please.com/>