

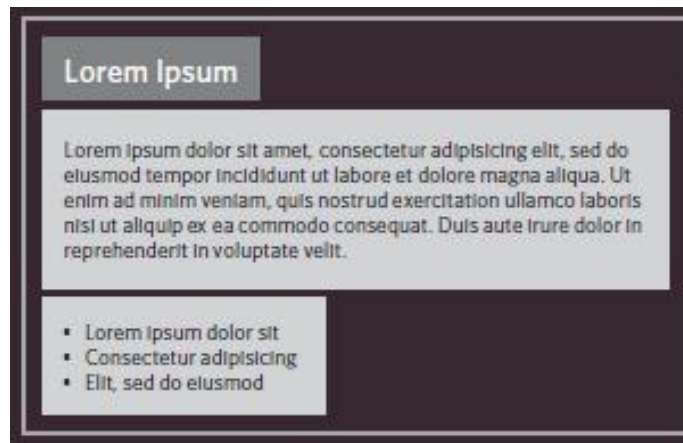
LAYOUT

Concepte cheie in pozitionarea elementelor

Elemente la nivel de bloc

Incep pe o linie noua.

Exemple: <h1>, <p>,



Elemente inline

Se afla pe acelasi rand cu elementele invecinate.

Exemple: , , <i>



- ❑ CSS trateaza fiecare element HTML ca si cum ar fi in propriul container.
- ❑ Acest container poate fi:
 - un container la nivel de bloc.
 - un container inline.
- ❑ Container la nivel de bloc:
 - Containerele la nivel de bloc incep pe o linie noua si reprezinta elementele de baza ale unui layout, in timp ce elementele inline sunt pozitionate pe acelasi rand cu elementele invecinate.
 - Se poate controla dimensiunea spatiului ocupat de un container prin setarea latimii (width) acestuia (de asemenea se poate seta si inaltimea).
 - Pentru a separa containerele se pot utiliza borduri, margini, padding-uri si culori de fundal.

Pozitionari

- ❑ CSS pune la dispozitie proprietatea `position` prin intermediul careia *putem modifica layout-ul* unei pagini web. Exista 4 tipuri de pozitionari: *fixa, relativa, absoluta si statica.*
- ❑ Pozitionarea statica
 - Pozitionarea statica este *pozitionarea default (flow-ul normal)*, ceea ce inseamna ca elementele se pozitioneaza in functie de locul liber si in functie de ordine, dupa citirea normala a codului html pe o pagina. *Aceasta este interpretarea obisnuita a browserului.*

CSS

```
#div-1{  
    position: static;  
}
```

```
id = div-before  
id = div-1  
id = div-1a  
Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Integer pretium dui sit amet felis. Integer sit amet diam.  
Phasellus ultrices viverra velit.  
id = div-1b  
Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Integer pretium dui sit amet felis. Integer sit amet diam.  
Phasellus ultrices viverra velit. Nam mattis, arcu ut  
bibendum commodo, magna nisi tincidunt tortor, quis  
accumsan augue ipsum id lorem.  
id = div-1c  
id = div-after
```

Pozitionarea relativa

- Pozitionarea relativa reprezinta o pozitionare care ia in considerare pozitia originala (pozitia default) a elementului.
- In cazul in care se seteaza `position: relative` atunci se pot utiliza proprietatile `top` sau `bottom` si `left` sau `right` pentru a pozitiona un element relativ la POZITIA ANTERIOARA (pozitia originala in care ar fi fost afisat in mod normal).
- In cazul in care nici unul din attributele de pozitionare (`top`, `right`, `bottom` sau `right`) nu este folosit atunci `position: relative` nu va avea niciun efect in pozitionarea elementului.

CSS

```
#div-1{  
    position: relative;  
    top: 20px;  
    left:-40px;}
```

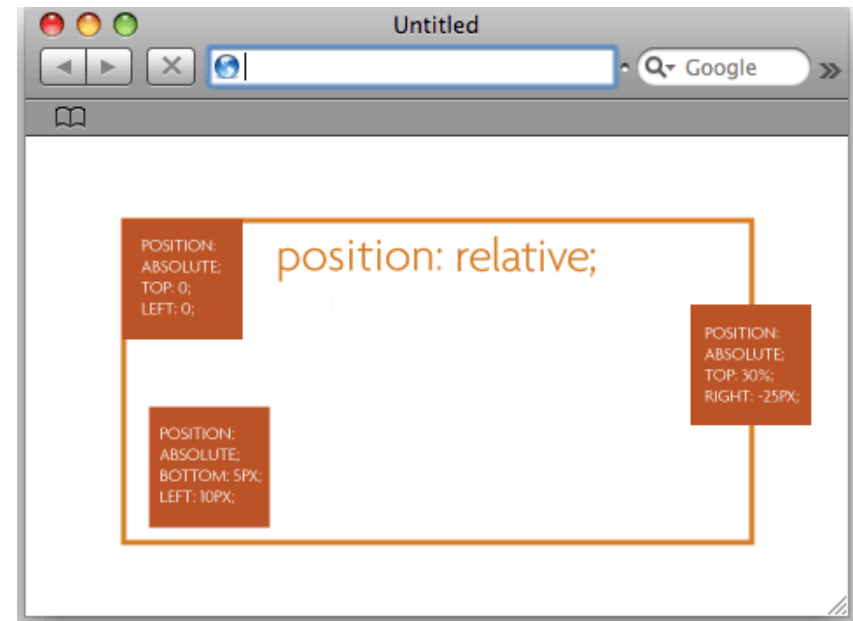
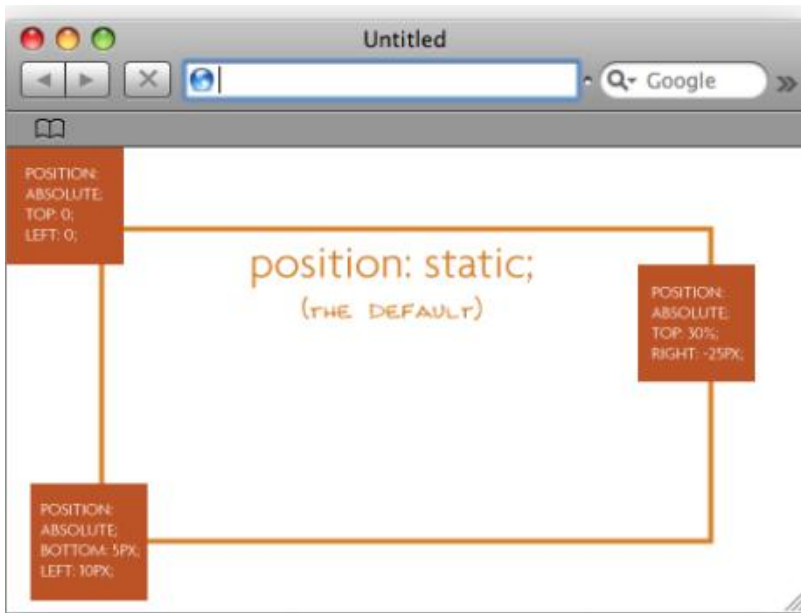
- `div-1` este pozitionat 20px mai jos si 40px mai la stanga fata de pozitia anterioara.
- elementul `div-after` nu a fost miscat atunci cand `div-1` a fost repozionat.

id = div-before

```
id = div-1  
id = div-1a  
Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Integer pretium dui sit amet felis. Integer sit amet diam.  
Phasellus ultrices viverra velit.  
id = div-1b  
Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Integer pretium dui sit amet felis. Integer sit amet diam.  
Phasellus ultrices viverra velit. Nam mattis, arcu ut  
bibendum commodo, magna nisi tincidunt tortor, quis  
accumsan augue ipsum id lorem.  
id = div-1c
```

