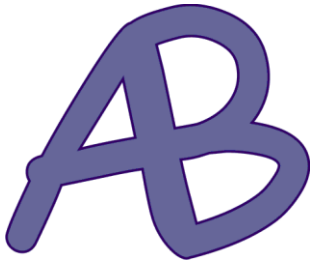
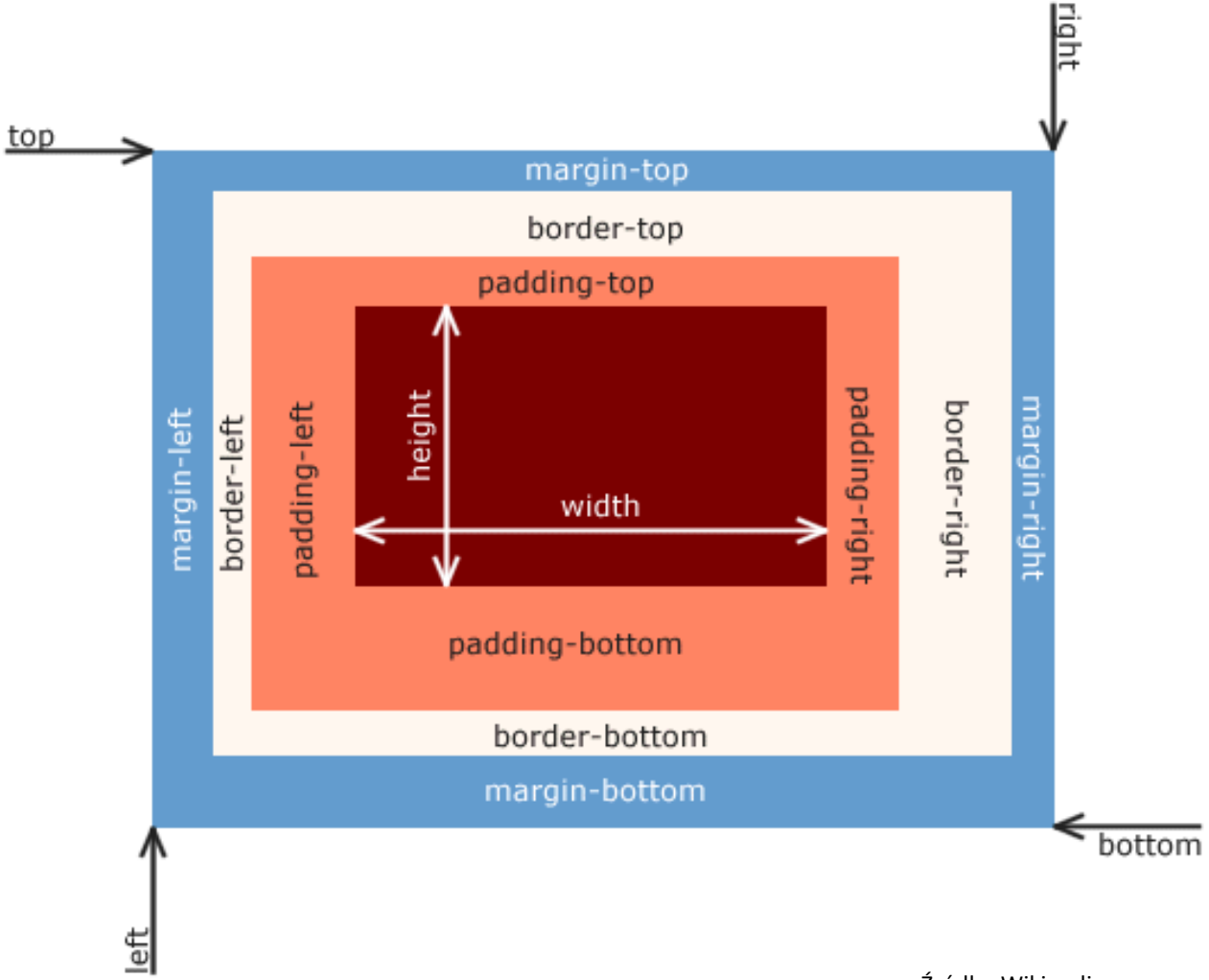
Wykład 1



CSS

Model pudełkowy (CSS box model)



Źródło: Wikipedia

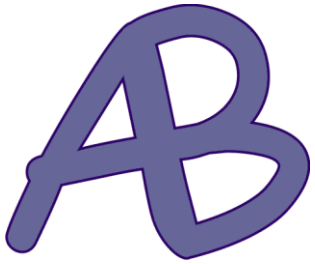
Do ustalania rozmiarów pudełka służą atrybuty **`width`** (szerokość) i **`height`** (wysokość). Określają one rozmiar zarezerwowany dla zawartości.

