

## 5. Introducere in HTML si CSS

Adrian Adiaconitei

***LINK******Academy***

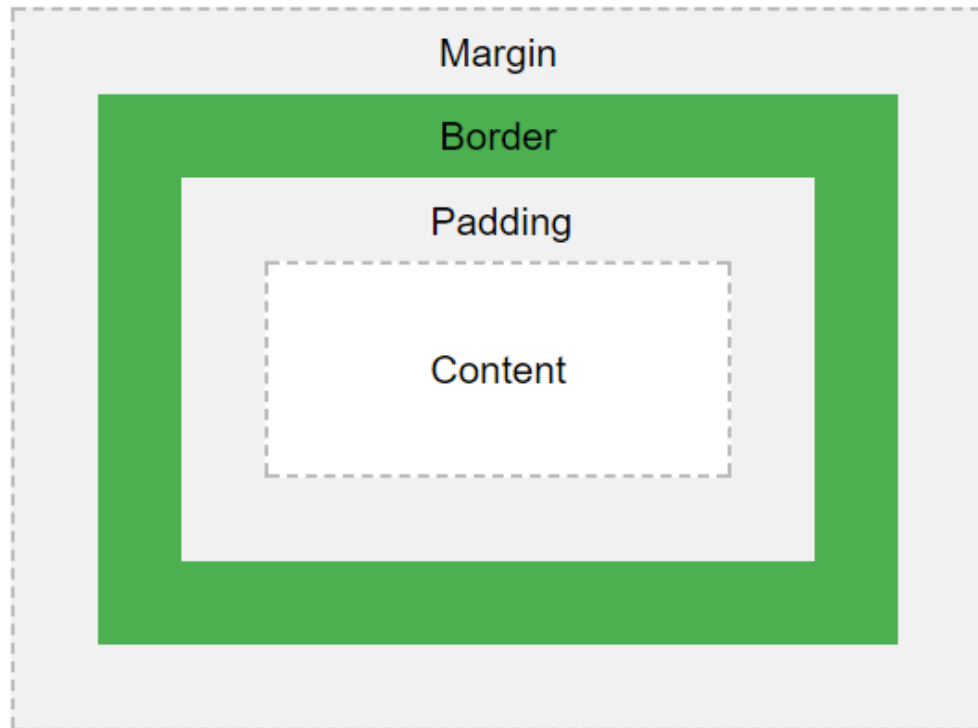
# Objective

- ✓ CSS Box
- ✓ CSS Layout
- ✓ Grid
- ✓ Responsive web design
- ✓ Media Queries
- ✓ Fonturi
- ✓ Recapitulare

# CSS box

Toate elementele HTML pot fi considerate ca fiind “cutii” sau containere pentru continutul lor

- ✓ **Content** - unde apar text, imagini si alte elemente HTML
- ✓ **Padding** - zona libera (transparenta) din jurul continutului
- ✓ **Border** – chenarul (frontier) din jurul padding-ului si continutului
- ✓ **Margin** – zona libera (transparenta) din jurul “cutiei”



# CSS box

CSS are urmatoarele proprietati de contur (in afara de border):

***outline-style:***

***outline-color:***

***outline-width:***

***outline-offset:***

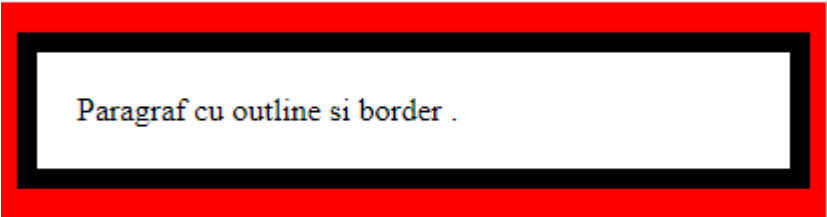
***outline:*** *outline-width outline-style outline-color*

*box-shadow* -

[https://www.w3schools.com/css/css3\\_shadows\\_box.asp](https://www.w3schools.com/css/css3_shadows_box.asp)

```
p.ex1 {  
    border: 10px solid black;  
    outline-style: solid;  
    outline-color: red;  
    outline-width: 15px;  
    padding :20px;  
}
```

## Prorrietatea outline



Paragraf cu outline si border .