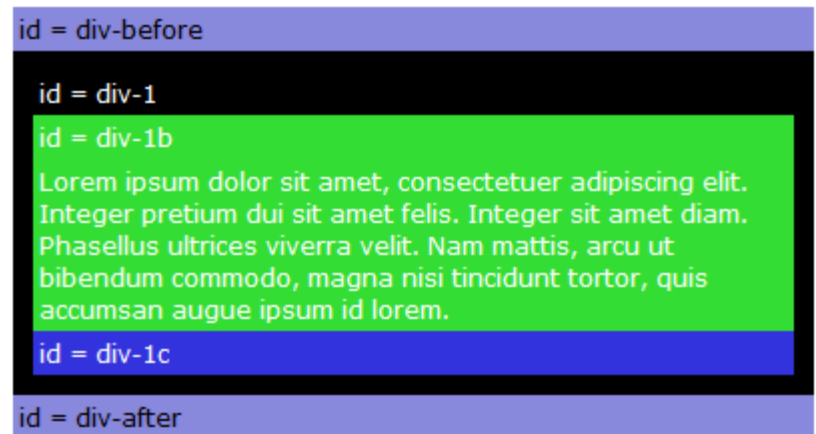
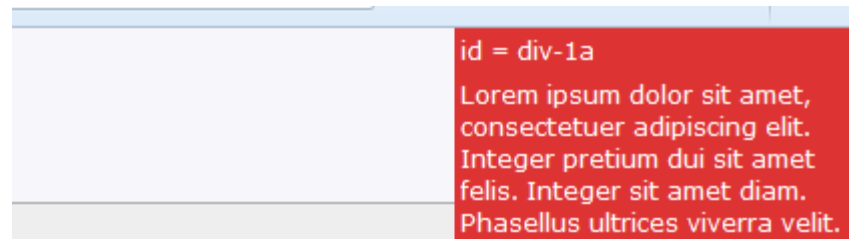
Pozitionarea absoluta

- Pozitionarea absoluta este o pozitionare folosita relativ des. O pozitionare absoluta (`position:absolute`) **inseamna pozitionarea elementului relativ la ELEMENTUL PARINTE** pentru care s-a definit deja o pozitionare. Se folosesc proprietatile `top`, `left`, `bottom` sau `right` pentru a seta pozitia elementului.
- Daca elementul parinte nu a fost pozitionat, adica are o *pozitionare statica*, atunci elementul parinte este considerat tagul `<html>`, adica pagina in sine.
- **Atentie!** – pentru ca **pozitionarea absoluta** sa functioneze corect, elementul parinte trebuie sa aiba **proprietatea de pozitionare definita**, si nu are voie sa fie **pozitionat static**.



CSS

```
#div-1a{  
    position: absolute;  
    top: 0;  
    left: 0;  
    width: 200px}
```



Pozitionare relativa si absoluta

- Daca setam o **pozitionare relativa** pentru `div-1`, atunci orice element din cadrul lui `div-1` va fi pozitionate relativ la `div-1`.
- Daca dupa aceea setam o **pozitionare absoluta** pentru `div-1a`, atunci `div-1a` va fi mutat in coltul SUS-DREAPTA al elementului `div-1`.

CSS

```
#div-1 {  
    position: relative; }  
#div-1a {  
    position: absolute;  
    top: 0;  
    right: 0;  
    width: 200px; }
```

id = div-before

id = div-1

id = div-1b

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer pretium dui sit amet felis. Integer sit amet diam. Phasellus ultrices viverra velit. bibendum commodo, magna nisi tincidunt tortor, quis accumsan augue ipsum id lorem.

id = div-1c

id = div-after

id = div-1a

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer pretium dui sit amet felis. Integer sit amet diam. Phasellus ultrices viverra velit.

Doua coloane cu pozitionare absoluta

- Se poate genera un layout cu doua coloane folosind pozitionarile relativa si absoluta.
- Un avantaj al utilizarii pozitionarii absolute este acela ca putem pozitiona elementele in pagina in orice ordine indiferent de ordinea in care apar in codul HTML. In exemplul de mai jos *div-1b* este plasat inaintea lui *div-1a*.
- Dar dupa cum se poate observa celelalte elemente HTML au fost ACOPERITE de elementele cu pozitionare absoluta.

CSS

```
#div-1 {  
    position: relative; }  
#div-1a {  
    position: absolute;  
    top: 0;  
    right: 0;  
    width: 200px; }  
#div-1b {  
    position: absolute;  
    top: 0;  
    left: 0;  
    width: 200px; }
```

id = div-before

id = div-1b

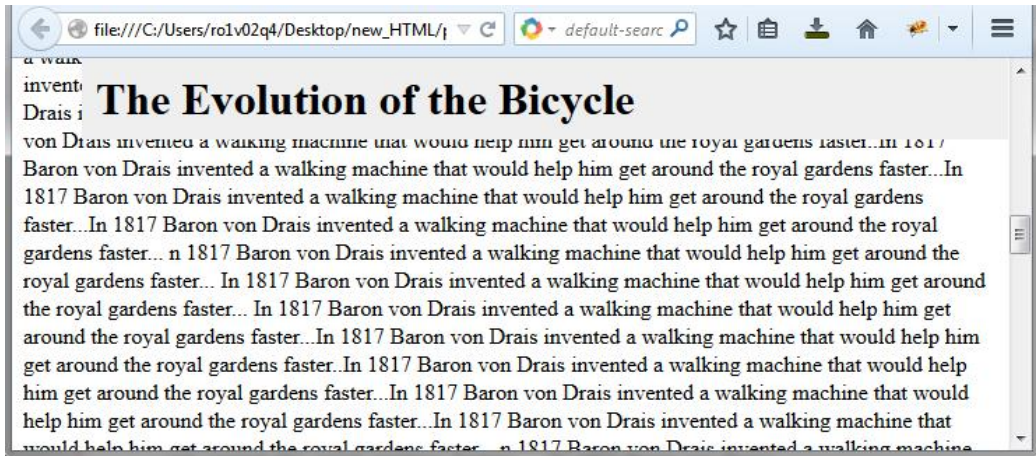
Lorem ipsum dolor sit amet,
consectetuer adipiscing elit.
Integer pretium dui sit amet
felis. Integer sit amet diam.
Phasellus ultrices viverra velit.
Nam mattis, arcu ut bibendum
commodo, magna nisi tincidunt
tortor, quis accumsan augue
ipsum id lorem.

id = div-1a

Lorem ipsum dolor sit amet,
consectetuer adipiscing elit.
Integer pretium dui sit amet
felis. Integer sit amet diam.
Phasellus ultrices viverra velit.

Pozitionare fixa

- Pozitionarea fixa este un tip de pozitionare absoluta care este folosita mai rar, dar uneori este foarte utila. Daca asupra unui element se aplica o pozitionare fixa atunci elementul respectiv va fi pozitionat **relativ la FEREASTRA BROWSERULUI**, si nu se va misca chiar daca vom face scroll la pagina.
- Se folosesc proprietatile **top**, **left**, **bottom** sau **right** pentru a seta pozitia elementului fata de fereastra browserului.
- Elementul la care adaugam pozitionare fixa nu va lua parte din flow-ul normal al paginii, in consecinta nu va afecta pozitia celorlalte elemente din pagina (acestea se vor comporta ca si cum acest element nu ar fi acolo).



