

Tema: Proprietăți de stil pentru elementele flotante și pentru poziționarea elementelor

După această lecție:



- vei afla ce reprezintă "fluxul normal de prezentare al elementelor" într-un document web;
- vei cunoaște care este rolul proprietății "float" în aranjarea elementelor web în pagină, dar și a proprietății "clear";
- vei cunoaște proprietățile CSS utilizate pentru poziționarea elementelor HTML, altfel decât conform "fluxului normal de prezentare";
- vei înțelege rolul proprietății "z-index";
- vei cunoaște unele valori ce pot fi atribuite proprietății "display".

La lecția anterioară am învățat să creăm machete simple de pagini web cu ajutorul elementului DIV. Sper că ai reușit să creezi și propriile machete... La această lecție vom continua cu poziționarea elementelor în pagini.

Ordinea prezentării elementelor HTML

Ordinea prezentării elementelor în pagină este definită de **fluxul documentului HTML** sau fluxul normal al elementelor din documentul web. Aceasta înseamnă că toate elementele **de tip bloc**, atunci când nu definim nici un stil pentru ele – sunt prezentate în pagină sub formă dreptunghiulară și se succed unul după altul, de sus în jos. Iar elementele **liniare** – sunt plasate, în pagină, de sus în jos și de la dreapta la stânga (simultan, dacă reușești să-ți imaginezi). La necesitate, toate aceste elemente, sunt trecute din rând nou.

Dar, trebuie să știi că putem utiliza proprietățile CSS pentru a schimba, la necesitate fluxul normal al elementelor din documentul web.

Proprietatea "float"

Proprietatea CSS „**float**” specifică cum un anumit element poate fi „împins”, plutind la stânga sau la dreapta, permițând altor elemente să-l înconjoare. Proprietatea „**float**” este foarte des folosită atunci când se inserează imagini, dar este, utilă și atunci când se machetează paginile web. Elementele „plutesc” pe orizontală, acest lucru însemnând că un element poate fi „împins” la stânga sau la dreapta, însă **nu!** în sus sau în jos.

Elementele amplasate după elementul flotant, îl vor „cuprinde” pe acesta. Elementele din fața elementului flotant nu vor fi afectate de valorile proprietății **float**. În următorul tabel sunt prezentate valorile, ce pot fi atribuite proprietății **float**:

Valoarea	Descrierea
none	Elementul nu va fi flotant. Este valoarea implicită a proprietății
left	Elementul se deplasează spre stânga
right	Elementul se deplasează spre dreapta

Voi încerca efectul proprietății **float** asupra blocului cu imagini, din pagina creată la lecția trecută. Voi adăuga imaginilor, din **DIV**-ul cu clasa **.content**, proprietatea **float: left;**



```
.content img {
  max-width: 20%;
  height: auto;
  border: 3px dotted rgb(247, 238, 239);
  padding: 3px;
  margin: 5px;
  float: left;
}
```

Voi obține următorul rezultat:



Observă că blocul cu clasa **.footer**, care urmează, după blocul **.content** s-a ridicat, din motiv că, am menționat deja, toate blocurile care urmează după elementele flotante vor fi afectate.

Această proprietate este frecvent utilizată pentru a crea pagini-galerii cu imagini. Poți accesa link-ul http://www.w3schools.com/css/css_image_gallery.asp, pentru a vedea cum poți face și tu o galerie.

Proprietatea "clear"

Această proprietate, de obicei, se folosește împreună cu proprietatea **float**. Toate elementele care urmează după un element pentru care a fost definită proprietatea **float**, îl „înconjoară” – am menționat deja. **Clear** este folosită pentru a controla comportamentul elementelor „flotante”. În tabelul următor sunt descrise valorile posibile pentru proprietatea **"clear"**.

Valoare	Descriere
none	Valoarea implicită. Permite cuprinderea elementelor flotante pe ambele părți
left	Nu va mai cuprinde elementele flotante pe partea stângă
right	Nu va mai cuprinde elementele flotante pe partea dreaptă
both	Nu va mai cuprinde elementele flotante nici pe dreapta nici pe stângă

Ai văzut deja ce s-a întâmplat cu **DIV**-ul care urma după blocul cu imagini – s-a ridicat, încercând să "cuprindă" blocul cu clasa **.content**. În continuare voi adăuga o singură proprietate **DIV**-ului cu clasa **.footer**:



```
.footer{
  background-image: url("../images/logo.png");
  background-repeat: repeat-x;
  background-size: 5%;
  clear: both;
}
```

