

TEHNICI WEB

CSS

Claudia Chiriță . 2024/2025

METADATA

HTML METADATA

```
<head>
  <title>Titlul paginii</title>
  <meta charset="utf-8">
  <meta name="autor" content="nume autor">
  <meta name="description" content="descriere">
  <meta name="keywords" content="cuvinte cheie">
  <base href="base URL for relative paths">

  <style type="text/css"></style>
  <script type="text/javascript"></script>
  <script type="text/javascript" src="script.js"></script>
  <link rel="stylesheet" href="style.css" type="text/css">
  <link rel="icon" href="favicon.ico" type="image/x-icon">
</head>
```

ELEMENTUL BASE

- definește URL-ul de bază pentru toate URL-urile relative din document
- într-un document poate exista un singur element *base*

ELEMENTUL BASE

```
<!DOCTYPE html>
<html lang="ro">
  <head>
    <title>Titlul paginii</title>
    <base
href="https://cs.unibuc.ro/~cechirita/tw/c3/
" target="_blank">
    <link rel="stylesheet" href="demo.css">
  </head>
  <body>
    <p>
      <a
href="secret/secret.html">secret</a>
    </p>
    
  </body>
</html>
```



CSS

CSS = CASCADING STYLE SHEETS

- limbaj style sheet folosit pentru definirea aspectului unei pagini web:
 - stilizarea elementelor: culoare, mărime, font etc.
 - așezarea elementelor în pagină
 - determinarea vizibilității
- folosit și pentru animații

FIȘIERUL DE STIL

```
<head>
...
<link rel="stylesheet" href="filename.css" type="text/css">
...
</head>
```


REGULI CSS

- fișierul de stil conține una sau mai multe reguli CSS

```
selector {  
  prop1: valoare1;  
  prop2: valoare2;  
  prop3: valoare3;  
  ...  
}
```

```
p {  
  color: pink;  
  font-family: sans-serif;  
  font-size: 12pt;  
}
```

HTML + CSS ALIGNMENT

în fișier separat

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <link rel="stylesheet" href="demo.css"
type="text/css">
  </head>
  <body>
    <p>
      <a
href="https://en.wikipedia.org/wiki/Alignmen
t_(Dungeons_%26_Dragons)">lawful good</a>
    </p>
  </body>
</html>
```

>

HTML + CSS ALIGNMENT

în fișier

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <style type="text/css">
      a {color: pink;}
    </style>
  </head>
  <body>
    <p>
      <a
href="https://en.wikipedia.org/wiki/Alignmen
t_(Dungeons_%26_Dragons)">chaotic
neutral</a>
    </p>
  </body>
</html>
```

>

HTML + CSS ALIGNMENT

in line

```
<!DOCTYPE html>
<html lang="en">
  <head>
  </head>
  <body>
    <p>
      <a style="color: red"
href="https://en.wikipedia.org/wiki/Alignmen
t_(Dungeons_%26_Dragons)">chaotic evil</a>
    </p>
  </body>
</html>
```

>

PRECEDENȚA STILURILOR

în ordine crescătoare a precedenței:

1. stilul implicit al browser-ului
2. fișier CSS extern și stiluri *in file*; se aplică stilul care apare scris ultimul în <head>
3. stiluri inline

SELECTORI

```
selector1, selector2 {  
  prop1: valoare1;  
}  
selector1 {  
  prop2: valoare2;  
}
```