HTML

```
<body>
  <h1>The Evolution of the Bicycle</h1>
  <p class="example">In 1817 Baron von Drais
    invented a walking machine that would help him
    get around the royal gardens faster...</p>
</body>
```

CSS

```
h1 {
  position: fixed;
  top: 0px;
  left: 50px;
  padding: 10px;
  margin: 0px;
  width: 100%;
  background-color: #efefef;}
p.example {
  margin-top: 100px;}
```

- In exemplul alaturat, heading-ul a fost pozitionat in partea stanga-sus a ferestrei browserului. Atunci cand userul face scroll la pagina, paragrafele dispar in spatele heading-ului.
- Astfel elementul `<p>` ignora spatiul pe care elementul `<h1>` il ocupa.

Elemente flotante

- Pozitionarea absoluta nu functioneaza pentru coloanele cu inaltime variabila.
- O solutie o reprezinta utilizarea proprietatii `float`.
- Proprietatea `float` permite modificarea pozitiei unui element si plasarea acestuia cat mai la stanga (`float: left`) sau dreapta (`float: right`) in interiorul elementului parinte.
- Orice alt element care se afla in interiorul elementului parinte va fi aranjat in jurul elementului flotant (div-1b).
- Atunci cand se utilizeaza proprietatea `float`, trebuie sa se utilizeze si proprietatea `width` pentru a indica latimea elementului flotant; **in caz contrar rezultatele vor fi inconsistente** deoarece containerul va ocupa intreaga latime a paginii (la fel ca in cazul in care nu s-ar folosi proprietatea `float`).

CSS

```
#div-1a {  
    float: left;  
    width: 200px;}
```

id = div-before

id = div-1

id = div-1a

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Integer pretium dui sit amet
felis. Integer sit amet diam.
Phasellus ultrices viverra velit.

id = div-1b

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Integer pretium dui sit amet
felis. Integer sit amet diam.
Phasellus ultrices viverra

velit. Nam mattis, arcu ut

bibendum commodo, magna nisi tincidunt tortor, quis
accumsan augue ipsum id lorem.

id = div-1c

id = div-after

Elemente flotante – intrebare?

Rezultat cu css-ul initial

The Evolution of the Bicycle

"Life is like riding a bicycle. To keep your balance you must keep moving." - Albert Einstein

In 1817 Baron von Drais invented a walking machine that...

Ce trebuie adaugat la css pentru a obtine un rezultat similar celui din figura?

The Evolution of the Bicycle

In 1817 Baron von Drais invented a walking machine that would help him get around the royal gardens faster: two same-size in-line wheels, the front one steerable, mounted in a frame ...

In 1817 Baron von Drais invented a walking machine that would help him get around the royal gardens faster: two same-size in-line wheels, the front one steerable, mounted in a frame In 1817 Baron von Drais invented a ...

*"Life is like riding a bicycle.
To keep your balance you
must keep moving." - Albert
Einstein*

HTML

```
<h1>The Evolution of the Bicycle</h1>
<blockquote>"Life is like riding a bicycle.
To keep your balance you must keep
moving." - Albert Einstein</blockquote>
<p>In 1817 Baron von Drais invented a
walking machine that... </p>
```

CSS

```
blockquote {
  font-size: 130%;
  font-style: italic;
  font-family: Georgia, Times, serif;
  margin: 0px 0px 10px 10px;
  padding: 10px;
  border-top: 1px solid #665544;
  border-bottom: 1px solid #665544;}
```

Elemente flotante - **rezolvare**

CSS

```
blockquote {  
    float: right;  
    width: 275px;  
    font-size: 130%;  
    font-style: italic;  
    font-family: Georgia, Times, serif;  
    margin: 0px 0px 10px 10px;  
    padding: 10px;  
    border-top: 1px solid #665544;  
    border-bottom: 1px solid #665544;}
```

Utilizarea proprietatii **float** pentru a plasa containerele unul langa altul

- Foarte multe structuri plaseaza containerele unul langa celalalt folosind proprietatea **float**.
- Atunci cand elementele sunt flotante, **inaltimea** containerelor poate afecta pozitia in care urmatorul element este plasat.
- Exemplul alaturat prezinta 6 paragrafe, fiecare avand setate proprietatile **width** si **float**.
- Cel de-al patrulea paragraf nu este plasat in partea stanga sub paragraful 1 (asa cum ar fi de asteptat), ci este plasat sub cel de-al treilea paragraf.
- Motivul este acela ca cel de-al patrulea paragraf are spatiu sa inceapa sub cel de-al treilea paragraf, dar nu se poate duce mai departe, catre stanga, deoarece cel de-al doilea paragraf este in drumul lui.

The Evolution of the Bicycle

Paragraf 1 Paragraf 1 Paragraf 1
Paragraf 1 Paragraf 1 Paragraf 1
Paragraf 1 Paragraf 1 Paragraf 1

Paragraf 2 Paragraf 2 Paragraf 2
Paragraf 2 Paragraf 2 Paragraf 2
Paragraf 2 Paragraf 2 Paragraf 2
Paragraf 2 Paragraf 2 Paragraf 2
Paragraf 2 Paragraf 2 Paragraf 2
Paragraf 2 Paragraf 2 Paragraf 2

Paragraf 3 Paragraf 3 Paragraf 3
Paragraf 3 Paragraf 3 Paragraf 3
Paragraf 3 Paragraf 3 Paragraf 3
Paragraf 3 Paragraf 3 Paragraf 3

Paragraf 4 Paragraf 4 Paragraf 4
Paragraf 4 Paragraf 4 Paragraf 4
Paragraf 4 Paragraf 4 Paragraf 4
Paragraf 4 Paragraf 4 Paragraf 4
Paragraf 4 Paragraf 4 Paragraf 4
Paragraf 4 Paragraf 4 Paragraf 4

Paragraf 5 Paragraf 5 Paragraf 5
Paragraf 5 Paragraf 5 Paragraf 5
Paragraf 5 Paragraf 5 Paragraf 5
Paragraf 5 Paragraf 5 Paragraf 5

Paragraf 6 Paragraf 6 Paragraf 6
Paragraf 6 Paragraf 6 Paragraf 6
Paragraf 6 Paragraf 6 Paragraf 6
Paragraf 6 Paragraf 6 Paragraf 6

Utilizarea proprietatii `float` pentru a plasa containerele unul langa altul

Solutii:

- ✓ una din solutii este aceea de a seta `inaltimea` paragrafelor sa fie egala cu cea a *celui mai inalt paragraf* (dar aceasta solutie este folosita foarte rar deoarece lungimea textului fiecarui paragraf poate varia foarte mult)
- ✓ se utilizeza proprietatea `clear` (varianta preferata).

HTML

```
<h1>The Evolution of the Bicycle</h1>
```

```
<p>Paragraf 1 Paragraf 1 Paragraf 1 Paragraf 1  
Paragraf 1 Paragraf 1 Paragraf 1 Paragraf 1 Paragraf 1  
Paragraf 1</p>
```

CSS

```
body {  
    width: 750px;  
    font-family: Arial, Verdana, sans-serif;  
    color: #665544;}
```

```
p {  
    width: 230px;  
    float: left;  
    margin: 5px;  
    padding: 5px;  
    background-color: #efefef;}
```


Proprietatea clear

- Proprietatea `clear`, atunci cand este folosita asupra unui container, specifica faptul ca niciun alt element nu va atinge partea *dreapta* sau *stanga* a acestuia. Poate avea urmatoarele valori:

- `left` – partea *stanga* a containerului trebuie sa nu atinga alte elemente care apar in acelasi element parinte.
- `right` – partea *dreapta* a containerului trebuie sa nu atinga alte elemente care apar in acelasi element parinte.
- `both` – nici partea *dreapta* si nici partea *stanga* a containerului nu vor avea evenimente laterale in interiorul elementului parinte.
- `none` – in orice parte a elementului se poate afla un alt element.