# CSS layout

*display (position, float) este cea mai importanta proprietate a CSS pentru controlul aspectului.*

*Elemente HTML au o valoare de afisare implicita*

*Valoarea de afisare implicita pentru majoritatea elementelor este :*

- ✓ **block**
- ✓ **inline.**

# CSS layout

## HTML5

de tip **block** :

<header> - antetul paginii

<nav> - navigare

<article> - articol din pagina

<section> - sectiune din pagina

<aside> - coloanele laterale

<footer> - subsolul paginii

de tip **inline**:

- <strong> - cuvinte **importante**
- <em> - cuvinte *accentuate*
- <img>

# CSS layout

*display: inline; (implicit)*

*display: none;*

*display: block;*

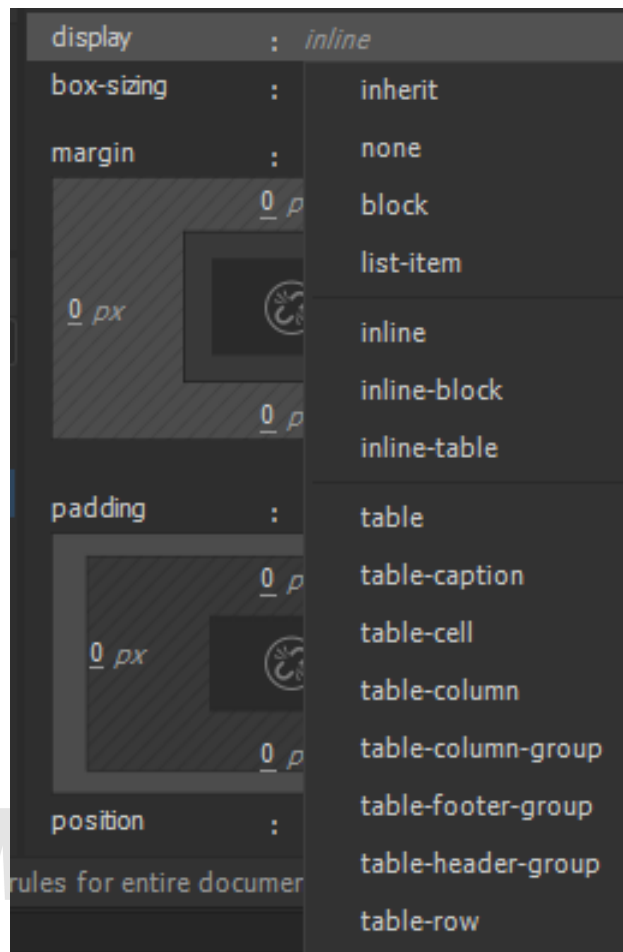
*display: inline-block;*

*display: flex;*

*position: static | absolute | fixed | relative  
| sticky | initial | inherit;*

[https://www.w3schools.com/cssref/pr\\_class\\_display.asp](https://www.w3schools.com/cssref/pr_class_display.asp)

[https://www.w3schools.com/cssref/playit.asp?filename=playcss\\_display&preval=flex](https://www.w3schools.com/cssref/playit.asp?filename=playcss_display&preval=flex)



# display - Grid

CSS Grid proprietati:

[https://www.w3schools.com/css/css\\_grid.asp](https://www.w3schools.com/css/css_grid.asp)

display: **grid**;

**grid**: none | *grid-template-rows / grid-template-columns | grid-template-areas | grid-template-rows / [grid-auto-flow] grid-auto-columns | [grid-auto-flow] grid-auto-rows / grid-template-columns* | initial | inherit;