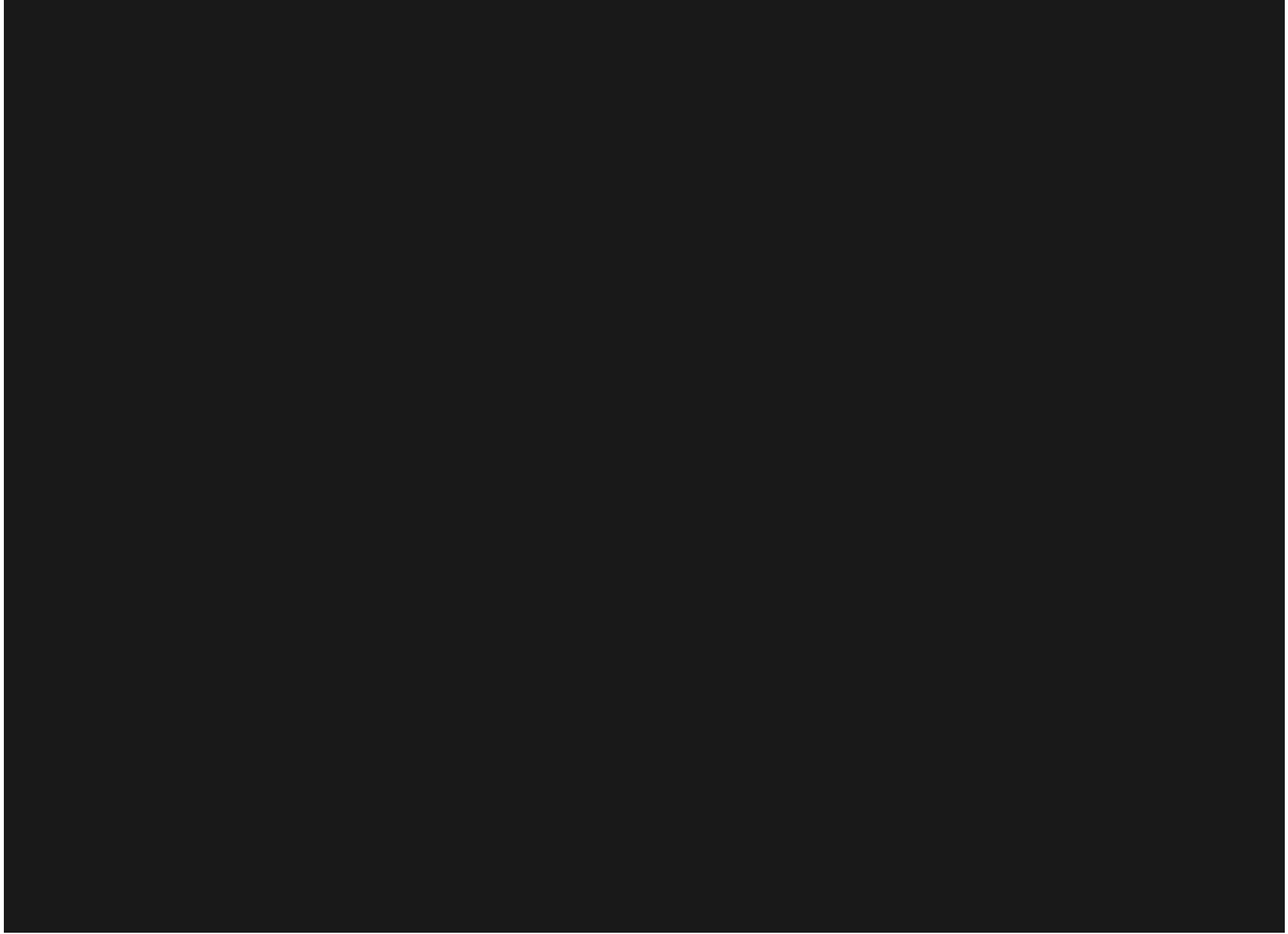
- unui element i se pot aplica mai multe reguli
- se grupează elementele care au un stil comun; stilurile specifice se scriu în reguli separate
- unele stiluri sunt moștenite de la elementele părinte (în arborele DOM), dar pot fi schimbate folosind reguli specifice

PROPRIETĂȚI: CULORI

`color:` definirea culorii textului;

`background-color:` definirea culorii fundalului;

- specificate prin `nume`: red, green, blue, pink, ...



PROPRIETĂȚI: CULORI

- specificate prin valoarea `rgb(red, green, blue)`
red, green, blue au valori între 0-255
- specificate prin valoarea hexazecimală `#rrggbb`
rr, gg, bb au valori între 0-255

PROPRIETĂȚI: CULORI

- prin valoarea `hsl(hue, saturation, lightness)`
 - hue reprezintă gradul culorii (0-360)
 - saturation reprezintă intensitatea (0%-100%)
 - lightness reprezintă luminozitatea (0%-100%)

PROPRIETĂȚI: CULORI

- prin `valoarea rgba(red, green, blue, alpha)`
extensie a valorii `rgb` cu valoarea `alpha` care reprezintă opacitatea (0.0-1.0)
- prin `hsla(hue, saturation, lightness, alpha)`
extensie a valorii `hsl` cu `alpha`
- `convertor culori`