HTML

```
<p class="clear">Paragraf 4 Paragraf 4....</p>
```

CSS

```
.clear {  
    clear: left;}  
}
```

The Evolution of the Bicycle

Paragraf 1 Paragraf 1 Paragraf 1
Paragraf 1 Paragraf 1 Paragraf 1
Paragraf 1 Paragraf 1 Paragraf 1
Paragraf 1 Paragraf 1 Paragraf 1

Paragraf 2 Paragraf 2 Paragraf 2
Paragraf 2 Paragraf 2 Paragraf 2
Paragraf 2 Paragraf 2 Paragraf 2
Paragraf 2 Paragraf 2 Paragraf 2
Paragraf 2 Paragraf 2 Paragraf 2
Paragraf 2 Paragraf 2 Paragraf 2
Paragraf 2 Paragraf 2 Paragraf 2

Paragraf 3 Paragraf 3 Paragraf 3
Paragraf 3 Paragraf 3 Paragraf 3
Paragraf 3 Paragraf 3 Paragraf 3
Paragraf 3 Paragraf 3 Paragraf 3

Paragraf 4 Paragraf 4 Paragraf 4
Paragraf 4 Paragraf 4 Paragraf 4
Paragraf 4 Paragraf 4 Paragraf 4
Paragraf 4 Paragraf 4 Paragraf 4
Paragraf 4 Paragraf 4 Paragraf 4
Paragraf 4 Paragraf 4 Paragraf 4
Paragraf 4 Paragraf 4 Paragraf 4

Paragraf 5 Paragraf 5 Paragraf 5
Paragraf 5 Paragraf 5 Paragraf 5
Paragraf 5 Paragraf 5 Paragraf 5
Paragraf 5 Paragraf 5 Paragraf 5
Paragraf 5 Paragraf 5 Paragraf 5

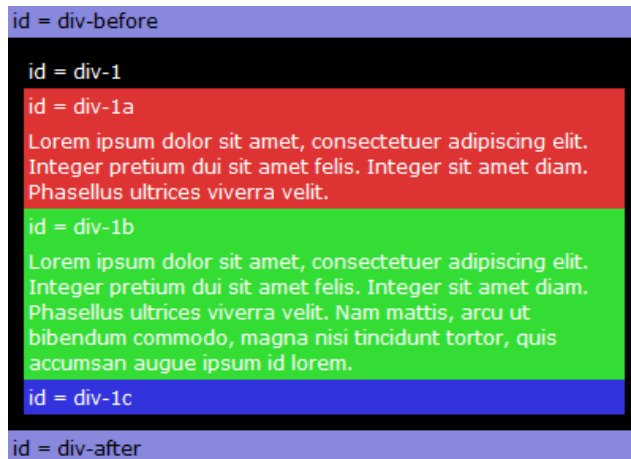
Paragraf 6 Paragraf 6 Paragraf 6
Paragraf 6 Paragraf 6 Paragraf 6
Paragraf 6 Paragraf 6 Paragraf 6
Paragraf 6 Paragraf 6 Paragraf 6

Coloane flotante – intrebare?

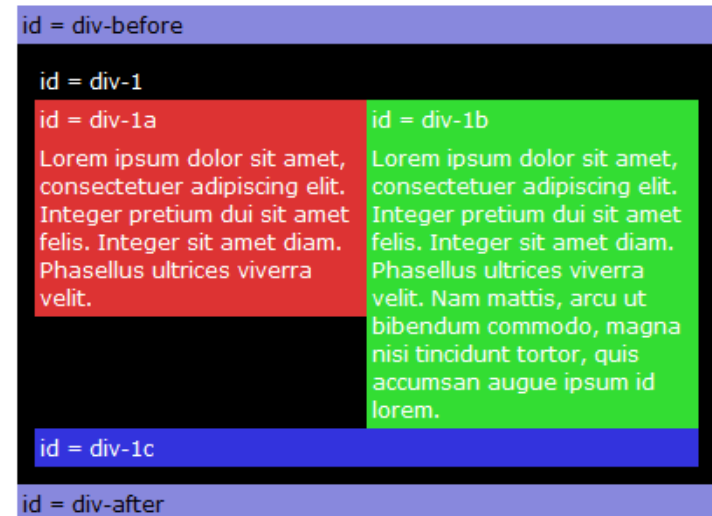
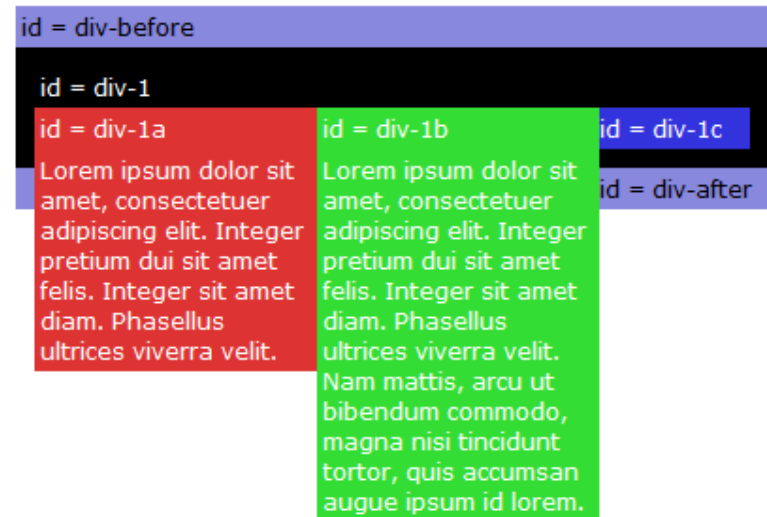
- Daca utilizam `float:left` pentru o coloana, si apoi si pentru cea de a doua coloana atunci cele doua coloane vor ajunge una langa alta.

Pornind de la aceasta figura, care sunt regulile CSS pentru a obtine rezultatele din celelalte 2 figuri?

CSS ??



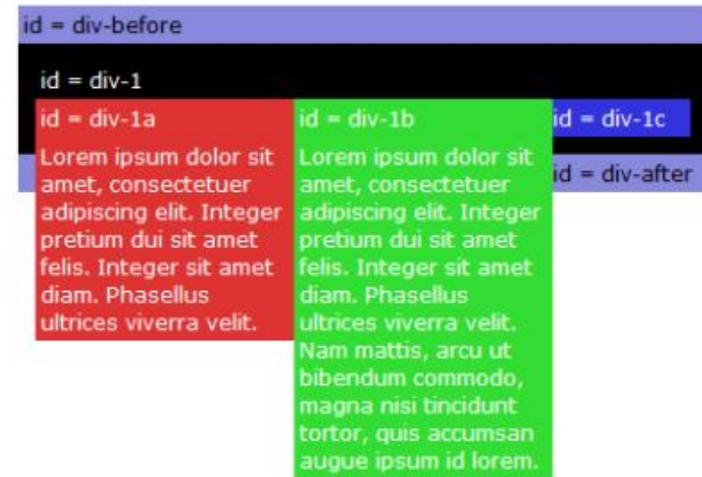
CSS ??



