

Formatarea paginilor Web prin intermediul foilor de stil CSS

Mihai Gabroveanu

Cascading Style Sheets

- Un Cascading Styl e Sheet (CSS) descrie modul de reprezentare al unei pagini HTML, de regulă într-un fișier separat
- CSS are următoarele avantaje:
 - Separă conținutul de prezentare
 - Permite definirea modului de afișare și layout-ul tuturor paginilor unui site web într-un singur loc
 - Poate fi utilizat atât pentru pagini HTML cât și pentru documente XML
- CSS are următoarele dezavantaje:
 - Nu toate browserele suportă CSS-uri

Neajunsuri ale limbajului HTML

- HTML a fost creat inițial pentru a descrie *conținutul* unui document
- Creatori de pagini nu descriu modul de reprezentare – browser-ul se ocupă de acest lucru
- Acesta este o abordare bună, dar uneori nu e satisfăcătoare pentru anumiți designeri
 - Adesea se dorește un control mai mare al modului în care sunt afișate paginile Web
- Apare necesitatea introducerii de noi tag-uri pentru a controla *modul de reprezentare*

Sintaxa fișierelor CSS, I

- Sintaxa fișierelor CSS este foarte simplă – Aceasta constă dintr-o listă de:
 - selectori (pentru identificarea tag-urilor) și
 - descriptori (pentru a indica ce vrem să facem cu acestea)
 - Exemplu: `h1 {color: green; font-family: Verdana}` spune că tot ce este inclus în tag-ul `h1` va fi afișat în verde utilizând fontul Verdana
- Un fișier CSS va conține deci o listă cu astfel de perechi (selector,descriptor)
 - Selectorii pot fi simple tag-uri HTML sau XML, dar se pot defini și moduri de combinare a tag-urilor
 - Descriptorii sunt definiți în standard-ul CSS

Sintaxa CSS

- Sintaxa generală este:

```
selector {proprietate: valoare}
```

– sau

```
selector1, ..., selectorN {  
  proprietate1: valoare1;  
  . . .  
  proprietatek: valoarek;  
}
```

– unde

- selector* este tagul ce va fi afectat (acesta este case-sensitive dacă și numai dacă limbajul documentului este case-sensitive), ex: HTML este case-insensitive, XHTML și XML sunt case sensitive
- proprietate* și *valoare* descriu modul de afișare al aceluiași tag
- Spațiile și punct și virgula sunt opționale
- Punct și virgula este utilizată pentru a separa perechile proprietate:valoare

Selectori, I

- Așa cum am văzut, un tag (X)HTML sau XML poate fi un selector simplu :
body { background-color: #ffffff }
- Se pot utiliza și selectori multipli:
em, i {color: red}
Putem repeta selectori:
h1, h2, h3 {font-family: Verdana; color: red}
h1, h3 {font-weight: bold; color: pink}
 - Când se suprapun proprietățile se utilizează ultima valoare definită
- Selectorul universal * se aplică asupra tuturor elementelor:
* {color: blue}
 - Când valorile se suprapun, selectorul se specifică primează înaintea celui general

Exemplu de fișier CSS

- /* Acesta este un comentariu */*
- h1,h2,h3 {font-family: Arial, sans-serif;} */* utilizeaza primul font disponibil */*
- p, table, li, address {
 font-family: "Courier New"; */* se cuprind intre " " valorile cu spatii */*
 margin-left: 15pt; */* specifica indentarea */*
}
- p, li, th, td {font-size: 80%;} */* dimensiunea fontului este de 80% */*
- th {background-color:#FAEBD7} */* culorile se pot specifica in hexa */*
- body { background-color: #ffffff;}
- h1,h2,h3,hr {color:red;} */* se pot specifica culori si prin nume */*
- a:link {color:darkred} */* un link nevizitat */*
- a:visited {color:darkred} */* un link vizitat */*
- a:active {color:red} */* un link activ */*
- a:hover {color:red} */* cand mouse-ul este peste el */*