Obszar ten może mieć tło (**`background-color`**) i obramowanie (**`border`**).

Aby określić całkowity rozmiar zajmowany przez pudełko należy więc uwzględnić:

- ✓ marginesy wewnętrzne (**`padding`**),
- ✓ obramowanie (**`border`**),
- ✓ marginesy (**`margin`**).

(Pamiętaj o deklaracji typu dokumentu — jest ona potrzebna, aby przeglądarka poprawnie zinterpretowała model pudełkowy).



Klasy i selektory

Selektor potomka

- Jeśli selektorem w regule CSS jest nazwa znacznika, to będzie jej podlegał każdy znacznik tego typu.
- Na przykład, poniższa reguła sprawi, że kolor tekstu we wszystkich akapitach będzie czerwony:

```
p {color:red;}
```

- Co jednak zrobić, jeśli chcemy, aby tylko jeden, określony akapit miał czerwony tekst?
- Do bardziej precyzyjnej selekcji znaczników służy selektor potomka. Oto przykład takiego selektora:

```
div p {color: red;}
```

- Teraz czerwony kolor tekstu będą miały akapity znajdujące się wewnątrz znacznika `div`.

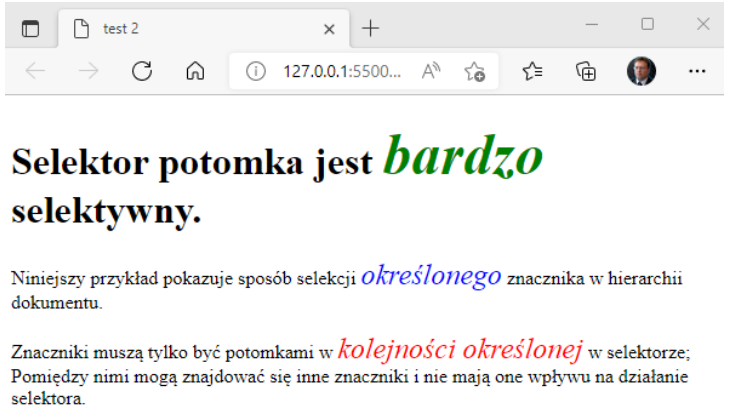
Selektor potomka (przykład)

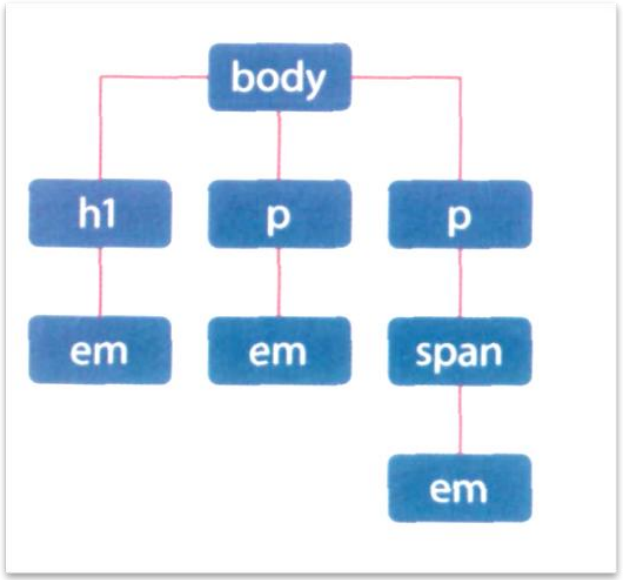
```
<h1>Selektor potomka jest <em>bardzo</em> selektywny.</h1>

<p>Niniejszy przykład pokazuje sposób selekcji  <em>określonego</em> znacznika w hierarchii  dokumentu.</P>

<p>Znaczniki muszą tylko być potomkami  <span>w <em> kolejności określonej</em> w selektorze</span>; Pomiędzy nimi mogą znajdować się inne znaczniki  i  nie mają one wpływu na działanie selektora.</p>
```

```
em {font-size: 1.5em;}
p em {color: blue;}
h1 em {color: green;}
p span em {color:  red;}
```