Coloane flotante - rezolvare

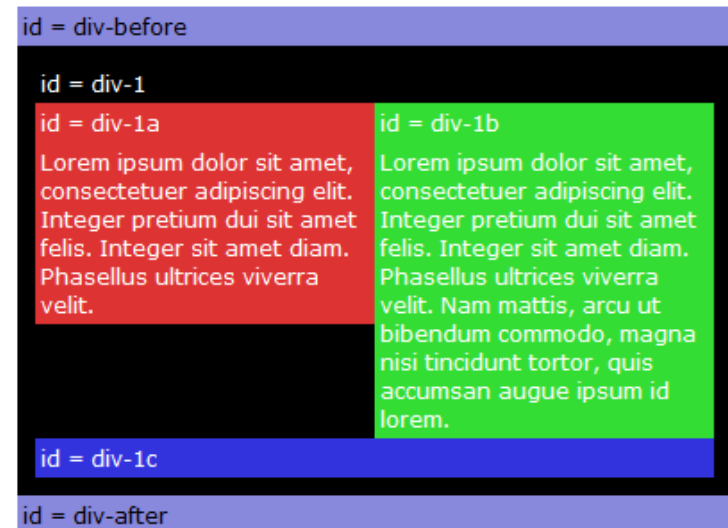
CSS

```
#div-1a {  
    float:left;  
    width:150px; }  
  
#div-1b {  
    float:left;  
    width:150px; }
```



CSS

```
#div-1a {  
    float:left;  
    width:150px; }  
  
#div-1b {  
    float:left;  
    width:150px; }  
  
#div-1c {  
    clear: both; /*varianta 1*/  
    clear: left ; /* varianta 2*/  
}
```



Crearea unor structuri cu mai multe coloane folosind elemente float

- Multe pagini web folosesc o structura compusa din coloane multiple. Fiecare coloana este reprezentata printr-un element `<div>`.
- Pentru a pozitiona coloanele unele langa altele se utilizeaza urmatoarele trei proprietati:
 - `width` – seteaza latimea coloanei.
 - `float` – pozitioneaza coloanele unele langa altele.
 - `margin` – introduce un spatiu intre coloane.
- **Exemplul** prezinta o structura cu 2 coloane => folosim 2 elemente `<div>`,
 - un `<div>` pentru continutul principal al paginii
 - si celalalt pentru explicatii suplimentare.
- In interiorul elementelor `<div>` se pot afla titluri, imagini si chiar si alte elemente `<div>`.

The Evolution of the Bicycle

The First Bicycle

In 1817 Baron von Drais invented a walking machine that would help him get around the royal gardens faster: two same-size ...

Further Innovations

In 1817 Baron von Drais invented a walking machine that would help him get around the royal gardens faster: two same-size ...

Bicycle Timeline

- 1817: Draisienne
- 1865: Velocipede
- 1870: High wheel-bicycle
- 1876: High wheel safety
- 1885: Hard tired safety
- 1888: Penumatica safety

Crearea unor structuri cu mai multe coloane folosind elemente float

HTML

```
<div class="column1of2">  
  <h3>The First Bicycle</h3>  
  <p>In 1817 Baron von Drais invented a walking  
  machine that would help him get around the  
  royal gardens faster: two same-size ...</p>  
</div>
```

```
<div class="column2of2">  
  <h3>Bicycle Timeline</h3>  
  <ul>  
    <li>1817: Draisienne</li>  
    <li>1865: Velocipede</li>  
    <li>1870: High wheel-bicycle</li>  
    <li>1876: High wheel safety</li>  
    <li>1885: Hard tired safety</li>  
    <li>1888: Penumatica safety</li>  
  </ul>  
</div>
```

CSS

```
.column1of2 {  
  float: left;  
  width: 620px;  
  margin: 10px; }
```

```
.column2of2 {  
  float: left;  
  width: 300px;  
  margin: 10px; }
```

Intrebare?

- Ce trebuie sa facem ca sa obtinem o structura ca in figura alaturata (adica 3 coloane alaturate)?

The Evolution of the Bicycle

The First Bicycle

In 1817 Baron von Drais invented a walking machine that would help him get around the royal gardens faster: two same-size in-line wheels, the front one steerable, mounted in a frame upon which you straddled. The device was propelled by pushing your feet against the ground, thus rolling yourself and the device forward in a sort of gliding walk.

The machine became known as the Draisienne (or "hobby horse"). It was made entirely of wood. This enjoyed a short lived popularity as a fad, not being practical for transportation in any other place than a well maintained pathway such as in a park or garden.

Further innovations

The next appearance of a two-wheeled riding machine was in 1865, when pedals were applied directly to the front wheel. This machine was known as the velocipede (meaning "fast foot") as well as the "bone shaker," since it's wooden structure combined with the cobblestone roads of the day made for an extremely uncomfortable ride. They also became a fad and indoor riding academies, similar to roller rinks, could be found in large cities.

In 1870 the first all-metal machine appeared. (Prior to this, metallurgy was not advanced enough to provide metal which was strong enough to make small, light parts out of.) The pedals were attached directly to the front wheel with no freewheeling mechanism. Solid rubber tires and the long spokes of the large front wheel provided a much smoother ride than its predecessor.

Bicycle Timeline

- 1817: Draisienne
- 1865: Velocipede
- 1870: High-wheel bicycle
- 1878: High-wheel safety
- 1885: Hard-tired safety
- 1888: Pneumatic safety

Dimensiunile ecranelor

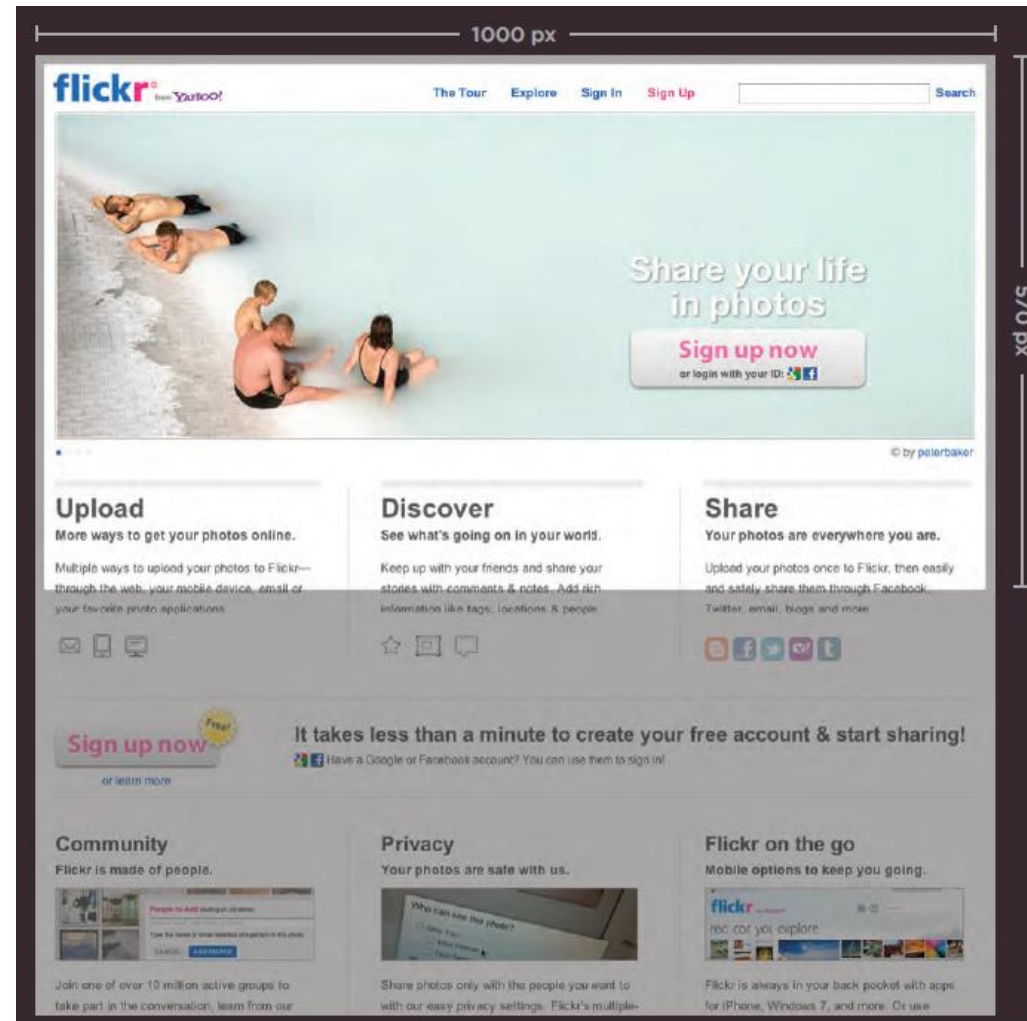
- Diferiti vizitatori ai unui site vor folosi ecrane de dimensiuni diferite pentru a-l vizualiza, ceea ce inseamna ca design-ul site-ului trebuie sa fie flexibil astfel incat sa fie afisat corespunzator pe ecrane de dimensiuni diferite.



