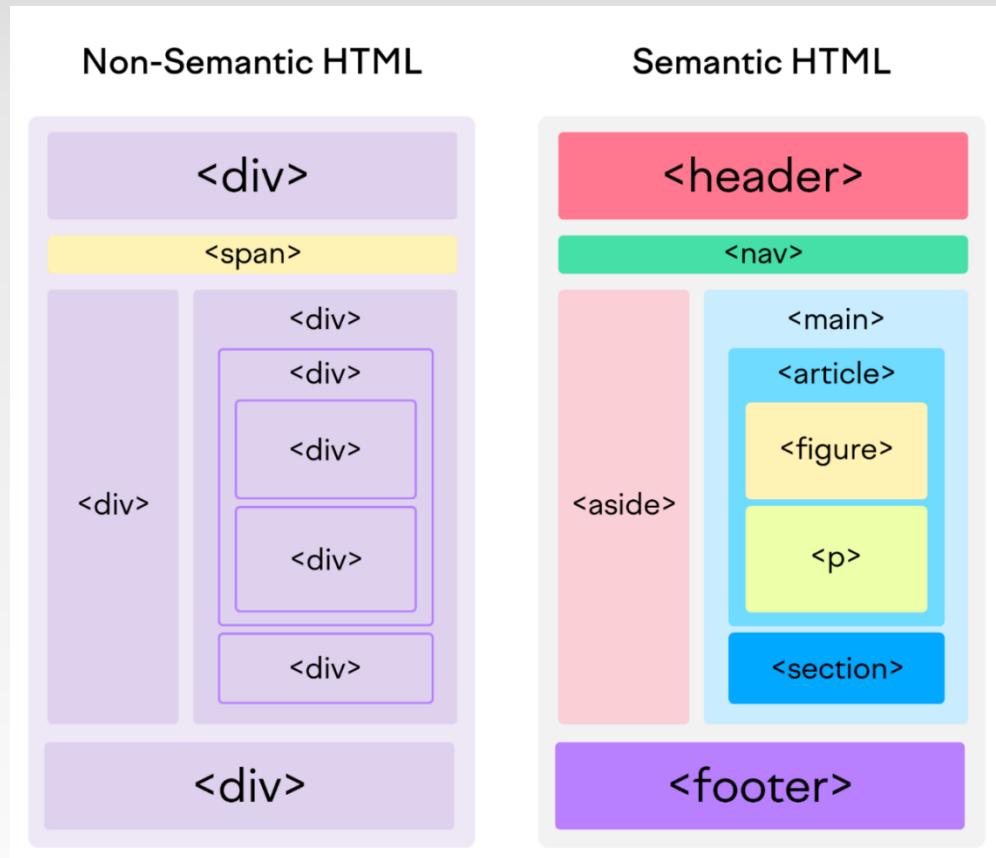




# Limbajul HTML și CSS

Ușurează munca unui web developer

# Structura HTML5



# Tabele

---

Tag-uri pentru un tabel sunt:

<table> și </table> - definește începutul unei tabele

<th> și </th> - definire header într-un tabel

<tr> și </tr> - începutul unei linii într-o tabelă

<td> și </td> - formează o celulă pe o linie dintr-o tabelă.

# Tabel

---

```
<table> Studentii de la seria AD <br> <hr/>
<tr>
    <td> Nume
    <td> Prenume
    <td> Grupa
</tr>
<tr>
    <td> Popescu
    <td> Razvan
    <td> 621
</tr>
</table>
```

Studentii de la seria AD

---

Nume	Prenume	Grupa
Popescu	Razvan	621

# Atribute table

---

**Border** – trasarea liniilor într-un tabel

**Width** – exprimă dimensiunea ocupată de tabel pe orizontală, în procente

**Height** - exprimă dimensiunea ocupată de tabel pe verticală

**Colspan și Rowspan** - pentru a îmbina mai multe linii sau coloane

- <table border="10" width=50% height="70">

Studentii de la seria AD		
Nume	Prenume	Grupa
Popescu	Razvan	621

<table border="5" width=70% height="90">

Studentii de la seria AD		
Nume	Prenume	Grupa
Popescu	Razvan	621

- <table rowspan=2>

Studentii de la seria AD		
Nume	Prenume	Grupa
	Popescu	Razvan
		621

# Aliniere tabele

---

Pe orizontală folosim următoarele atrbute:

- **Left**
- **Right**
- **Center**

Pe verticală folosim următoarele atrbute:

- **Valign= “top”**
- **Valign =Bottom**
- **Valign = Middle**

# Aliniere table

```
<table border=5 width=70% height=90> Studentii de la seria AD  
<br> <hr/>
```

```
<tr> <td align=left> Nume  
<td align=right> Prenume  
<td align=center> Grupa</tr>  
<tr><td align=top> Popescu  
<td align=bottom> Razvan  
<td align=middle> 621 </tr>
```

```
</table>
```

Studentii de la seria AD

Nume	Prenume	Grupa
Popescu	Razvan	621

# Formatare celule

---

- ❑ Width și Height pentru lățime și lungime celulă
- ❑ Inserare de imagini în celulă
- ❑ O celulă poate să primească aceleași attribute ca un tabel

# Formatare celule

---

Studentii de la seria AD

Nume	Prenume	Grupa
Popescu	Razvan	621

<tr>

<td align=left width=20% height=50> Nume

<td align=right width=50% height=70 >  
Prenume

<td align=center width=90% height=30> Grupa  
</tr>

<tr><td align=top width=15% height=30>  
Popescu

<td align=bottom width=40% height=40>  
Razvan

<td align=middle width=70% height=40> 621

</tr>

# Colorăm tabelul

```
<table bgcolor="#BEFADE> Studentii de la seria AD <br>
<hr/><tr>

<td width=20% height=50 bgcolor="#FACADE> Nume
<td width=50% height=70 > Prenume
<td align=center width=90% height=30> Grupa</tr>

<tr bgcolor="#AFAD0E>

<td align=top width=15% height=30 > Popescu
<td align=bottom width=40% height=40> Razvan
<td align=middle width=70% height=40> 621 </tr>
</table>
```

Studentii de la seria AD		
Nume	Prenume	Grupa
Popescu	Razvan	621

# Încadrarea textului

---

Se pot adăuga opțiuni unui chenar astfel:

- { *overflow: valoare* }

Visible

Hidden

Scroll

Auto

# Încadrarea textului

---

```
.div6 {  
background-color: #eee;  
width: 200px;  
height: 50px;  
border: 1px dotted black;  
overflow: visible;  
}
```

Lorem ipsum dolor sit amet,  
consectetuer adipiscing elit,  
sed diam nonummy nibh  
euismod tincidunt ut laoreet  
dolore magna aliquam erat  
volutpat.

```
.div7 {  
background-color: #eee;  
border: 1px dotted black;  
overflow: hidden;  
}
```

Lorem ipsum dolor sit amet,  
consectetuer adipiscing elit,

# Încadrarea textului

---

```
.div7 {  
background-color: #eee;  
width: 200px;  
height: 100px;  
border: 1px dotted black;  
overflow: scroll;  
}
```

A avut o conferință de presă  
în cadrul căreia nu a vorbit  
cu presa din două motive:  
întâi că presa nu era de față  
*/am avut o întâlnire cu reprezentanții*