**grid-area**: *grid-row-start / grid-columns-start / grid-row-end / grid-column-end | areaname*;

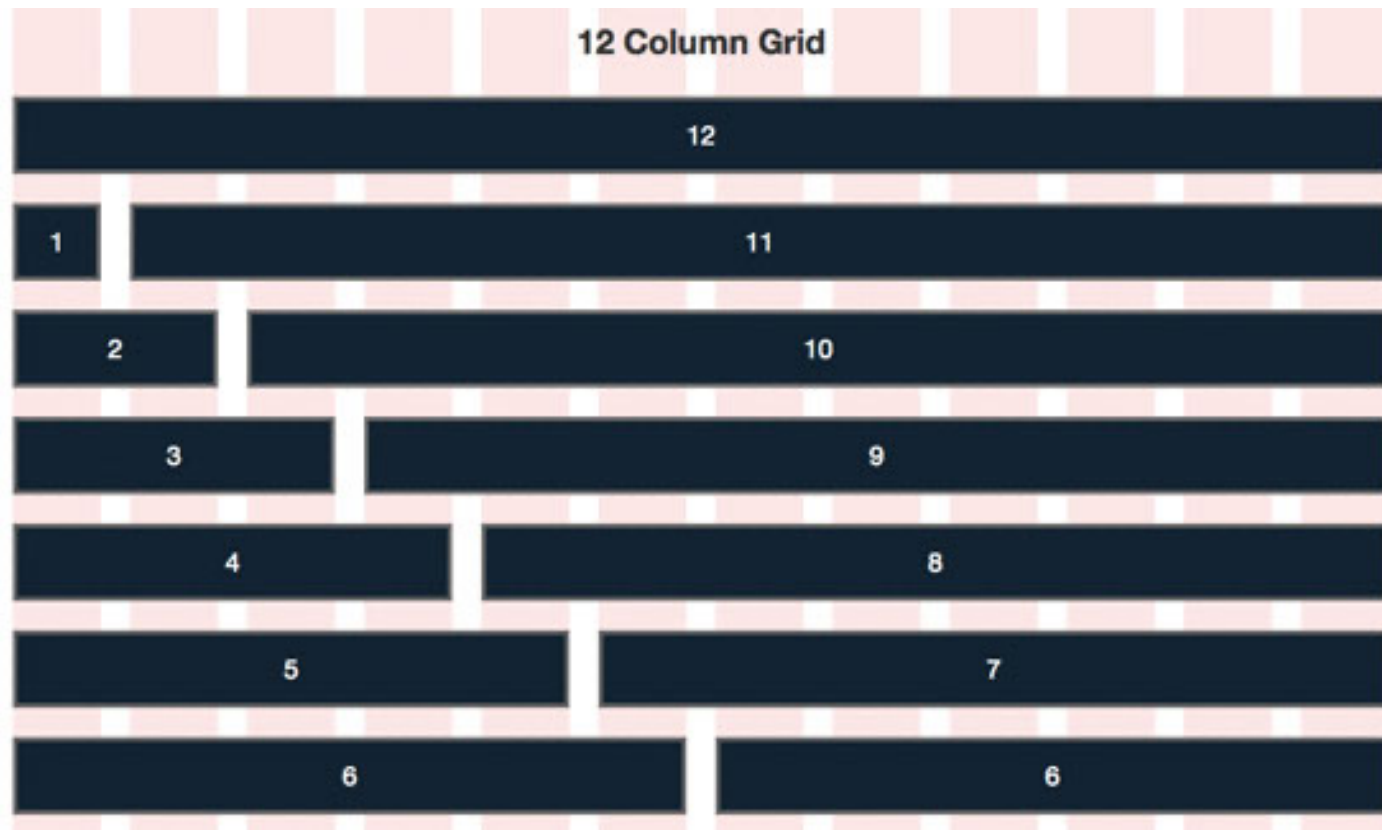
[https://www.w3schools.com/cssref/pr\\_grid.asp](https://www.w3schools.com/cssref/pr_grid.asp)



# Responsive design - Grid

## CSS Grid Layout

ofera un system bazat pe randuri si coloane, fara a fi nevoie sa folosim float si position.