- Atunci cand se realizeaza un design pentru printare, se cunoaste intotdeauna dimensiunea paginii pe care designul va fi printat.
- Totusi, cand se realizeaza designul pentru vizionare web, trebuie luat in calcul faptul ca diferiti utilizatori vizualizeaza informatia pe ecrane diferite.
- Dimensiunea ecranelor utilizatorilor determina cat de mari pot fi ferestrele si cat de mult din continutul paginii pot vedea.
- Exista un numar mare de dispozitive (telefoane mobile si tablete) care au ecrane mai mici

Dimensiunile paginilor

- Deoarece dimensiunea și rezoluția ecranelor variază foarte mult designerii paginilor web încearcă de obicei să creeze pagini cu o lățime de 960-1000 pixeli, (majoritatea utilizatorilor vor putea să vizioneze astfel de pagini).
- De asemenea majoritatea designelor încearcă să le prezinte utilizatorilor scopul site-ului în primii 570-600 pixeli (înălțime).
- Suprafața umbră este ascunsă de constrangerile ferestrei browserului, de aceea userul trebuie să utilizeze scroll pentru a vedea regiunea din partea de jos a paginii.
- În consecință în prima parte a paginii se încearcă să se trezească interesul utilizatorilor astfel încât aceștia să utilizeze scroll și să vizualizeze și restul paginii.

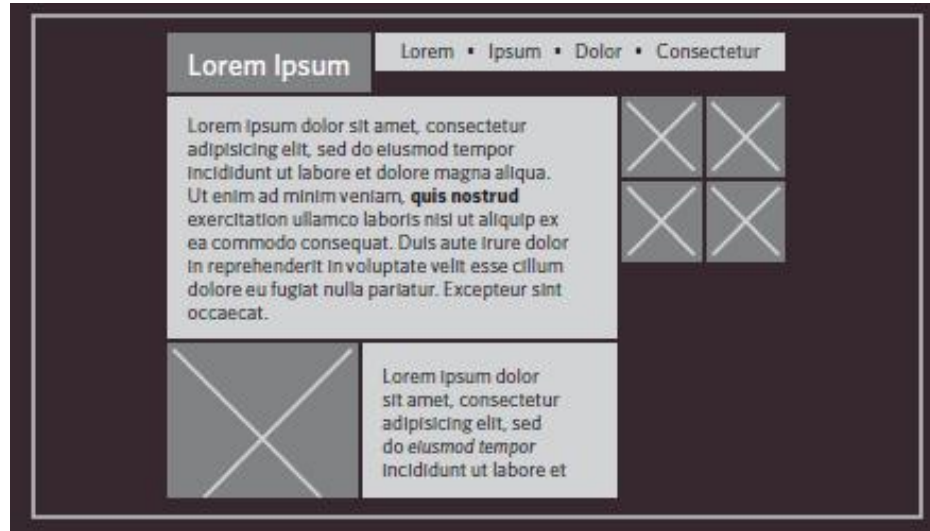


Structura cu *latime fixa*

- Un design care foloseste o **structura cu latime fixa** nu va modifica dimensiunea paginii in functie de dimensiunile ecranelor de pe care este accesata.
- Dimensiunile tind sa fie specificate in **pixeli**.

DEZAVANTAJE

- Pot ramane spatii mari in zonele laterale ale paginii.
- Daca ecranele utilizatorilor au o rezolutie mult mai mare decat cea a paginii, atunci pagina poate arata mult mai mica si este mai dificil sa se citeasca textul.
- Daca utilizatorul mareste dimensiunile fonturilor, s-ar putea ca textul sa nu incapa in spatiul alocat.
- Acest design functioneaza cel mai bine pe dispozitive care au o rezolutie similara cu cea a calculatoarelor de tip desktop sau laptop.
- Pagina va ocupa de obicei mai mult spatiu vertical decat in configuratia cu latime variabila.



AVANTAJE

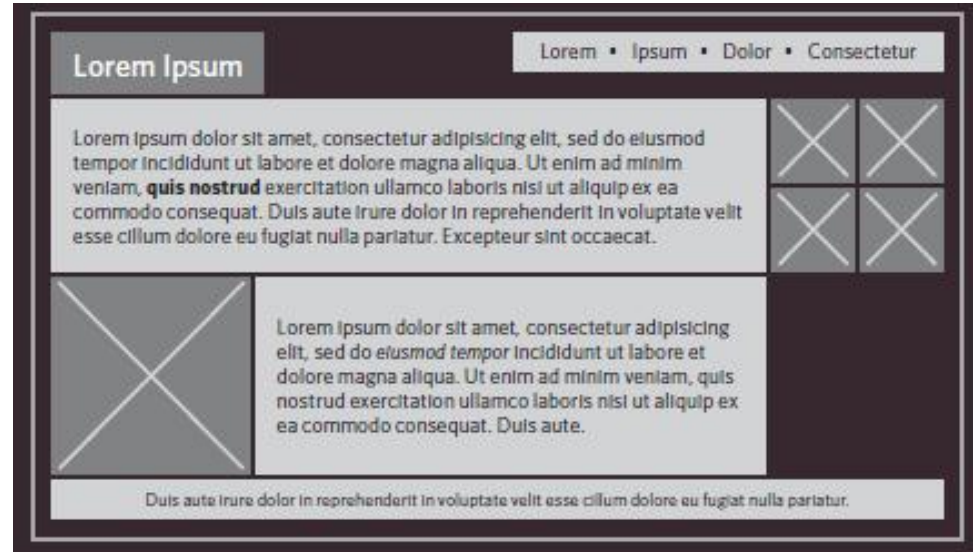
- Valorile in pixeli sunt exacte.
- Designerii au un control mult mai mare asupra modului in care elementele sunt afisate si positionate in pagina fata de structura cu latime variabila.
- Lungimea liniilor de text poate fi controlata indiferent de dimensiunea ferestrei utilizatorilor.
- Dimensiunea unei imagini va ramane la fel relativ la restul paginii.