*overflow-x și overflow-y*

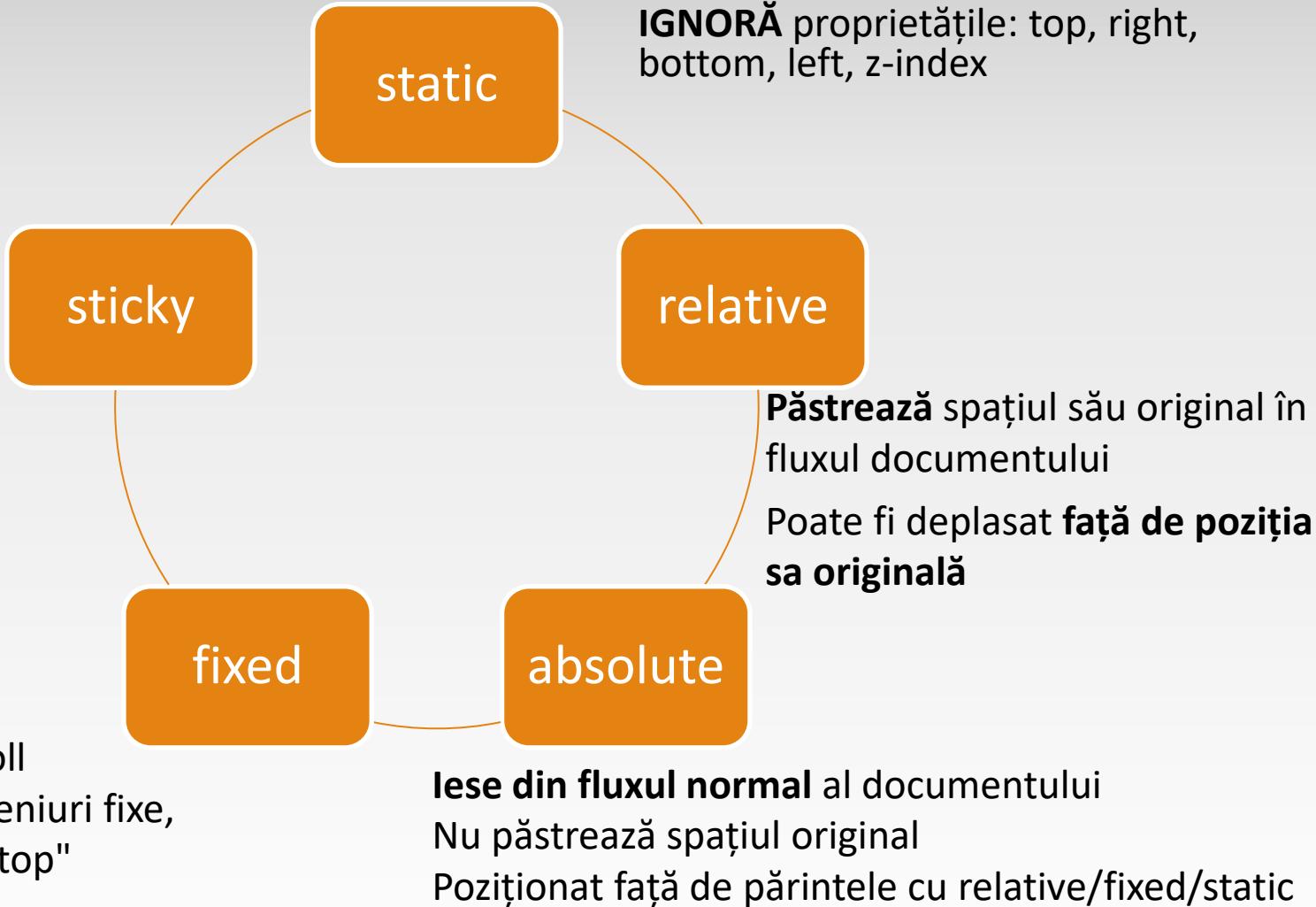
# Proprietatea position

Se comportă ca relative până când ajunge la o anumită poziție  
Apoi se comportă ca fixed

**Față de viewport**

Rămâne fix la scroll

Perfect pentru: meniuri fixe,  
butoane "back to top"



# Exemple position

```
.container {  
    /* position nu este setat */  
}
```

```
.child {  
    position: absolute;  
    top: 0;  
    left: 0;  
}
```

```
/* CORECT */  
.container {  
    position: relative; /* Acum child se  
    poziționează corect */  
}
```