# display - Grid

Aplicația1: Pornind de la acest exemplu

[https://www.w3schools.com/cssref/tryit.asp?filename=trycss\\_grid-area](https://www.w3schools.com/cssref/tryit.asp?filename=trycss_grid-area)

generați următorul grid

1		2	3
		4	5
6	7	8	9

# Responsive design

**Site-uri pentru toate dispozitive,** dar site-uri care se vor **adapta automat** si vor reactiona (engl.*respond* – *responsive*) in mediul in care sunt prezentate.

**Diferenta dintre structurile fixe, fluide, adaptive si *responsive***



# Responsive design

## ✓ Viewport

`<meta name="viewport" content="width=device-width, initial-scale=1.0">`

**width=device-width** stabileste latimea paginii (care va varia in functie de dispozitivul respectiv).

**initial-scale=1.0** stabileste nivelul initial de zoom atunci cand pagina este mai intai incarcat de catre browser.

1. Nu folositi elemente de mari dimensiuni cu latime fixa (**width:100%; max-width:dimensiunea dorita px;**)
2. Folositi **viewport**
3. Folositi **Media query** pentru a aplica diferite stilizare pentru ecrane mici si mari

# Responsive design

✓ **Tipurile media** nu sunt o noutate in CSS3

Au fost folosite in CSS2 pentru o eventuala separare de stiluri, care trebuia sa se aplice tiparului. De exemplu:

```
<link href="base.css" rel="stylesheet" type="text/css" media="screen" />
```

```
<link href="print.css" rel="stylesheet" type="text/css" media="print"/>
```

Au mai fost disponibile unele valori, dar practic nu s-au folosit des. Pentru lista completă accesati linkul urmator (specificație mai veche):

<http://www.w3.org/TR/CSS21/media.html#media-types>

✓ **Media query** ne permite nu doar targetarea unui anumit tip de dispozitiv sau scop (cum a fost mai devreme), ci si sa verificam caracteristicile prezentarii curente in browser si caracteristicile dispozitivelor in sine.

```
<link href="test.css" rel="stylesheet" type="text/css" media="screen and  
(max-width: 480px) and (orientation: portrait)" />
```

# Responsive design - Media queries

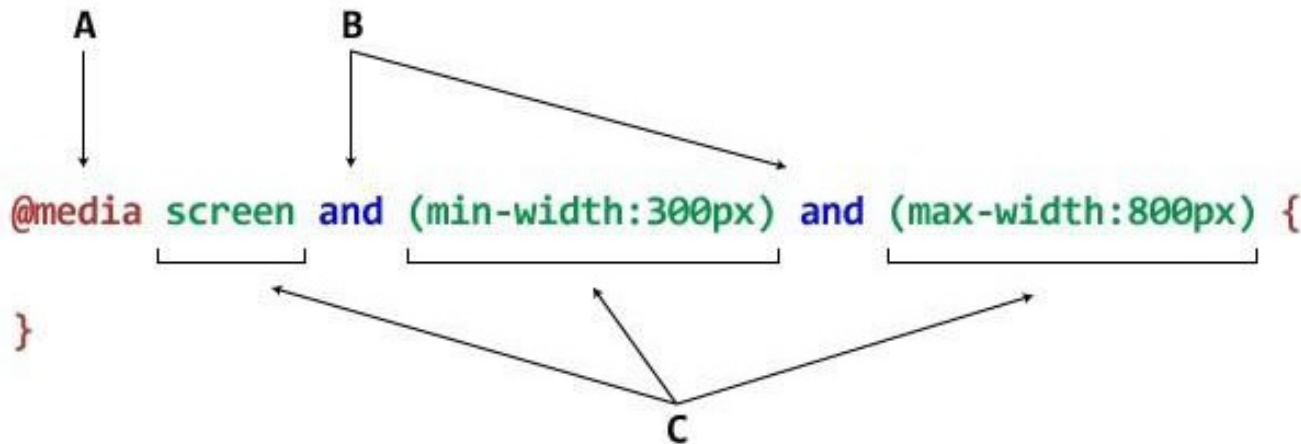
**Media queries** este o tehnica CSS introdus in CSS3.