Selektor dziecka

- Istnieje także możliwość napisania selektora odnoszącego się tylko do elementów będących dziećmi (bezpośrednimi potomkami) innego elementu. Służy do tego znak **>**

```
p > em {color: green;}
```

Uwaga:
Przeglądarka Internet Explorer 6 całkowicie je ignoruje (natomiast w Internet Explorerze 7 i nowsze są już obsługiwane)

Wspólne selektory dla wielu znaczników

- Możliwe jest zdefiniowanie stylu dla grupy znaczników

Przykładowo, poniższy styl może zostać użyty do określenia domyślnej czcionki witryny:

```
body, p, td, th, div, dl, ul, ol
{
  font-family: Tahoma, Verdana, Arial, Helvetica,
  sans-serif:
  font-size: 1em:
  color: #000000:
}
```

SELEKTOR UNIWERSALNY

Selektor uniwersalny * (potocznie nazywany gwiazdką) oznacza wszystko. Dlatego jeśli w arkuszu stylów użyjemy następującej reguły:

```
* {color:green;}
```

tekst na całej stronie będzie zielony, chyba że zostaną utworzone dodatkowe reguły określające inaczej

Interesującym zastosowaniem tego selektora jest użycie go jako odwrotności selektora dziecka:

```
p * em {font-weight:bold;}
```

Powyższy selektor wybiera wszystkie znaczniki em, które są nie-bezpośrednimi potomkami znacznika p.

Selektor braci

Selektor ten pozwala nadać formatowanie znacznikowi znajdującemu się za określonym znacznikiem mającym tego samego rodzica (czyli znajdującym się na tym samym poziomie w hierarchii dokumentu). Oto przykład jego użycia:

```
h1 + p (font-variant:small-caps;}
```

Jest to dobry sposób na pogrubienie pierwszego elementu listy. Na przykład:

```
ul + li (font-weight: bold;}
```

Selektor braci działa w przeglądarkach zgodnych ze standardami, przykładowo działa w IE 7 i nowszych, nie działa natomiast w IE 6.

Identyfikatory (wprowadzenie)

- Identyfikatory pisze się podobnie jak klasy, z tą różnicą, że zamiast kropką oznacza się je znakiem # w arkuszu stylów.

Jeśli akapit ma identyfikator jak poniżej:

```
<p id="specialtext">To jest specjalny tekst.</p>
```

odpowiadający mu selektor potomka wygląda następująco:

```
p#specialtext  {reguły stylistyczne}
```

- Poza tym identyfikatory działają tak samo jak klasy — wszystka co zostało powiedziane do tej pory na temat klas, ma także zastosowanie do identyfikatorów. Co je zatem różni?

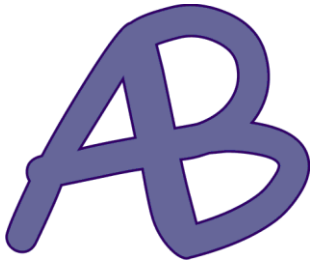
Identyfikatory (wprowadzenie)

- Zgodnie z zasadami XHTML dany identyfikator (np. id="mainmenu") może wystąpić w dokumencie tylko raz, natomiast klasa — dowolną liczbę razy.
- Aby zatem oznaczyć unikatową część strony, na przykład menu nawigacyjne, do którego ma być zastosowany specjalny zestaw reguł CSS, można użyć znacznika div z identyfikatorem.
- Do identyfikacji kilku specjalnych akapitów na stronie, które wymagają tego samego stylu różniącego się od stylu głównego akapitów, należy użyć klasy
- Dodatkowo identyfikatory służą do identyfikacji znaczników w języku JavaScript

Identyfikatory (wprowadzenie)

- Selektory ID mogą być stosowane w połączeniu z innymi rodzajami selektorów. Na przykład w poniższym stylu selektor ID odnosi się do nieodwiedzonych łączy występujących w akapicie `pasek1`:

```
#pasek1 a:link {
font-weight: bold:
color: #FFFFFF: }
```



KASKADOWOŚĆ I DZIEDZICZENIE

Dziedziczenie

Znacznik **body** jest przodkiem wszystkich pozostałych znaczników — wszystkie znaczniki formatowane za pomocą CSS są jego potomkami.

Dlatego dzięki dziedziczeniu w CSS, jeśli w arkuszu stylów napiszemy poniższą regułę:

```
body {font-family: verdana, helvetica, sans-serif; color:blue;)
```

tekst we wszystkich elementach dokumentu oddziedziczy te style i będzie napisany niebieską czcionką kroju Verdana (lub inną z podanych, jeśli Verdana będzie niedostępna). Poziom hierarchii nie ma tu żadnego znaczenia.