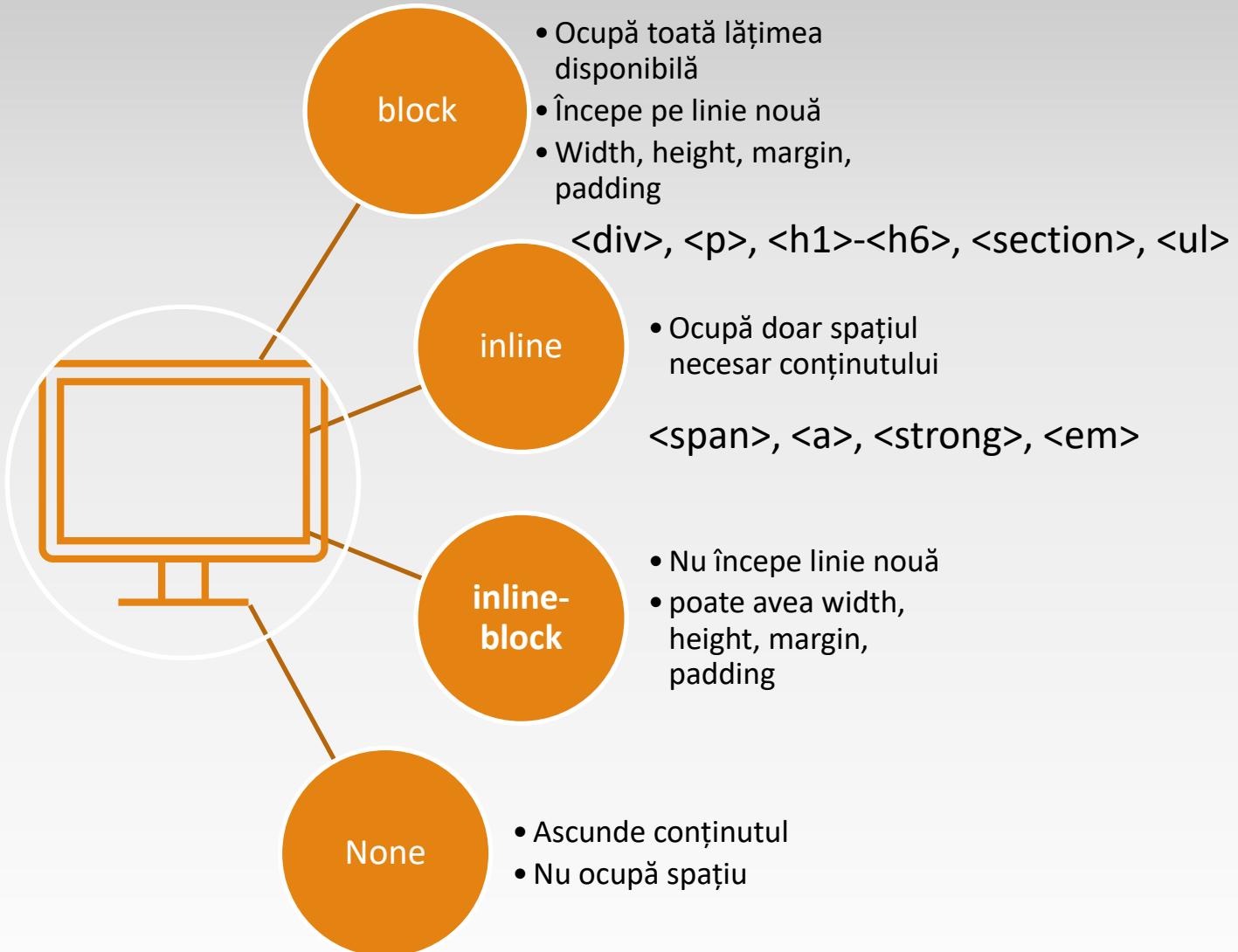
```
.header {  
    position: fixed;  
    top: 0;  
    height: 60px;  
}
```

```
.content {  
    /* Conținutul va fi acoperit de header */  
}
```

```
/* CORECT */  
.header {  
    position: fixed;  
    top: 0;  
    height: 60px;  
    width: 100%;  
    z-index: 1000;  
}
```

```
.content {  
    margin-top: 60px; /* Compensare */  
}
```