DISTANȚA/MĂRIMEA

- unități absolute: mm, cm, px, in, pt
- unități relative:
 - em: relativ la dimensiunea fontului elementului
 - %: relativ la dimensiunea părintelui
 - vw, vh: relativ la 1% din lățimea/înălțimea viewportului
 - rem: relativ la dimensiunea fontului rădăcinii

SELECTORI

SELECTORUL UNIVERSAL *

```
<!DOCTYPE html>
<html lang="ro">
  <head>
    <link rel="stylesheet" href="demo.css">
    <style type="text/css">
      * {border: 1px solid pink;}
    </style>
  </head>
  <body>
    <h1>Titlu</h1>
    <div><p>paragraf în div</p></div>
    <p>paragraf fără div</p>
    <p>alt paragraf fără div</p>
  </body>
</html>
```



SELECTAREA PRIN TIP

numele unui element din document: h1, p, div, a...

```
<!DOCTYPE html>
<html lang="ro">
  <head>
    <link rel="stylesheet" href="demo.css">
    <style type="text/css">
      p { color: pink;
        margin-left: 25px;}
    </style>
  </head>
  <body>
    <h1>Titlu</h1>
    <div><p>paragraf în div</p></div>
    <p>paragraf fără div</p>
    <p>alt paragraf fără div</p>
  </body>
</html>
```

>

SELECTAREA PRIN ID

- #identif selectează elementul cu id="identif"
- <tag id="identif"> conținut </tag>
un document HTML poate conține un singur atribut id cu valoarea "identif"

```
<!DOCTYPE html>
<html lang="ro">
  <head>
    <link rel="stylesheet" href="demo.css">
    <style type="text/css">
      div#galben { background-color:
yellow;}
    </style>
  </head>
  <body>
    <h1>Titlu</h1>
    <div id="galben"><p>paragraf în div</p>
</div>
    <p>paragraf fără div</p>
    <p>alt paragraf fără div</p>
  </body>
</html>
```

>

SELECTAREA PRIN CLASĂ

- .cls selectează elementele cu class="cls"
- <tag1 class="cls"> conținut </tag1>
<tag2 class="cls"> conținut </tag2>
cu atributul class grupăm mai multe elemente, care vor avea un stil comun

```
<!DOCTYPE html>
<html lang="ro">
  <head>
    <link rel="stylesheet" href="demo.css">
    <style type="text/css">
      .galben { background-color: yellow;}
    </style>
  </head>
  <body>
    <h1>Titlu</h1>
    <div class="galben"><p>paragraf în
div</p></div>
    <p>paragraf fără div</p>
    <p class="galben">alt paragraf fără
div</p>
  </body>
</html>
```

>

SELECTAREA PRIN CLASĂ

```
<!DOCTYPE html>
<html lang="ro">
  <head>
    <link rel="stylesheet" href="demo.css">
    <style type="text/css">
      p.a.b { background-color: yellow;}
      p.a, p.b { border: 1px solid pink; }
      p.a:not(.b){ color: red; }
      p.b:not(.a){ color: cyan; }
    </style>
  </head>
  <body>
    <h1>Titlu</h1>
    <div class="a">div</div>
    <p class="a b">paragraf 1</p>
    <p class="a">paragraf 2</p>
    <p class="b">paragraf 3</p>
    <p> paragraf 4</p>
  </body>
```



SPECIFICITATE

selectarea prin id are specificitate mai mare decât
selectarea prin class

```
<!DOCTYPE html>
<html lang="ro">
  <head>
    <link rel="stylesheet" href="demo.css">
    <style type="text/css">
      #rosu { color: red; }
      .bleu { color: cyan; }
    </style>
  </head>
  <body>
    <h1>Titlu</h1>
    <p id="rosu" class="bleu">paragraf</p>
  </body>
</html>
```

>

SPECIFICITATE

când două stiluri au aceeași specificitate, câștigă cel care este scris ultimul în fișierul CSS

```
<!DOCTYPE html>
<html lang="ro">
  <head>
    <link rel="stylesheet" href="demo.css">
    <style type="text/css">
      .rosu { color: red; }
      .bleu { color: cyan; }
    </style>
  </head>
  <body>
    <h1>Titlu</h1>
    <p class="rosu bleu">paragraf</p>
  </body>
</html>
```