Vezi ce s-a întâmplat în urma interpretării – **DIV**-ul cu clasa **.footer**, a coborât ascultător la locul lui:



Proprietăți de stil pentru poziționarea elementelor HTML

Proprietățile de poziționare în CSS permit gestionarea amplasării unui element HTML. Este posibilă plasarea unui element în spatele altuia, sau specificarea modalității de afișare a unui anumit element care are conținutul foarte mare. Elementele pot fi poziționate utilizând proprietățile **top**, **bottom**, **left** și **right**. Aceste proprietăți, însă, nu vor fi funcționale dacă proprietatea **position** nu va fi setată prima. De asemenea, contează mult metoda de poziționare. Există **patru metode** diferite **de poziționare**. Le vom analiza succesiv pe toate patru.

position: static

Elementele HTML sunt implicit poziționate static. Un element poziționat static, este mereu amplasat, ținând cont de fluxul normal al elementelor din pagină. Elementele poziționate static nu vor fi afectate de proprietățile **top**, **bottom**, **left** și **right**.

position: fixed

Un element cu poziția fixată va fi poziționat **în raport cu fereastra browser-ului**. Acest element nu se va mișca chiar dacă conținutul ferestrei defilează – în sus sau în jos. Elementele cu poziția fixată nu țin cont de fluxul normal de afișare pe pagină – nu contează ce alte elemente mai sunt în pagină. Elementele cu poziția fixată se pot suprapune peste alte elemente.

Să analizăm un exemplu. Vreau ca blocul ce conține meniul, din pagina făcută la lecția trecută, să nu defileze ca până acum. Voi dubla numărul de poze din pagină ca să se vadă cum defilează bara cu meniul.

(Vezi filmulețul anexat **defilare.mp4**)

Sper ca ai observat că acum defilează bara cu meniul...

Acuma voi adăuga **DIV**-ului cu meniul câteva proprietăți CSS, pentru a-l fixa în partea de sus a ecranului. Deci, voi adăuga în fișierul de stiluri:



```
.menu{  
    position:fixed;  
    top:0;  
    left:0;  
}
```

Dar acum apare un neajuns – blocul cu pozele "nu vede" blocul cu meniul și a alunecat sub el – textul este incomplet, din motiv că e sub blocul cu meniul:



Pentru a atenua acest neajuns vom adăuga o proprietate și blocului cu clasa **.content**:



```
.content{
  margin-top: 50px;
}
```

Am evaluat înălțimea blocului cu meniul și am coborât tot conținutul lui cu 50px. Acum blocul cu meniul este fixat și nu defilează, iar textul din următorul bloc este vizibil. (Vezi filmulețul **fixed.mp4**)

position: relative

Un element poziționat relativ este **poziționat în raport cu poziția sa normală** în flux. Conținutul elementelor relativ-poziționate pot fi mutate și se pot suprapune peste alte elemente, însă spațiul rezervat pentru acest element în fluxul normal, se păstrează după el. Elementele poziționate relativ sunt adesea utilizate ca blocuri-containere pentru elemente poziționate absolut.

position: absolute

Un element în poziție absolută este poziționat **în raport cu primul element-părinte**, care are o altă poziție decât statică. În cazul în care nu se găsește nici un astfel de element-părinte, blocul se va poziționa în raport cu tag-ul-rădăcină - **<html>**.

Elementele poziționate „absolut” sunt îndepărtate din fluxul normal. Documentul și alte elemente, se comportă ca și cum elementul poziționat absolut nu ar exista. Elementele poziționate „absolut” se pot suprapune peste alte elemente HTML. Implicit elementele **“absolute”** se poziționează într-un sistem de coordonate bazat pe fereastra browserului. Însă, sistemul de coordonate poate fi schimbat. Dacă un anumit element-părinte al unui element poziționat absolut, este poziționat relativ, atunci elementul-urmas se va poziționa în raport cu elementul-părinte. Dacă sunt mai multe elemente-părinte poziționate relativ, atunci elementul-urmas poziționat absolut se va poziționa în raport cu elementul părinte cel mai apropiat de el (părintele direct). Să analizăm câteva exemple pentru claritate.