Structura cu *latime variabila*

- Un design care foloseste o structura cu *latime variabila* se maresc si se micsoreaza in functie de dimensiunile ecranelor de pe care este accesat.
- Dimensiunile tind sa fie date in **procente**.

DEZAVANTAJE

- Daca latimile diferitor sectiuni ale paginii nu sunt controlate, atunci designul poate arata foarte diferit fata de modul intentionat, cu spatii neasteptate in jurul anumitor elemente.
- Daca userul are o fereastră mai largă, liniile de text pot deveni foarte lungi, fiind mai greu de citit.
- Dacă userul are o fereastră foarte îngustă, cuvintele pot fi înghesuite și numărul acestora pe un singur rând este mic.
- În cazul în care un element cu lățime fixă (de ex. o imagine) se află într-un container care este prea mic (deoarece userul a micșorat fereastra) imaginea poate suprapune textul.



AVANTAJE

- Paginile se maresc pentru a ocupa intreaga fereastră a browserului astfel incat sa nu fie spatii in jurul paginii.
- Dacă userul are o fereastră mai mică, pagina se poate micșora pentru a încapă fără ca userul să fie nevoit să facă scroll.
- Designul nu este afectat dacă utilizatorii setează fonturi de dimensiuni mai mari decât cea prevăzută de designer (deoarece pagina se poate întinde)

Exemplu – structura cu latime fixa

Logo

[Home](#) [Products](#) [Services](#) [About](#) [Contact](#)

Feature

Column One

Column Two

Column Three

© Copyright 2011

Exemplu – structura cu latime fixa

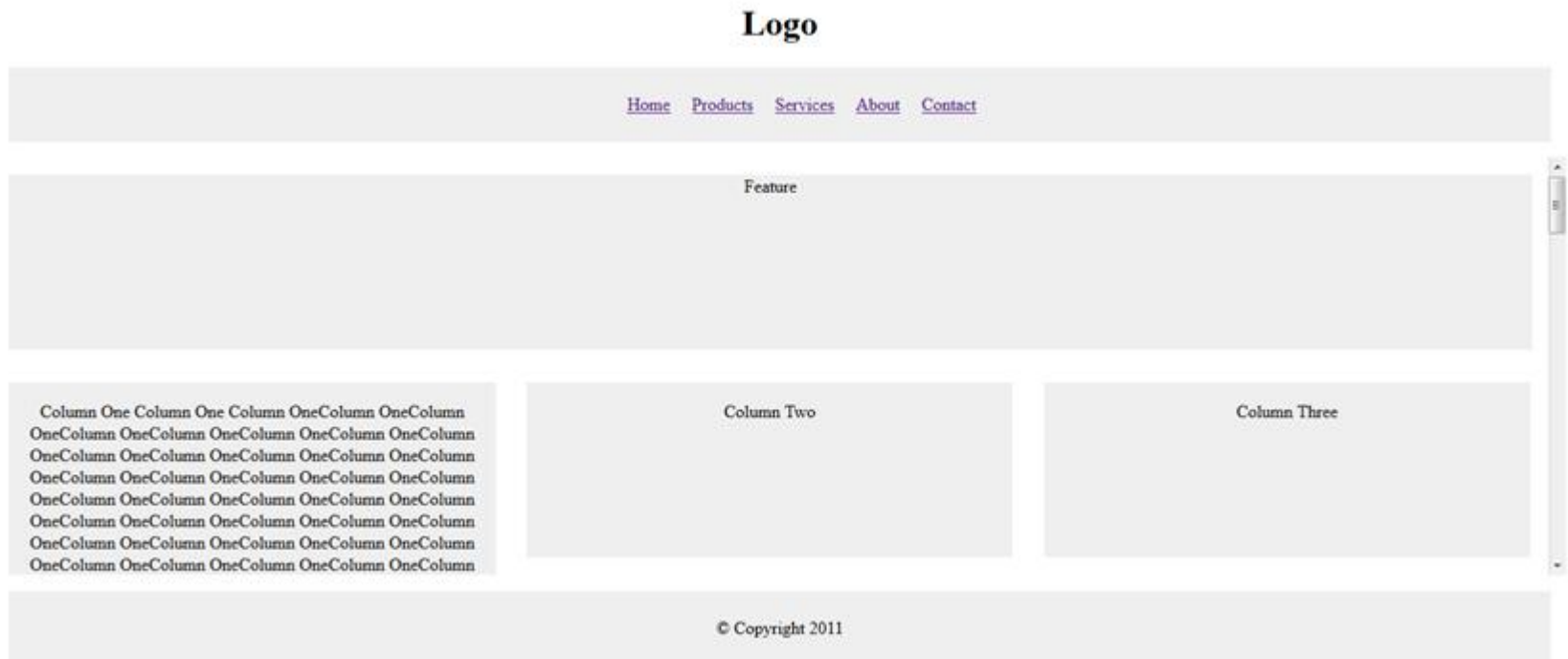
- Regula pentru elementul `<body>` stabileste latimea paginii la 960pixeli. In plus, acesta este centrat prin setarea valorii "auto" pentru marginile left si right.
- Principalele containere din pagina au o margine de 10 pixeli pentru a mentine un spatiu intre ele.
- Containerele *navigation*, *feature* si *footer* se intind astfel incat sa aiba aceeasi latime ca si elementul lor parinte (`<body>` in acest caz). In consecinta nu mai trebuie specificata latimea acestor elemente.
- Cele trei coloane au cate o latime de 300 pixeli si utilizeza proprietatea `float`, care le permite sa stea unele langa altele.

CSS

```
body {  
    width: 960px;  
    margin: 0 auto; }  
#content {  
    overflow: auto;  
    height: 100%; }  
#nav, #feature, #footer {  
    background-color: #efefef;  
    padding: 10px;  
    margin: 10px; }  
.column1, .column2, .column3 {  
    background-color: #efefef;  
    width: 300px;  
    float: left;  
    margin: 10px; }  
li {  
    display: inline;  
    padding: 5px;}  
h1, ul, p{  
    text-align: center; }
```

Exemplu – structura cu latime *variabila*

- Daca va imaginati o fereastră de browser foarte lata sau foarte ingusta, puteti vedea cum liniile de text devin foarte lungi, respectiv foarte scurte.
- Pentru a evita o astfel de situatie se pot utiliza proprietatile [min-width](#) si [max-width](#) pentru a crea limite intre care structura poate varia (aceste proprietati sunt suportate incepand cu IE7)



Exemplu – structura cu latime variabila

- Structura cu latime variabila utilizeaza **procente** pentru a specifica **latimea fiecarui container** astfel incat designul se va intinde/comprima pentru a ocupa latimea ecranului.
- **Nota:** atunci cand testati exemplul in browser micorati si mariti fereastra.
- Regula pentru elementul `<body>` stabileste latimea paginii la **90%** astfel incat sa existe un spatiu mic la stanga si la dreapta continutului principal.
- Cele trei coloane au cate o margine de 1% si o latime de 31.3%. Adunate reprezinta 99.9% din latimea totala a elementului `<body>`.
- Containerele *navigation*, *feature* si *footer* se vor intinde pentru a ocupa toata latimea elementului `<body>`. Aceste elemente au setata o margine de 1% pentru a mentine un spatiu in jurul fiecarei coloane.