>

!IMPORTANT

proprietățile marcate !important sunt aplicate
indiferent de specificitatea lor

```
<!DOCTYPE html>
<html lang="ro">
  <head>
    <link rel="stylesheet" href="demo.css">
    <style type="text/css">
      .rosu { color: red; }
      .bleu { color: cyan; }
      p { color: pink !important; }
    </style>
  </head>
  <body>
    <h1>Titlu</h1>
    <p class="rosu bleu">paragraf</p>
  </body>
</html>
```

>

INHERIT

- unele proprietăți pot lua valoarea inherit:
proprietatea este moștenită de la elementul părinte
- folosită pentru a suprascrie un alt stil și a forța comportamentul implicit

```
<!DOCTYPE html>
<html lang="ro">
  <head>
    <link rel="stylesheet"
href="demo.css">
    <style type="text/css">
      div { color: red; }
      p { color: cyan; }
      div>p { color: inherit; }
    </style>
  </head>
  <body>
    <h1>Titlu</h1>
    <p>paragraf în afara divului</p>
    <div><p>paragraf în div</p></div>
  </body>
</html>
```

>

COMBINAREA SELECTORILOR

- selectori pentru descendenți

$E \ F$ selectează toate elementele F care sunt descendenți ai lui E

- selectori pentru copii

$E > F$ selectează elementele F care sunt copii ai lui E

- selectori pentru frați

$E + F$ selectează toate elementele F care sunt frați cu E și care urmează imediat după E

$E \sim F$ selectează toate elementele F care sunt frați cu E și care urmează după E

COMBINAREA SELECTORILOR

```
<!DOCTYPE html>
<html lang="ro">
  <head>
    <link rel="stylesheet" href="demo.css">
    <style type="text/css">
      div > p { color:black; background-
color: yellow; }
    </style>
  </head>
  <body>
    <h1>Titlu</h1>
    <div><p>paragraf 1 în div</p>
      <p>paragraf 2 în div</p>
      <aside><p>paragraf 3 în div</p>
</aside>
    </div>
    <p>paragraf 4 în afara divului</p>
    <p>paragraf 5 în afara divului</p>
  </body>
```