Dziedziczenie

Wiele własności CSS podlega dziedziczeniu w ten sposób w szczególności style dotyczące tekstu. Istnieje jednak duża liczba własności CSS których dziedziczenie nie miałoby sensu.

Są to przede wszystkim własności związane z pozycjonowaniem i wyświetlaniem elementów prostokątnych, jak obramowania, marginesy i dopełnienie.

Dodatkowo należy zachować ostrożność podczas używania względnych jednostek wielkości, jak procenty czy em. Jeśli rozmiar tekstu w znaczniku zostanie ustawiony na 80% i znacznik ten jest potomkiem innego znacznika, którego rozmiar tekstu również jest ustawiony na 80%, to w tym pierwszym rzeczywisty rozmiar tekstu wyniesie 64% (80% z 80%).

Kaskadowość

- ✓ Jak sama nazwa wskazuje, kaskadowość oznacza kaskadę arkuszy stylów związaną z hierarchią elementów w dokumencie.
- ✓ Jej celem jest pozwolenie przeglądarce na podjęcie decyzji, którego z wielu źródeł określonej własności dla określonego znacznika należy użyć.

> Kaskadowość to potężny mechanizm. Dzięki jego zrozumieniu można pisać bardzo oszczędny i łatwy do edycji kod CSS. Pozwala na tworzenie dokumentów, które wyglądają tak, jak zostało to zaplanowane, jednocześnie pozostawiając niektóre aspekty kontroli wyglądu, jak rozmiary czcionek, w rękach użytkowników, którzy mogą mieć nietypowe potrzeby.

Źródła stylów

Kolejność lokalizacji przeszukiwanych przez przeglądarkę w celu znalezienia arkuszy stylów (kaskada) przedstawia się następująco:

- Domyślny arkusz stylów przeglądarki.
- Zewnętrzny arkusz stylów.
- Osadzony arkusz stylów.
- Lokalny styl znacznika.

Źródła stylów

Arkusz stylów przeglądarki (domyślny) - ukryty głęboko w jej wnętrzu, ponieważ każdy znacznik ma jakiś określony styl.

Na przykład znacznik <hi> jest powiększony i pogrubiony, zawartość znacznika jest pisana kursywą, a listy są wcięte i każdy ich element jest poprzedzony punktorem. Nie trzeba pisać żadnych arkuszy stylów, aby to formatowanie było zastosowane.

Domyślny arkusz stylów przeglądarki Firefox ma nazwę html.css. Można go modyfikować według własnych upodobań.

Zasady kaskadowości

Zasada 1: Odszukanie wszystkich deklaracji mających zastosował nie do każdego elementu i każdej własności. Kiedy przeglądarka ładuje stronę, sprawdza każdy znacznik pod kątem tego czy nie ma pasującej do niego reguły.

Zasady kaskadowości

Zasada 2: Sortowanie według kolejności i wagi. Przeglądarka sprawdza po kolei wszystkie źródła stylów i ustawia w trakcie tego procesu wszystkie znalezione własności.

Jeśli jakaś ustawiona własność jest zdefiniowana także na niższym poziomie, zostaje ustawiona ponownie według nowej wartości.

Działanie to jest powtarzane w razie potrzeby tak długa aż przeglądarka przetworzy ostatni dostępny arkusz stylów. Styl zastosowany do formatowania każdego elementu to ostatni z napotkanych przez przeglądarkę.

Zasady kaskadowości

Zasada 2: Sortowanie według kolejności i wagi.

Przykład: Tabela przedstawia ten proces dla strony zawierającej kilka znaczników p. Zakładamy, że dwa z tych znaczników posiadają style lokalne, które ustawiają kolor tekstu jako czerwony. Tekst wszystkich pozostałych akapitów będzie miał w tym przypadku kolor niebieski.

TABELA 2.1. Przykłady kaskadowości

Lokalizacja	Znacznik	Własność	Wartość
Domyślny arkusz stylów	p	color	black
Arkusz stylów użytkownika			
Autorski zewnętrzny arkusz stylów	p	color	blue
Autorski osadzony arkusz stylów			
Autorski lokalny arkusz stylów	p	color	red

Zasady kaskadowości

Zasada 2: Sortowanie według kolejności i wagi.