Exemplu

```
<html>  
<head>  
  <title>HTML/CSS Test Suprascriere stiluri</title>  
  <style TYPE="text/css">  
    body { background-color: #ffffff }  
    em, i {color: red}  
    h1, h2, h3 {font-family: Verdana; color: red}  
    h1, h3 {font-weight: bold; color: pink}  
    * {color: blue}  
  </style>  
</head>  
<body>  
  <p>Am utilizat urmatoarele stiluri CSS, in ordine:</p>  
  <em>em {color: red}</em><br>  
  <b>b {color: blue}</b><br>  
  <h3>h3 {font-family: Verdana;font-weight: bold; color: pink}</h3>  
  <h2>h2 {font-family: Verdana; color: red}</h2>  
  <h4>h4 {color: blue}</h4>  
</body>  
</html>
```



Selectorii, II

- Selectorul descendent selectează un tag cu un anumit ascendent (strămoș):
 - `p code { color: brown }`
 - selectează tag-ul `code` dacă acesta se află undeva în interiorul unui paragraf (indiferent pe ce nivel de adâncime)
- Selectorul copil `>` selectează un tag dacă are un anumit părinte:
`h3 > em { font-weight: bold }`
selectează tag-ul `em` numai dacă părintele lui imediat este tag-ul `h3`
- Selectorul adiacent selectează un element dacă se află imediat după un altul:
`b + i { font-size: 8pt }`
Exemplu: `<b>Text Bold</b> <i>Text italic</i>`
Se va afișa astfel: **Text Bold** *Text italic*

Selectorii, III

- Un selector simplu de atribut permite selectarea elementelor care au anumite atribute, indiferent de valoarea acestora:
 - Sintaxa: `element[atribut] { ... }`
 - Exemplu: `table[border] { ... }`
- Un selector de atribut cu valoare permite selectarea elementelor ce au un atribut cu o valoare dată:
 - Sintaxa: `element[atribut="valoare"] { ... }`
 - Exemplu: `table[border="0"] { ... }`

Exemplu

```
<html>
<head>
<title>HTML/CSS Test Selectorii</title>
<style TYPE="text/css">
p code { color: green }
h3 > em { color: red; font-weight: bold }
b + i { font-size: 8pt }
</style>
</head>
<body>
Secvența de cod:
<code>if (x ==0) exit(1); </code><br>
<p><code>if (x ==0) exit(1); </code></p>
<p><i><code>if (x ==0) exit(1); </code></i></p>
<h3><em>Copil</em></h3>
<h3><u><em>Neptot</em></u></h3>
<p><b>Text Bold</b> <i>Text italic</i> <i>Text italic</i></p>
<p><i>Text italic</i> <b>Text Bold</b></p>
</body>
</html>
```



Atributul class

- Atributul `class` permite definirea de stiluri multiple (clase de stiluri) pentru același element (tag)
 - În CSS:
`p.important {font-size: 24pt; color: red}`
`p.fineprint {font-size: 8pt}`
 - În HTML:
`<p class="important">Text important!</p>`
`<p class="fineprint">Text pt. listare</p>`
- Pentru definirea unui selector care să se aplice tuturor elementelor din acea clasă, se elimină numele elementului (dar păstrăm punctul):
`.fineprint {font-size: 8pt}`

Atributul id

- Atributul id se definește similar cu atributul class, dar utilizează simbolul # în loc de .
 - În CSS:
`p#important {font-style: italic}` sau
`#important {font-style: italic}`
 - În HTML:
`<p id="important">text important</p>`
- class și id pot fi utilizate simultan, fără ca numele să difere:
`<p class="important" id="important">`

Diferența dintre class și id

- Atributele class și id deși ambele definesc stiluri multiple pentru un tag în practică se folosesc astfel:
 - Atributul class este utilizat atunci când într-o pagină există mai multe tag-uri ce au asociat acel nume de clasă
 - Atributul id este utilizat atunci când într-o pagină există un singur tag ce are asociat acel nume de identificator. De asemenea acel identificator asociat se utilizează adesea pentru a identifica din JavaScript acel element.
- Atunci când un tag are definite stiluri prin class și id simultan, stilurile definite prin id sunt prioritare.