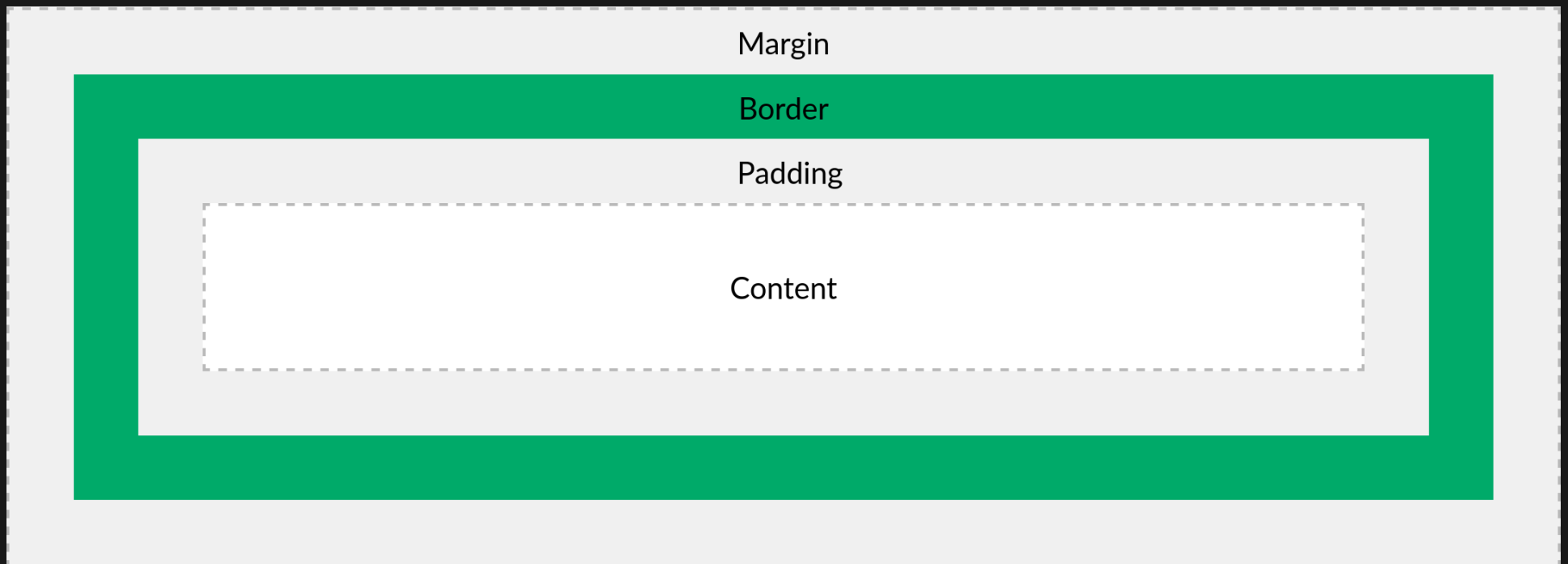


SELECTAREA PRIN ATRIBUTE

- `e[attr]` -> elementele e care au atributul attr
- `e[attr="val"]` -> elementele e cu attr="val"
- `e[attr~="val"]` -> elementele e cu proprietatea că valoarea lui attr conține cuvântul val
- `e[attr*="val"]` -> elementele e cu proprietatea că valoarea lui attr conține subșirul val

CSS BOX MODEL

- reprezentare a spațiului pe care îl ocupă un element pe o pagină și a modului în care acesta se raportează la alte elemente



CSS BOX MODEL

- width / height
- padding
- border
- margin

valori CSS default

WIDTH + HEIGHT

```
width: 500px /* 50% */  
/* height, max-width, min-width, max-height, min-height */
```

BORDER

```
border: grosime stil culoare;  
border-left: grosime stil culoare;  
border-right: grosime stil culoare;  
border-top: grosime stil culoare;  
border-bottom: grosime stil culoare;
```

- grosime: exprimată în px, pt, em, *thin, medium, thick*
- stil: *none, solid, dotted, dashed, inset, outset, double, groove, ridge, hidden*

BORDER

```
border-color: ...;  
border-style: ...;  
border-width: ...;  
border-left-color: ...;  
border-bottom-width: ...;
```

```
border-radius: 12px; /* margini cu colțuri rotunjite */
```

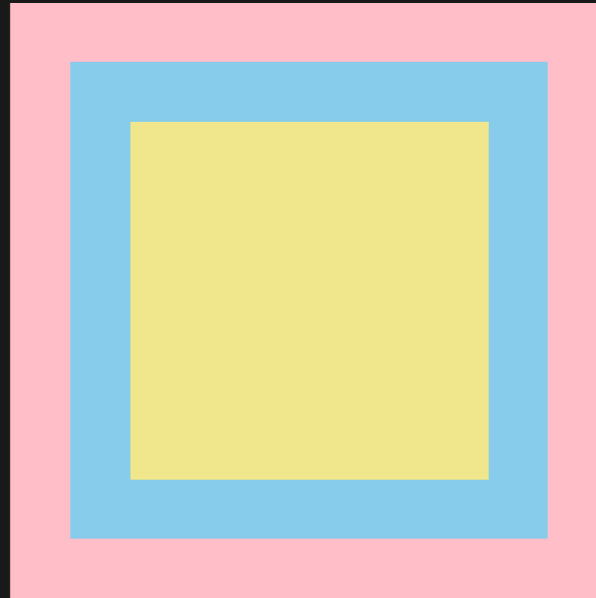
PADDING + MARGIN

```
padding: 15px;  
padding-top: 15px;  
padding-right: 20%;  
padding-bottom: 15px;  
padding-left: 10%;
```

```
margin: 15px;  
margin-top: 15px;  
margin-right: 20%;  
margin-bottom: 15px;  
margin-left: 10%;
```

```
margin: 15px 20% 15px 10%;  
padding: 15px 20% 15px 10%;
```

FLASH QUIZ



- pătratul galben are lățimea și înălțimea de 150px
- pătratul mare are lățimea și înălțimea de 250px
- banda roz și bleu au grosimea de 25px

FLASH QUIZ

```
<div id="big-square">  
  <div id="small-square"></div>  
</div>
```

- pătratul galben are lățimea și înălțimea de 150px
- pătratul mare are lățimea și înălțimea de 250px
- banda roz și bleu au grosimea de 25px