Voi crea câteva blocuri, incluse. Ultimul inclus va avea poziția absolută, dar niciunul din blocurile lui părinte nu vor avea poziționare relativă – adică poziționarea blocului absolut se va face implicit – în raport cu fereastra browserului. Iată codul HTML:



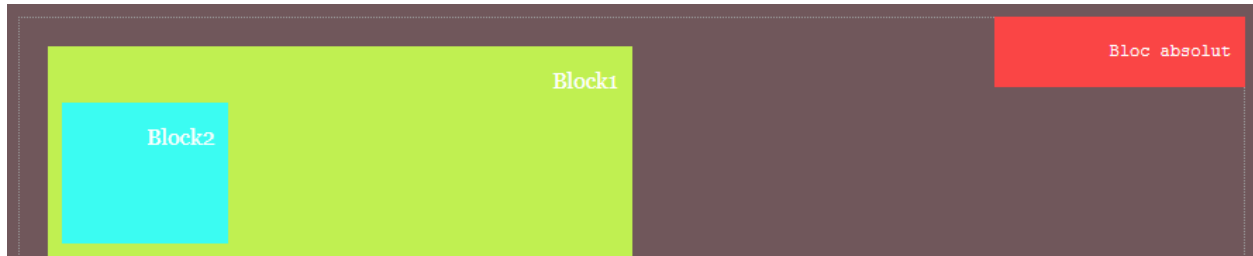
```
<div class="block1">Block1
  <div class="block2">Block2
    <div class="block_abs">
      Bloc absolut
    </div>
  </div>
</div>
```

Și iată stilurile:

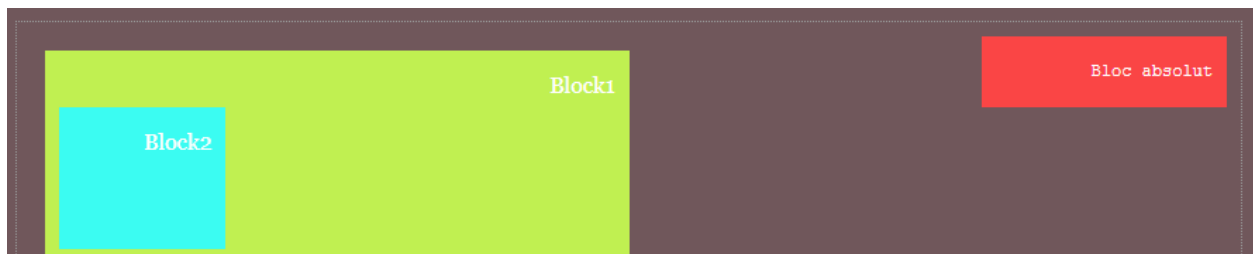


```
body{
  font: 16px/30px Georgia, serif;
  color: white;
  background-color: rgba(77, 46, 51, 0.8);
  margin: 10px;
  padding: 20px;
  border: 1px dotted #999;
}
.block1 {
  min-height: 150px;
  width:50%;
  padding: 10px;
  background: rgb(192, 240, 81);
  text-align: right;
}
.block2 {
  min-height: 100px;
  width:30%;
  padding: 10px;
  background: rgb(59, 252, 242);
  text-align: right;
}
.block_abs {
  position: absolute;
  top: 10px;
  right: 10px;
  padding: 10px;
  font-family: "Courier New", monospace;
  font-size: 12px;
  background: rgb(250, 69, 69);
  min-height: 50px;
  width:20%;
  text-align: right;
}
```

Observă cum s-a poziționat blocul cu poziționare absolută – special i-am setat elementului **BODY** un **border**.



Dacă voi adăuga: `position: relative;` pentru **BODY** - deja se poziționează în raport cu **BODY** – observi cum a coborât?



Iar dacă voi adăuga `position: relative;` lui **.block2** – vezi cum se schimbă poziția blocului absolut – a migrat în blocul cu clasa **.block2**:



În practică, poziționarea absolută adesea se utilizează în scopuri decorative, atunci când este necesară plasarea într-o anumită poziție a unor iconițe, elemente de control și alte "frumuseți".