# Proprietatea display

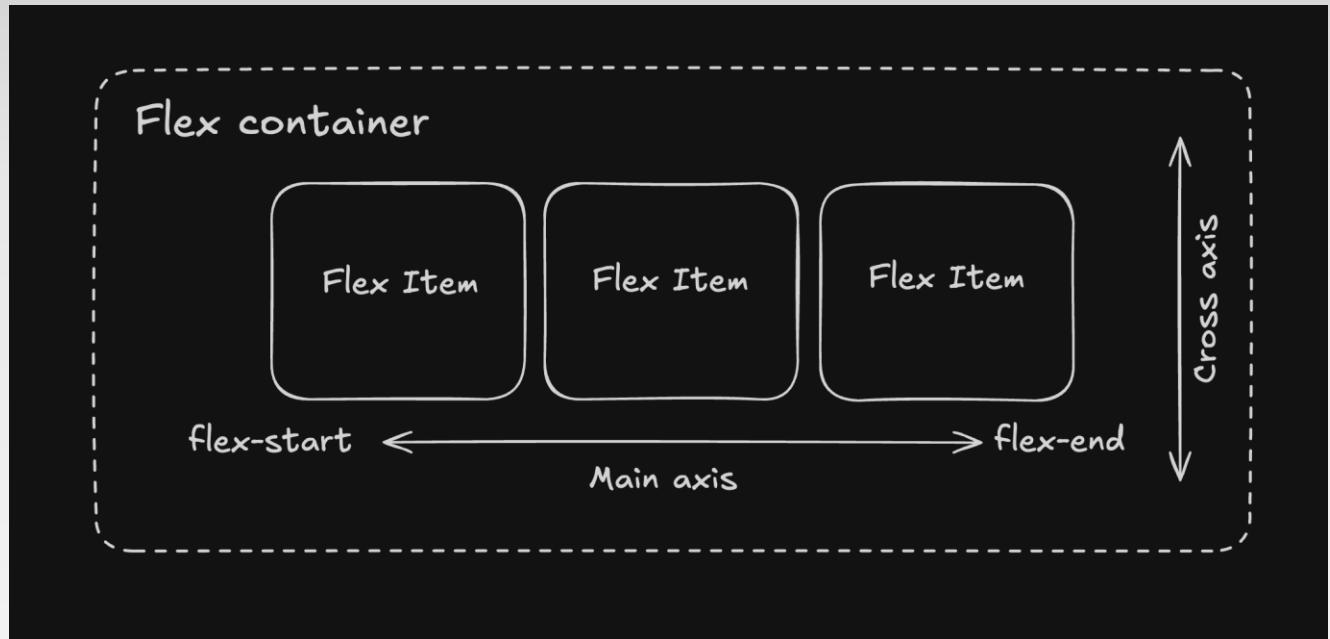


# Proprietatea display: flex

**Flexbox (Flexible Box Layout)** este un sistem modern de aranjare a elementelor

Se ocupă singur de:

- cât spațiu ia fiecare
- cum se aliniază între ele
- ce se întâmplă dacă fereastra se micșorează



Flex-direction: direcția elementelor

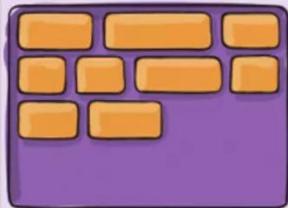
Justify-content: aliniere pe axa principală