Tag-urile div și span

- Taguri HTML div și span sunt definesc containere ce susțin un set de taguri HTML asupra cărora putem aplicăm stiluri de formatare CSS;
- div asigură o linie nouă înainte și după (similar cu paragraf); span nu
- Sintaxă: `<div> ...</div>`
`<span> ... </span>`
- Exemplu:
 - CSS: `div {background-color: #66FFFF}`
`span.color {color: red}`
 - HTML: `<div>Acest div este tratat similar cu un paragraf,dar <span class="color">acest span</span> nu este tratat ca un paragraf.</div>`

Utilizarea CSS-urilor

- Există trei moduri de utilizare a CSS-urilor:
 - Style sheet extern
 - Acesta este cel mai puternic
 - Se aplică atât la HTML cât și la XML
 - Toate elementele de sintaxă CSS pot fi utilizate
 - Style sheet intern
 - Se aplică numai la HTML, *nu* și la XML
 - Toate elementele de sintaxă CSS pot fi utilizate
 - Styles sheet inline
 - Se aplică numai la HTML, *nu* și la XML
 - **NU** toate elementele de sintaxă CSS pot fi utilizate

Style sheet-uri externe

- În cadrul documentelor HTML se specifică în cadrul elementului `<head>`:

`<link rel= "stylesheet" type="text/css" href="Style Sheet URL">`
- În cadrul unui document XML se specifică în prolog ca instrucțiune de procesare:

`<?xml-stylesheet href="Style Sheet URL" type="text/css"?>`
- Observație: "text/css" este tipul MIME

Style sheet-uri inline

- Se definesc în cadrul atributului `STYLE` care poate fi adăugat la orice element HTML:

`<html-tag STYLE="proprietate: valoare">` sau
`<html-tag STYLE=" proprietate: valoare;
proprietate: valoare; ...; proprietate: valoare ">`
- Avantaj:
 - Util atunci când dorim să facem mici schimbări ale stilului
- Dezavantaje:
 - Mixare informațiilor de afișare cu cele de HTML
 - Îngreunează citirea codului HTML
 - Nu pot fi utilizate toate elemente CSS-ului

Style sheet-uri interne

- În cadrul documentelor HTML se specifică în cadrul elementului `<head>`:
`<style TYPE="text/css">`
`<!--`
`CSS Style Sheet`
`-->`
`</style>`
- Observație: style sheet-urile interne se definesc ca și comentarii pentru a fi ascunse browserelor mai vechi care nu suportă CSS

Ordinea cascaderii

- În cadrul unui document HTML stilurile se aplică (cascadează) în următoarea ordine:
 1. Stilurile implicite ale browser-ului
 2. Style sheet-urile externe
 3. Style sheet-urile interne (definite în `<head>`)
 4. Style sheet-urile inline
- Atunci când apar conflicte, "cel mai recent" primează

Exemplu de cascada

- style sheet extern:

```
h3 { color: red;
      text-align: left;
      font-size: 8pt
    }
```
- style sheet intern:

```
h3 { text-align: right;
      font-size: 20pt
    }
```
- attribute rezultate:

```
color: red;
text-align: right;
font-size: 20pt
```

Pescurtarea proprietăților

- Frecvent, proprietăți multiple de forma:

```
h2 { font-weight: bold; font-variant: small-caps; font-size: 12pt; line-height: 14pt; font-family: sans-serif }
```


Pot fi rescrise astfel:

```
h2 { font: bold small-caps 12pt/14pt sans-serif }
```

Proprietăți și valori pentru font-uri

- font-family:
 - inherit (moștenește de la părinte)
 - Verdana, "Courier New", ... (fontul utilizat dacă există)
 - serif | sans-serif | cursive | fantasy | monospace (fonturi generice: browserul decide ce font se folosește)
- font-size:
 - inherit | smaller | larger | xx-small | x-small | small | medium | large | x-large | xx-large | 12pt
- font-weight:
 - normal | bold | bolder | lighter | 100 | 200 | ... | 700
- font-style:
 - normal | italic | oblique