FLASH QUIZ

```
<!DOCTYPE html>
<html lang="ro">
  <head>
    <link rel="stylesheet" href="demo.css">
    <style type="text/css">
      #big-square {}

      #small-square {}
    </style>
  </head>
  <body>
    <div id="big-square">
      <div id="small-square"></div>
    </div>
  </body>
</html>
```



FLASH QUIZ SOLUTION #1

```
<!DOCTYPE html>
<html lang="ro">
  <head>
    <link rel="stylesheet" href="demo.css">
    <style type="text/css">
      #big-square { background-color:
pink;
      height: 200px;
      padding: 25px;
      width: 200px; }

      #small-square { background-color:
khaki;
      border: 25px solid skyblue;
      height: 150px;
      width: 150px;
      }
    </style>
  </head>
  <body>
    <div id="big-square">
      <div id="small-square"></div>
    </div>
  </body>
```



FLASH QUIZ SOLUTION #2

```
<!DOCTYPE html>
<html lang="ro">
  <head>
    <link rel="stylesheet" href="demo.css">
    <style type="text/css">
      #big-square { background-color:
skyblue;
      border: 25px solid pink;
      height: 200px;
      width: 200px; }

      #small-square { background-color:
khaki;
      margin: 25px;
      height: 150px;
      width: 150px;
      }
    </style>
  </head>
  <body>
    <div id="big-square">
      <div id="small-square"></div>
    </div>
  </body>
```



FONTS

```
font-family: Helvetica,"Times New Roman",serif;  
font-size: 16px; /* 16pt, 1.25em, 125%, x-large*/  
font-weight: normal; /* bold, lighter, bolder, 100,...,900 */  
font-style: normal; /* italic, oblique */
```

font-family:

- fonturi generice: serif, sans-serif, monospace, cursive, fantasy
- fonturi specifice: Times New Roman, Arial, Georgia, Verdana

FONTS

```
font-family: Helvetica, "Times New Roman", serif;  
font-size: 16px; /* 16pt, 1.25em, 125%, x-large*/  
font-weight: normal; /* bold, lighter, bolder, 100, ..., 900 */  
font-style: normal; /* italic, oblique */
```

font-size:

- unități: px, pt, em (1em = dimensiunea fontului curent)
- valori: xx-small, x-small, small, medium, large, x-large, xx-large, ...
- procente: procent din dimensiunea fontului părintelui

FONTURI CUSTOM

```
/* pentru importarea unui font*/  
  
@font-face {  
  font-family: nume-font;  
  src:url(file-font.ext);  
}  
  
selector { font-family: nume-font;  
}
```

FONTURI CUSTOM

exemplu: embed mr. Roboto

FORMATARE TEXT

```
color: pink;
text-align: center; /* left, right, justify */
text-decoration: underline; /* overline, line-through */
text-shadow: -2px 5px 2px gray; /* none initial */
/* x-offset, y-offset, blur-radius, culoare */
text-indent: 3em;
text-transform: uppercase; /* lowercase, capitalize */
line-height: 150%; /* 1.5em , 30px */
word-spacing: 1em; /* +15mm */
letter-spacing: 0.1em; /* -1px */
```

VIZIBILITATE

```
visibility: visible; /* hidden */
```

- setează vizibilitatea unui element;
chiar dacă vizibilitatea este *hidden*, elementul ocupă
loc în pagină

OVERFLOW

```
overflow: hidden; /* visible, scroll */
```

- descrie modul în care este redat textul care iese dintr-un cadru

DISPLAY

```
display: inline; /* block, inline-block, none */
```

- descrie modul în care este afișat un element în pagină
- spațiul pe care îl ocupă este lungimea elementului
- ignoră proprietățile width și height, dar acceptă border, padding

DISPLAY

```
display: block; /* inline, inline-block, none */
```

- descrie modul în care este afișat un element în pagină
- permite stilizarea folosind box model
- plasare pe rând nou

DISPLAY

```
display: inline-block; /* inline, block, none */
```

- descrie modul în care este afișat un element în pagină
- element poziționat inline care poate fi stilizat ca un element bloc
- acceptă proprietățile width și height

DISPLAY

```
display: none; /* inline, block, inline-block */
```

- descrie modul în care este afișat un element în pagină
- elementul nu este afișat
- deși elementul este eliminat din pagină, el este prezent în DOM

EXTRA++

- cascading style sheets
- utf-8 vs utf-16 vs ascii

(" , . : ;) , - _ / ~ \ |

e

întrebări?