UWAGA: Oczywiście nie jest to takie proste. Istnieje jeszcze waga deklaracji.
Reguła może być zdefiniowana jako ważna:

Słowo !important powinno być oddzielone od deklaracji, której dotyczy, tylko jedną spacją. Dopiero po nim należy postawić średnik.

Niniejsza reguła informuje, że ważne jest, aby kolor tekstu był czerwony. Dzięki temu tekst będzie czerwony, nawet jeśli gdzieś na niższym poziomie kaskady znajduje się deklaracja mówiąca inaczej.

Przed zastosowaniem deklaracji !important należy się zawsze głęboko zastanowić

Zasady kaskadowości

Zasada 3: Sortowanie według precyzji. Precyzja (ang. specificity) określa, jak bardzo precyzyjny jest selektor.

Jak wiemy jeśli arkusz stylów zawiera dwie reguły:

```
p  {font-size:12px;}
```

i

```
p.largetext (font-size:16px;}
```

to tekst w poniższym fragmencie strony będzie miał wysokość 16 pikseli, ponieważ selektor drugiej z przedstawionych reguł jest bardziej precyzyjny niż pierwszego.

```
<p class="largetext">Fragment tekstu.</p>
```

Zasady kaskadowości

Zasada 3: Sortowanie według precyzji.

Może się to wydawać oczywiste, ale co się stanie z zaprezentowanym fragmentem kodu XHTML, jeśli zostaną użyte następujące reguły?

```
p {font-size:12px;}

.largetext {font-size:16px;}
```

Obie pasują do naszego znacznika, ale klasa przesłania selektor znacznika i tekst będzie miał 16 pikseli wysokości.

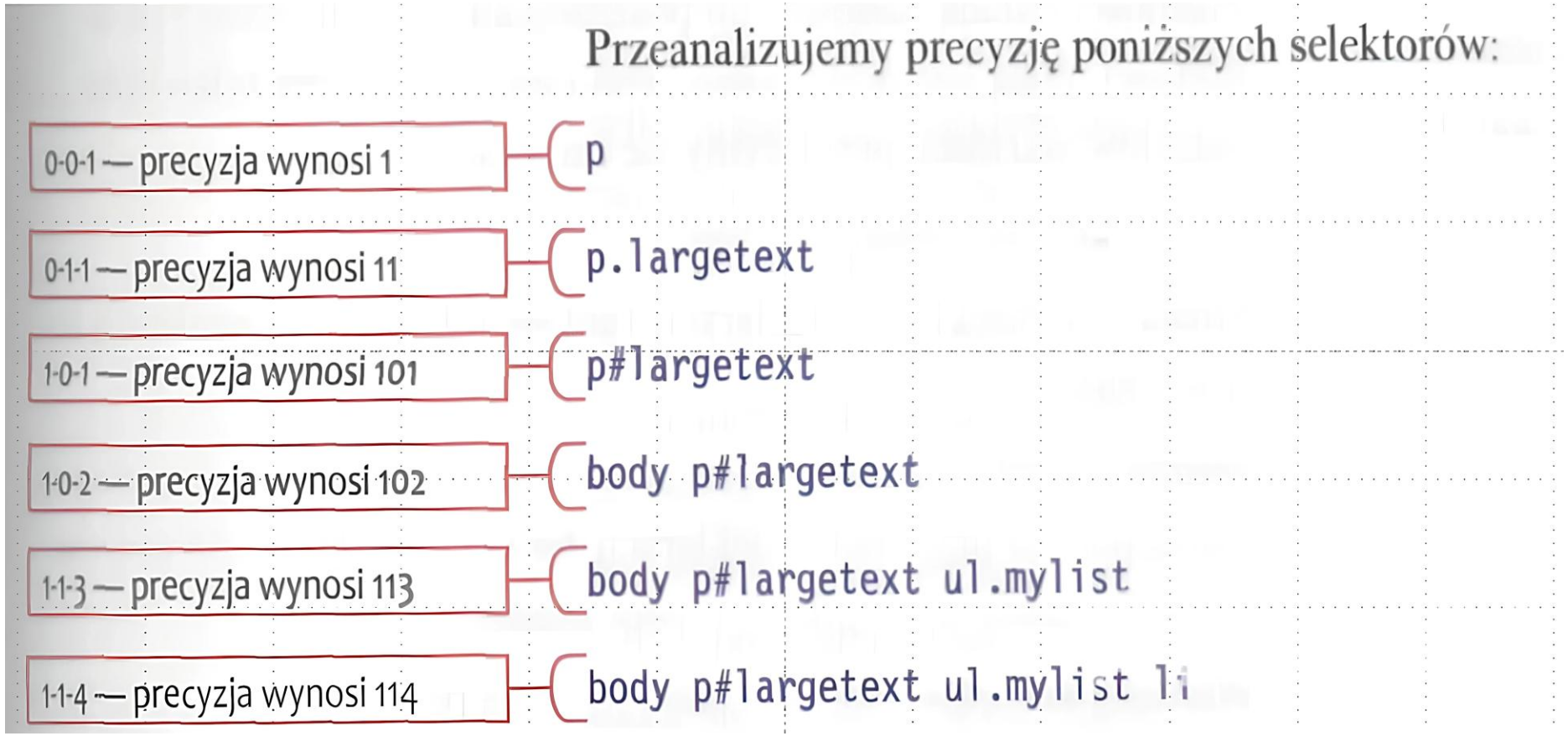
Oto dlaczego: precyzja selektora znacznika wyrażona liczbowo to w tym przypadku 1, natomiast klasy to 1-0.