Align-items: aliniere pe axa secundară

Flex-wrap: permite trecerea pe rânduri noi

## align-content

flex-start



flex-end



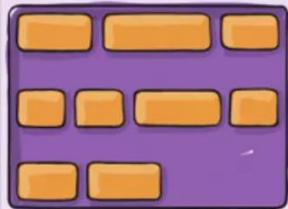
center



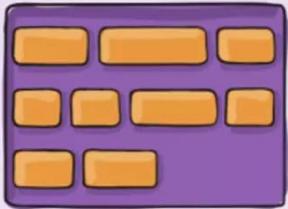
stretch



space-between

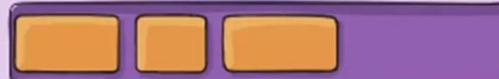


space-around

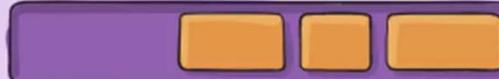


## justify-content

flex-start



flex-end



center



space-between



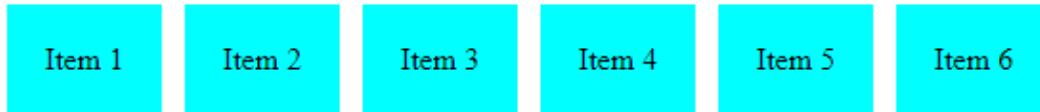
space-around



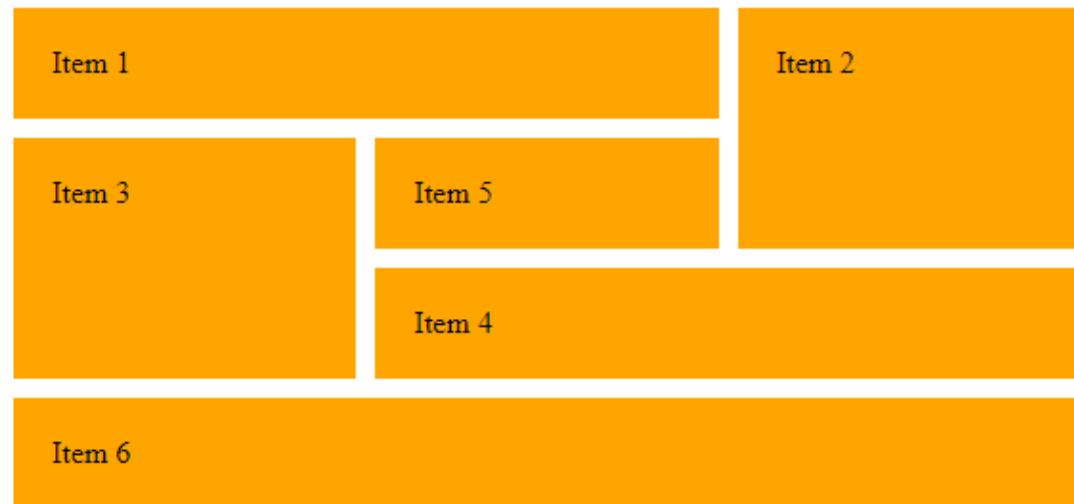
space-evenly



Display flex



Display grid



# Proprietary display: grid

---

# Responsive design cu Media Queries

---

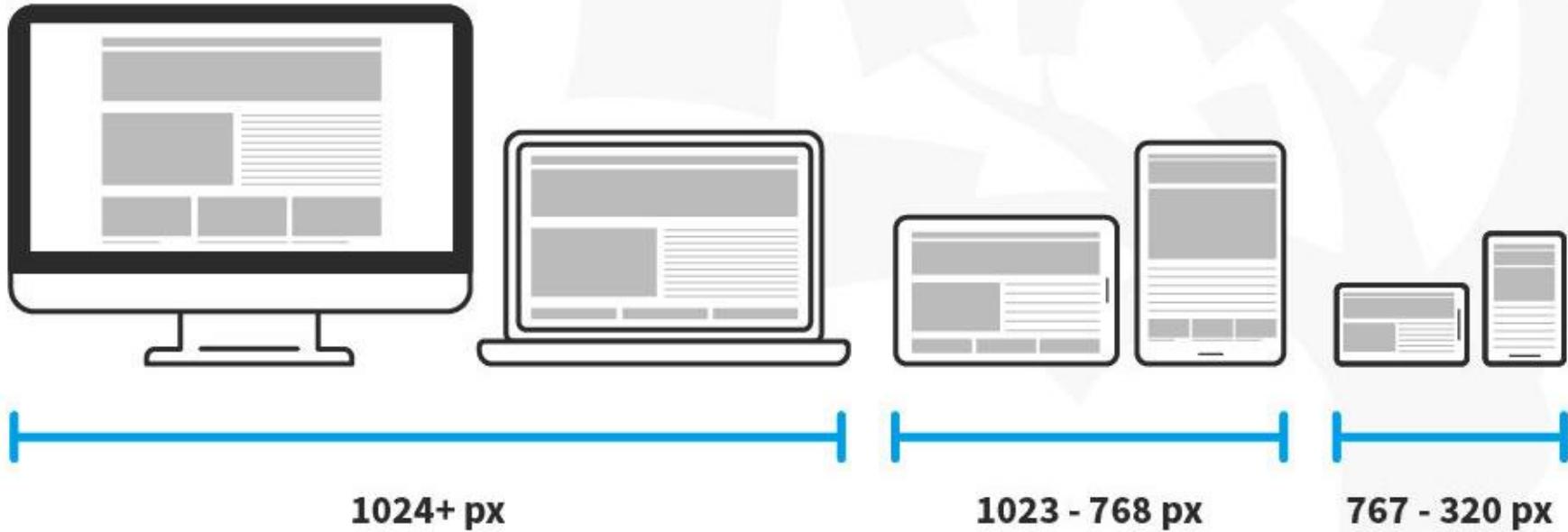