CSS

```
body {  
    width: 90%;  
    margin: 0 auto;}  
#content {overflow: auto;}  
#nav, #feature, #footer {margin: 1%;}  
.column1, .column2, .column3 {  
    width: 31.3%;  
    float: left;  
    margin: 1%;}  
.column3 {margin-right: 0%;}  
li {  
    display: inline;  
    padding: 0.5em;}  
#nav, #footer {  
    background-color: #efefef;  
    padding: 0.5em 0;}  
#feature, .article {  
    height: 10em;  
    margin-bottom: 1em;  
    background-color: #efefef;}  
p {background-color: #efefef;}
```

IMAGINI

Cum controlam *dimensiunea* imaginilor folosind CSS

- Dimensiunea unei imagini poate fi controlata in CSS prin intermediul proprietatilor `width` si `height`.
- Avantajul stabilirii dimensiunii unei imagini prin intermediul CSS si nu HTML este acela ca pagina va fi incarcata cu mai mare usurinta deoarece de cele mai multe ori codul CSS si HTML se va incarca inaintea imaginilor, si, comunicandu-i browserului cat spatiu sa lase pentru o imagine, ii permite acestuia sa afiseze restul paginii fara sa astepte ca imaginea sa fie downloadata.

HTML

```



```

CSS

```
img.large {
  width: 500px;
  height: 500px;}
img.medium {
  width: 250px;
  height: 250px;}
img.small {
  width: 100px;
  height: 100px;}
```



Alinierea imaginilor folosind CSS

HTML

`<p> <i>Magnolia</i>`
is a large genus that contains over 200 flowering plant species `</p>`

`<p>`

Some magnolias, such as `<i>Magnolia stellata</i>` and `<i>Magnolia soulangeana</i>`, flower quite ...`</p>`

- Pentru alinierea imaginilor se prefera utilizarea proprietatii `float` in locul atributului `align` al elementului ``.
- Modalitate - se creeaza 2 clase precum `align-left` si `align-right` pentru a alinia imaginile la stanga si la dreapta paginii. Aceste clase se folosesc suplimentar fata de clasa care indica dimensiunea imaginii.
- De obicei se adauga o margine la imagine pentru ca textul sa nu atinga colturile acesteia.

CSS

```
img.align-left {  
    float: left;  
    margin-right: 10px;}
```

```
img.align-right {  
    float: right;  
    margin-left: 10px;}
```

```
img.medium {  
    width: 250px;  
    height: 250px;}
```



Magnolia is a large genus that contains over 200 flowering plant species. It is named after French botanist Pierre Magnol and, having evolved before bees appeared, the flowers were developed to encourage pollination by beetle.

Some magnolias, such as *Magnolia stellata* and *Magnolia soulangeana*, flower quite early in the spring before the leaves open. Others flower in late spring or early summer, such as *Magnolia grandiflora*.



Centrarea imaginilor

HTML

```
<p>
<b><i>Magnolia</i></b> is a large genus that
contains over 200 flowering plant species...</p>
```

CSS

```
img.align-center {
    display: block;
    margin: 0px auto;}

img.medium {
    width: 250px;
    height: 250px;}
```



Magnolia is a large genus that contains over 200 flowering plant species. It is named after French botanist Pierre Magnol and, having evolved before bees appeared, the flowers were developed to encourage pollination by beetle.

- Imaginile sunt elemente inline. Aceasta inseamna ca imaginile sunt integrate in text.
- Pentru a centra o imagine aceasta trebuie transformata intr-un element bloc prin intermediul proprietatii **display**, setata la valoarea **block**.
- Dupa ce imaginea a fost transformata intr-un element bloc, se seteaza proprietatea **margin** a acesteia. Se utilizeaza valoarea **auto** pentru marginile din partea stanga si din partea dreapta.
- Tehnicile utilizate pentru specificarea dimensiunii unei imagini sau pentru centrarea acesteia se aplica si asupra elementului `<figure>` introdus in versiunea HTML 5.

Imagini de background

background-image

- Proprietatea `background-image` permite plasarea unei imagini in spatele oricarui element HTML.
- Implicit, imaginea de background se va repeta a.i. sa se umple intregul container.
- Calea catre imagine urmeaza dupa literele `url` si se afla in interiorul unor paranteze si ghilimele duble.
- Aceasta este imaginea care a fost utilizata in exemplu:



- De obicei imaginile de fundal sunt ultimele care se incarca la afisarea unei pagini. Cu cat imaginea este mai mare cu atat dureaza mai mult incarcarea acesteia.

CSS

body{

```
background-image: url("images/pattern.gif");}
```



CSS

p{

```
background-image: url("images/pattern.gif");}
```

Planting guide

Magnolia

Magnolia is a large genus that contains over 200 flowering plant species. It is named after French botanist Pierre Magnol and, having evolved before bees appeared, the flowers were developed to encourage pollination by beetle.

Repetarea imaginilor

background-repeat

- Proprietatea `background-repeat` poate avea 4 valori:
 - `repeat` – imaginea de background se repeta atat pe orizontala cat si pe verticala
 - `repeat-x` – imagine se repeta doar pe orizontala
 - `repeat-y` – imaginea se repeta doar pe verticala
 - `no-repeat` – imaginea este afisata o singura data
- Proprietatea `background-attachment` specifica daca o imagine de background trebuie sa stea in aceeaasi pozitie sau daca trebuie sa se mute in timp ce userul foloseste scroll. Poate avea 2 valori:
 - `fixed`
 - `scroll`

CSS

```
body{  
  background-image: url("images/header.gif");  
  background-repeat: repeat-x;}
```



Magnolia

Magnolia is a large genus that contains over 200 flowering plant species. It is named after French botanist Pierre Magnol and, having evolved before bees appeared, the flowers were developed to encourage pollination by beetle.

CSS

```
body{  
  background-image: url("images/tulip.gif");  
  background-repeat: no-repeat;  
  background-attachment: fixed;}
```

Planting guide

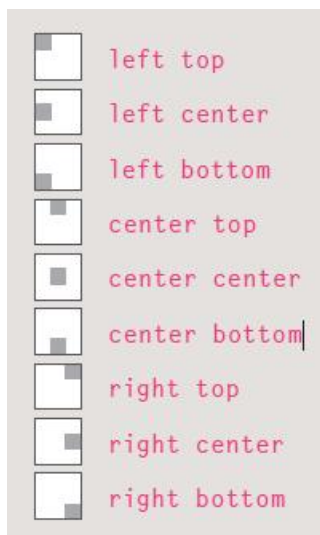
Magnolia

Magnolia is a large genus that contains over 200 flowering plant species. It is named after French botanist Pierre Magnol and, having evolved before bees appeared, the flowers were developed to encourage pollination by beetle.

Pozitionarea background-ului

background-position

- Atunci cand o imagine nu se repeta, se poate folosi proprietatea `background-position` pentru a specifica unde ar trebui plasata imaginea in fereastra browserului.
- Aceasta proprietate are de obicei o pereche de valori. Prima valoare reprezinta pozitia pe orizontala iar cea de-a doua valoare reprezinta pozitia pe verticala.



- In cazul in care se specifica o singura valoare, atunci cea de-a doua valoare va fi implicit `center`.

CSS

```
body{  
  background-image: url("images/tulip.gif");  
  background-repeat: no-repeat;  
  background-position: center top;}
```

Planting guide

Magnolia

Magnolia is a large genus that contains over 200 flowering plant species. It is named after French botanist Pierre Magnol and, having evolved before bees appeared, the flowers were developed to encourage pollination by beetle.

- De asemenea, valorile pot fi specificate si in procente. Acestea reprezinta distanta incepand din coltul STANGA SUS al ferestrei browserului (coltul stanga-sus coincide cu valorile 0% 0%).
- Pentru exemplul de mai sus, pentru a obtine acelasi rezultat se pot adauga si urmatorii valori:
`background-position: 50% 0%;`

END