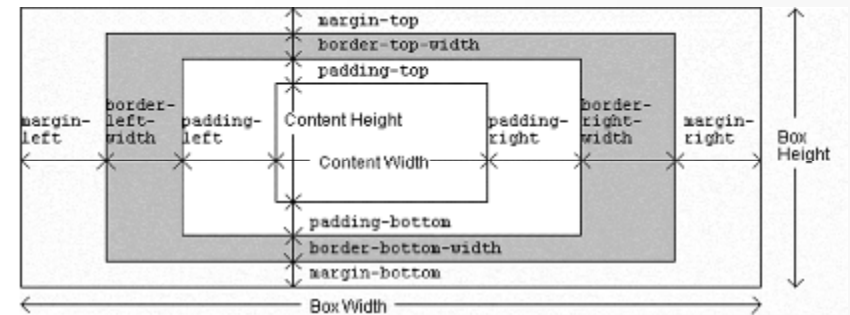
Specificarea culorilor și lungimilor

- color: și background-color:
 - aqua | black | blue | fuchsia | gray | green | lime | maroon | navy | olive | purple | red | silver | teal | white | #FF0000 | #F00 | rgb(255, 0, 0) | nume-specifice-browserelor (nerecomandate)
- Lungimile se specifică prin:
 - Lungimi relative: em, ex, %
 - Relativ la dimesiunea fontului, relative la x-height –ul fontului, procent din dimensiunea moștenită
 - Lungimi absolute: in, cm, mm, pt, pc, px
 - inchi, centimetri, milimetri, puncte, pica (1 pica = 12 points), pixeli

Specificarea alinierii textului

- text-align:
 - left | right | center | justify
- text-decoration:
 - none | underline | overline | line-through
- text-transform:
 - none | capitalize | uppercase | lowercase
- text-indent
 - *length* | 10% (indentează prima linie dintr-un text)
- white-space:
 - normal | pre | nowrap

CSS Box-model



Pseudo-clasele

- Pseudo-clasele sunt elemente a căror stare (și mod de afișare) se poate schimba în timp
- Sintaxa: *element:pseudo-class* {...}
 - `:link`
 - link care nu a fost vizitat
 - `:visited`
 - link care a fost vizitat
 - `:active`
 - link pe care facem click
 - `:hover`
 - Link deasupra căruia e mouse-ul (dar nu a fost apăsat)
- Pseudo-clasele sunt permise oriunde în selectori CSS

CSS Box-model: Exemplu

```
div {  
  width: 200px;  
  padding: 25px;  
  border: 25px solid red;  
  margin: 25px;  
}
```

Alegerea numelor

- CSS a fost elaborat pentru a separa conținutul de modul de afișare
 - Deci, numele tagurilor utilizate în HTML sau (în special) în XML trebuie să descrie *conținutul*, *și nu stilul de afișare*
- Exemplu:
 - Să presupunem că am definit stilul
span.huge {font-size: 36pt}
 - și l-am utilizat ... într-o mulțime de documente.
 - Dacă constatăm că stilul acesta de afișare este deranjant pentru utilizatori, putem schimba CSS-ul
span.huge {font-color: red}

Bibliografie

- **CSS Tutorial**, <https://www.w3schools.com/css/>
- **Cascading Style Sheets, level 1**, W3C Recommendation 17 Dec 1996, revised 11 Jan 1999, <http://www.w3.org/TR/CSS1>
- http://www.w3schools.com/css/css_syntax.asp
- <http://www.w3.org/MarkUp/Guide/Style>
- Andy Budd, Cameron Moll & Simon Collison: *CSS Mastery* (APress, 2006, ISBN 1590596145)
- **Cascading Style Sheets (CSS) by Example**, Steve Callihan

Utilitare pentru CSS-uri

- Editoare:
 - OxygenXML Editor
 - Rapid CSS editor
 - Visual Studio Code
 - Sublime Text Editor
 - Notepad ++
- Validatoare:
 - W3C CSS Validator: <http://jigsaw.w3.org/css-validator/>
 - Firebug – Plug-in pentru Firefox

Întrebări & Comentarii