



# WTF: CSS II



W poprzednim odcinku...

`<znacznik style="color:yellow;">treść</znacznik>`

Styl przy znaczniku z wykorzystaniem atrybutu  
*style* nazywamy stylem *inline*

WTF



Czy ma to sens?



#id  
.class  
element

# Prawdziwe selektory

<znacznik class="name">treść</znacznik>

Jak dostać się do takiego elementu?

```
<znacznik class="name">treść</znacznik>
```

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

. (kropka) – selektor klasy

```
<znacznik id="name">treść</znacznik>
```

```
#name {  
  właściwość: wartość;  
}
```

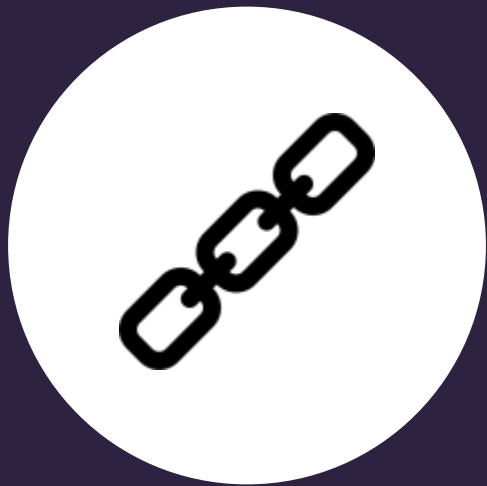
# (hash) – selektor id



**WTF**



!important



Łączenie selektorów

```
#name {  
  color: red;  
}
```

```
<span style="color: yellow" id="name">treść</span>
```

Jakiego koloru będzie treść?



Specyficzność w CSS

				razem
1  Polska	2	2	0	4
2  Wielka Brytania	2	1	0	3
3  Ukraina	1	1	0	2
4  Francja	1	0	0	1
4  Portugalia	1	0	0	1
6  Niemcy	0	1	1	2
7  Belgia	0	1	0	1
7  Słowacja	0	1	0	1
9  Białoruś	0	0	1	1
9  Hiszpania	0	0	1	1
9  Holandia	0	0	1	1
9  Turcja	0	0	1	1
9  Włochy	0	0	1	1
9  Węgry	0	0	1	1

WTF

( \_ , \_ ) , \_ , \_ , \_

!important

style=""

#name

.name

element

```
body h1 {  
  color: yellow;  
}
```

**WTF**

0,0,0,0,2



!important

style=""

#name

.name

element

```
h1 {  
  color: red;  
}
```

0,0,0,0,1

!important

style=""

#name

.name

element

```
html body h1 {  
  color: yellow;  
}
```

**WTF**

0,0,0,0,3

!important

style=""

#name

.name

element

```
#main-header {  
  color: red;  
}
```

0,0,1,0,0

!important

style=""

#name

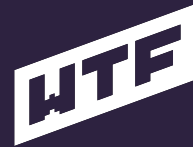
.name

element





```
html body h1#main-header {  
  color: yellow;  
}
```



0,0,1,0,3

!important      style=""      #name      .name      element

```
body .section #main-header {  
  color: red;  
}
```

0,0,1,1,1

!important      style=""      #name      .name      element



```
html body h1 {
  color: yellow;
}
```

**WTF**

0,0,0,0,3

!important

style=""

#name

.name

element

```
body #main-header {
  color: red !important;
}
```

1,0,1,0,1

!important

style=""

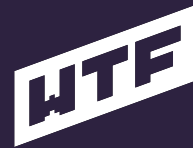
#name

.name

element



<h1 style="color:blue">Nagłówek</h1>



0, 1, 0, 0, 0

!important

style=""

#name

.name

element

```
h1{  
  color: red !important;  
}
```

1, 0, 0, 0, 1

!important

style=""

#name

.name

element



`<h1 style="color:blue">Nagłówek</h1>`

WTF

0, 1, 0, 0, 0



!important

style=""

#name

.name

element

```
body section#main h1.heading {  
  color: red;  
}
```

0, 0, 1, 1, 3

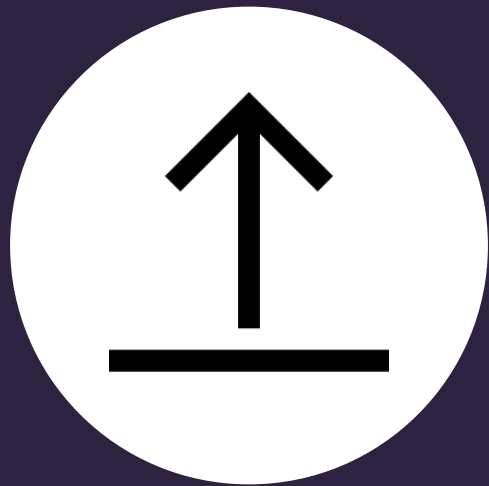
!important

style=""

#name

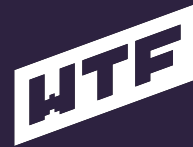
.name

element



Jeśli specyficzność jest równa –  
pierwszeństwo ma selektor "bliżej"  
elementu

# style.css

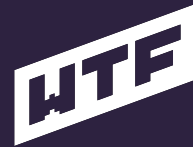


```
body section#main h1.heading {  
  color: red;  
}
```

```
body section#main h1.heading {  
  color: blue;  
}
```



# style.css



```
body section#main h1.heading {  
  color: red;  
  color: blue;  
}
```





Rekomenduję dodawać klasy do  
wszystkich elementów i unikać  
zagnieżdżeń

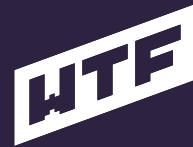




There are only two hard things in Computer Science: cache invalidation and naming things.

-- *Phil Karlton*

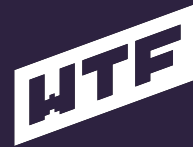
# Nazewnictwo klas



- używaj pojedynczych słów z j. angielskiego
- nazywaj rzeczy zgodnie z ich funkcją na stronie
- startuj z małej litery np. header nie Header
- nie zaczynaj od cyfry
- jeśli musisz użyć dwóch słów użyj tzw. kebab case np. news-listing
- staraj się, żeby element miał jedną klasę (nie zwiększaj specyficzności bez potrzeby)



# Podsumowanie



- html class="name" = css .name
- html id="name" = css #name
- stylujemy po klasach (bez zagnieżdżeń!)
- nazywamy klasy po angielsku zgodnie z przeznaczeniem
- tworzymy klasy nawet mamy jeden element danego typu
- specyficzność może dać popalić ;)
- nie używamy *!important* i stylów *inline*'owych





**To be continued...**