Media queries îți permit să **schimbi stilurile CSS în funcție de caracteristicile dispozitivului** (lățime ecran, orientare, preferințe user – dark mode, reduce motion etc.).

**Mobile-first** (recomandat): se scrie întâi stilul pentru mobil, apoi se adaugă reguli cu min-width pentru ecrane mai mari.

**Viewport meta** (obligatoriu în HTML):

`<meta name="viewport" content="width=device-width, initial-scale=1">`

# Responsive Design



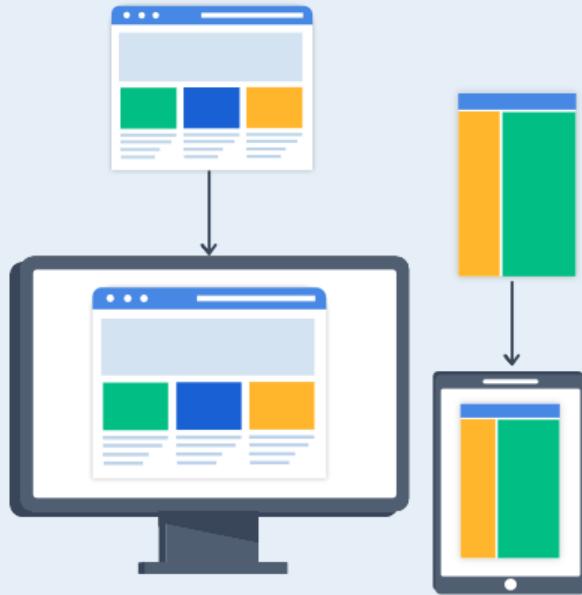
```
/* Mobile-first: se aplică peste stilul de bază când ecranul are  $\geq 768\text{px}$  */  
@media (min-width: 768px) { ... }
```

```
/* Desktop-first: pentru ecrane  $\leq 768\text{px}$  */  
@media (max-width: 768px) { ... }
```