Am decis să realizez un exemplu în baza pozelor – voi adăuga prețul produselor pe poza cu cafea. Am inclus un bloc cu elementul **DIV**, ce va avea poziție relativă. Imaginea am plasat-o pe fundalul acestui bloc, iar în colțul din dreapta-jos, voi adăuga prețul, utilizând un alt **DIV** poziționat absolut. Codul HTML:



```
<div class="card1">
  <div class="price">25 lei</div>
</div>
```


Stilurile și setarea pozițiilor celor două blocuri:



```
.card1{
  max-width: 30%;
  height: 33%;
  border: 5px dotted rgb(235, 202, 207);
  padding: 3px;
  margin: 5px;
  position: relative;
  background-image: url("../images/cappuccino.jpeg");
  background-position: 50% 50%;
  background-size: cover;
}
.card1 .price{
  width: 25%;
  position: absolute;
  bottom: 2%;
  right: 2%;
  background-color: rgba(148, 136, 137, 0.5);
  padding: 5px;
}
```

Observă, că primul bloc cu clasa **.card1**, are poziție relativă, iar celălalt cu clasa **.price**, va avea poziția absolută. Cu proprietățile **bottom** și **right** – am setat poziția în interiorul celuilalt bloc. Rezultatul va unul pe așteptările noastre:



Proprietatea "z-index"

Când elementele sunt poziționate în afara fluxului normal – **position: absolute** sau **position: relative** sau **position: fixed** – ele se pot suprapune peste alte elemente.

Proprietatea „**z-index**” din CSS specifică ordinea în stiva prezentării elementelor – elementul care ar trebui să fie plasat în față sau în spatele celorlalte elemente.

Un element poate avea un ordin pozitiv sau negativ în stivă. Un element cu un ordin mai mare în stivă este întotdeauna în față unui element cu un ordin mai mic

Proprietatea **z-index** funcționează doar pentru elementele poziționate absolut, relativ sau fix. Încearcă să adaugi **DIV**-ului cu clasa **.price**, din exemplul anterior, proprietatea **z-index: -1;**

Ai înțeles ce s-a întâmplat? A dispărut prețul? Nu... Blocul cu prețul stă acum în spatele blocului cu clasa **.card1...** 😊

Proprietatea „display”

Proprietatea „display” este folosită pentru a defini tipul boxei în care se va afișa un element HTML. Această proprietate este suportată de majoritatea browserelor (parțial, unele valori ale proprietății **display**, nu sunt suportate de Internet Explorer).

Sintaxa recomandată: **display: valoare;**

Dispune de foarte multe valori posibile. Accesați pentru a vedea toate valorile pentru această proprietate: http://www.w3schools.com/cssref/pr_class_display.asp

După cum am menționat deja, proprietatea are foarte multe posibilități de prezentare a unui bloc, inclusiv și sub formă de tabel – fără a utiliza elementul **TABLE**, ci doar elementul **DIV** și proprietatea **display: table;**

Valoarea **none** a proprietății **display** se folosește foarte des la construirea meniurilor expandabile. Cu ajutorul acestei valori, un element poate fi ascuns, de parcă el nici nu ar exista – nu ocupă spațiu pe pagină și nu se vede.

Voi realiza doar un mic exemplu, pentru a vedea efectul proprietății **display**.

Dacă pentru blocul cu meniul:



```
<div class="menu">
  <span>Coffee time</span>
  <a href="#unu">Despre noi</a>
  <a href="#doi">Produse</a>
  <a href="#trei">Contacte</a>
</div>
```

Voi scrie stilul:



```
.menu a{
  display: inline;
```

```
}  
.menu{  
  height: auto;  
}
```

...atunci voi obține următorul rezultat:



Iar dacă voi scrie stilul:



```
.menu a{  
  display: block;  
}
```

...atunci voi obține acest rezultat:



Interesant, nu? Poți încerca valoarea **none** – să vezi efectul... Vom mai reveni la **display: flex** în una din lecțiile următoare...

Să recapitulăm cele învățate în cadrul acestei lecții:



- "fluxul normal de prezentare al elementelor" reprezintă ordinea de prezentare ale elementelor în pagina web, exact la fel cum au fost acestea înscrise în documentul web. Elementele de tip bloc sau *inline*, precum DIV IMG sau P, se vor prezenta de sus în jos, și de la stânga la dreapta;
- "fluxul normal de prezentare al elementelor" poate fi schimbat dacă sunt utilizate proprietățile de poziționare sau **z-index**;
- poziționarea absolută a elementelor, de obicei, se combină cu elementele poziționate relativ, pentru a crea efecte vizuale mai deosebite;

- proprietatea *"display"* are foarte multe posibilități de prezentare a elementelor: pe orizontală, sub formă de bloc, sub formă de grilă, sub formă de tabel, poate ascunde unele elemente ca și cum ele nici nu ar exista în pagină etc. Important este să fie specificată valoarea potrivită pentru această proprietate.

Încheiem aici lecția, dar data viitoare numaidecât vom continua să studiem posibilitățile CSS.