Se foloseste de **@media** regula pentru a include un bloc de proprietati CSS numai in cazul in care o anumita conditie este adevarata.

<https://css-tricks.com/snippets/css/media-queries-for-standard-devices/>

# Responsive design - Media queries

Din ce este alcatuita interogarea media in cadrul unui fisier CSS



A - semnul pentru interogarea media. Întotdeauna începem cu `@media`

B - operatori logici

C -caracteristici media (engl. media features)

# Responsive design - Media queries

Majoritatea caracteristicilor media accepta prefixele ***min-*** si ***max-*** pentru a putea sa exprimam conditia sub forma de „*egal sau mai mare decat*“, respectiv „*egal sau mai mic decat*“.

```
@media screen and (width: 800px) { ... }
```

```
@media screen and (max-width: 800px) { ... }
```

```
@media screen and (min-width: 400px) and (max-width: 800px) { ... }
```

LINK Academy

<https://www.w3.org/TR/css3-mediaqueries/#media1>



# Responsive design - Media queries

## Aplicația2:

Faceti o pagina HTML care va afișa o imagine diferită pentru desktop, laptop, tableta, mobil

# CSS fonturi

Proprietatile de fonturi CSS definesc familia de fonturi, evidentierea, marimea si stilul unui text.

- **font**: Seteaza toate proprietatile fontului intr-o singura declaratie
- **font-family**: familia de fonturi
- **font-size**: dimensiunea fontului
- **font-style**: stilul fontului pentru text
- **font-variant**: daca un text ar trebui sau nu sa fie afisat intr-un font cu litere mici
- **font-weight**: evidentierea(bold) unui font



Sans-serif



Serif



Serif  
(red serifs)

# CSS fonturi/iconite

**Aplicația3:** Sa se adauge 10 iconite in pagina

<http://fontawesome.io/icons/>      <https://fonts.google.com/>

```
<!DOCTYPE html>
<html><head>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
</head>
<body>
<i class="fa fa-cloud"></i>
<i class="fa fa-heart"></i>
<i class="fa fa-car"></i>
<i class="fa fa-file"></i>
<i class="fa fa-bars"></i>
</body></html>
```

# CSS fonturi externe

*Making the Web Beautiful!*

```
<link  
href="https://fonts.googleapis.com/css?family=Supermercado+On  
e" rel="stylesheet">  
  
<style>  
@import  
url('https://fonts.googleapis.com/css?family=Supermercado+One')  
;  
</style>
```

[https://developers.google.com/fonts/docs/getting\\_started](https://developers.google.com/fonts/docs/getting_started)

<https://sia.codes/posts/making-google-fonts-faster/>

# CSS fonturi externe

*Making the Web Beautiful!*

```
@font-face {  
  font-family: myFirstFont;  
  src: url(sansation_light.woff);  
}
```

```
@font-face {  
  font-family: myFirstFont;  
  src: url(sansation_bold.woff);  
  font-weight: bold;  
}
```

```
div {  
  font-family: myFirstFont;  
}
```

**LINK**Academy

# CSS fonturi

*Making the Web Beautiful!*

## Aplicația4:

Adăugați, în proiect, fonturi speciale, de la google sau importate local.

Completați CSS –ul din style.css de la proiect.