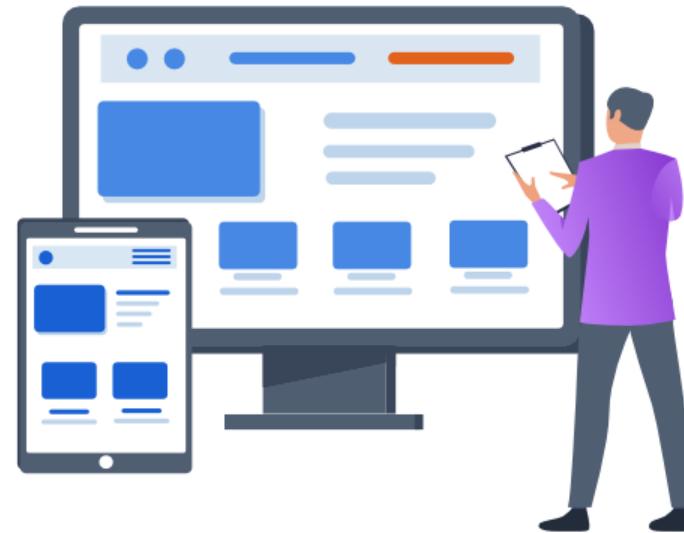
```
/* Interval modern (range context) – suport bun în browsere moderne */  
@media (400px  $\leq$  width  $\leq$  900px) { ... }
```



XENONSTACK



VS



# Adaptive Design

# Responsive Design

# Alegere design

---

## **RESPONSIVE:**

Budget limitat

Conținut similar pe toate dispozitivele

Vrei future-proof pentru dispozitive noi

Example: blog, site-uri corporative

## **ADAPTIVE:**

Experiențe radical diferite pe mobile vs desktop

Performanță critică pe mobile

Resurse pentru developement și menenanță

Example: e-commerce complex, aplicații web

## Exercițiu

Creați câte o pagină HTML cu stiluri interne pentru chenar după următoarele criterii:

A. Definiți 8 clase pentru paragraf, la care chenarul să fie definit în felul următor: 13

1. *dotted* – chenar punctat;
2. *dashed* – chenar cu line întreruptă;
3. *solid* – chenar cu linie simplă;
4. *double* – chenar cu două linii;
5. *groove* – chenar tip groove;

B. Definiți 4 clase pentru paragraf, la care chenarul va fi colorat diferite pe fiecare parte.

1. *one* – întreg chenarul de culoare albastră;
2. *two* – părțile de sus și jos roșii, iar din stânga și dreapta albastre;
3. *three* – partea de sus roșie, partea de jos albastră, părțile laterale verzi;
4. *four* – toate părțile chenarului de culori diferite.

C. Definiți o clasă de paragraf care cele patru părți ale chenarului de grosimi diferite: partea de sus 2 pixeli, partea de jos 5 pixeli, partea din stânga 20 de pixeli și partea din dreapta de 10 pixeli.

FRAMEWORK

Angular V18

Node.js

```
> .angular  
> .vscode  
> backend  
> node_modules  
└ src  
    └ app  
    └ assets  
    ★ favicon.ico  
    └ index.html  
TS main.server.ts  
TS main.ts  
# styles.css  
.editorconfig  
diamond .gitignore  
i README.md  
TS server.ts  
{} tsconfig.app.json  
TS tsconfig.json  
{} tsconfig.spec.json
```

# Creare proiect in VS Code cu Angular

In terminal: `ng new nume_proiect`

`cd /nume_proiect`  
`ng serve`

**/src** - directorul principal unde se află codul sursă al aplicației

**/src/app** - directorul unde se dezvoltă aplicația Angular propriu-zisă

**/src/assets** - director cu resurse statice: imagini, fișiere CSS, fonturi

**Index.html** - fișierul HTML principal al aplicației

**Styles.css** - Fișierul global de stiluri al aplicației

**Package.json** - Fisierul care gestionează dependențele aplicației

# Creare componentă

**componentele** sunt blocurile de bază ale unei aplicații și reprezintă elemente vizuale reutilizabile cu logica și structura lor proprie

---

**HTML**-ul structura componentei -template-ul.

**CSS**-ul care descrie cum ar trebui să fie stilizat.

**Logica** definită în TypeScript pentru a gestiona funcționalitățile și datele afișate.

## Cum creez o componentă?

În terminal: *ng generate component nume\_componenta*