Zasady kaskadowości

Zasady obliczania precyzji selektorów. Jest to trzycyfrowy system punktowania oparty na formacie trzech pól: a-b-c. Precyzję obliczamy następująco:

1. Policz, ile jest w selektorze identyfikatorów, i ich liczbę podstaw w miejsce a.
2. Policz, ile jest w selektorze klas, pseudoklas i innych atrybutów, a ich łączną liczbę podstaw w miejsce b
3. Policz, ile jest w selektorze nazw elementów, i ich liczbę podstaw w miejsce c.
4. Powstaje w ten sposób trzycyfrowa liczba (oczywiście jest to tylko umownie liczba, ponieważ ułatwia to odczyt wyników; nie należy zapominać, że można uzyskać wynik typu 0-1-12, od którego bardziej precyzyjny jest wynik 0-2-0).

Zasady kaskadowości



Zasady kaskadowości

Zasada 4: Sortowanie według kolejności. Jeśli dwa selektory mają dokładnie taką samą wagę, górę bierze ten, który znajduje się na niższym poziomie kaskady.

Kaskadowość w uproszczeniu

W uproszczeniu konieczne jest zapamiętanie tylko trzech zasad. Dotyczą one wszystkich możliwych sytuacji.

Zasada 1: Selektory identyfikatorów przesłaniają selektory klas, które z kolei przesłaniają selektory znaczników.

Zasada 2: Jeśli jedna własność dla danego znacznika jest zdefiniowana w więcej niż jednym miejscu w kaskadzie, style lokalne mają pierwszeństwo przed osadzonymi, które przesłaniają style zewnętrzne. Jednak zasada 1 ma większą wagę niż zasada 2 — najbardziej precyzyjny selektor ma zawsze pierwszeństwo przed innymi, mniej precyzyjnymi.

Zasada 3: Style zdefiniowane bezpośrednio przesłaniają style odziedziczone, bez względu na precyzję. Jeśli zasada ta nie jest jasna, spójrzmy na poniższy kod XHTML:

```
<div id="cascadedemo">
<p id="inheritancefact">Dziedziczenie w kaskadzie jest <em>słabe</em>.</p>
</div>
```

Poniższa, bardzo precyzyjna reguła spowoduje, że cały tekst, łącznie ze słowem `słabe`, będzie niebieski, ponieważ element `em` odziedziczy kolor po swoim rodzicu, znaczniku `p`.

2-0-4 — `html body div#cascadedemo p#inheritancefact {color:blue:}`

Jeśli dodamy poniższą regułę, która jest bardzo mało precyzyjna, tekst w znaczniku `em` będzie czerwony.

0-0-1 — `em {color:red}`

Odziedziczony styl zostanie przesłonięty przez bezpośredni styl znacznika `em`, bez względu na swoją wysoką precyzję.

Oto trzy proste zasady kaskadowości. Chyba nie ma problemu z ich zrozumieniem?

Literatura:

W prezentacji wykorzystano fragmenty i zadania z książek:

- Andrew R.; *CSS. Antologia. 101 wskazówek i tricków*. Helion, Gliwice 2005.
- Wyke-Smitch Ch.; *CSS Witryny szyte na miarę*. Helion, Gliwice 2008.
- Sokół M.; *CSS Ćwiczenia*. Helion, Gliwice 2007.