

# Crearea unei pagini web responsive

...

# HTML

HTML, sau HyperText Markup Language, este un limbaj care descrie structura unui document Web

# HTML

Fiecare element HTML are o semnificație anume, descriind atât conținutul cât și structura documentului.

Orice element poate avea un număr de attribute, majoritatea fiind specifice acelui element; toate elementele pot avea attributele `class` și `id`; `id` specifică un identificator unic pentru elementul în cauză, în timp ce `class` grupează logic elementele; un element poate avea mai multe clase dar un singur id.

# HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CV</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <main>
    <section id="despre">
      
      <h1>Prenume Nume</h1>
      <ul class="contact">
        <li><strong>Telefon:</strong> 07xx xxx xxx</li>
        <li><strong>Email:</strong> prenume.numa@email.com</li>
        <li><strong>LinkedIn:</strong> <a href="https://ro.linkedin.com/in/prenume-numa">
        <li><strong>Adresa:</strong> strada Principala nr 0, sector 7, Bucuresti, Romania
      </ul>
      <div>
        <h2>Despre mine</h2>
```

# HTML



## Prenume Nume

- **Telefon:** 07xx xxx xxx
- **Email:** prenume.numa@email.com
- **LinkedIn:** <https://ro.linkedin.com/in/prenume-nume>
- **Adresa:** strada Principala nr 0, sector 7, Bucuresti, Romania

## Despre mine

Sunt o persoana foarte fictiva, careia ii place sa pretinda ca exista atunci cand se mai tine cate-un workshop din acesta.

Pasiunile mele sunt diverse si submine dar lipsesc cu desavarsire.

Sunt motivat de motivatii, pasiuni si interese. Imi displac lucrurile neplacute.

Exemplu live  
HTML

...

# CSS

CSS, sau Cascading Style Sheets, este o tehnologie web care permite modificarea aspectului unui document web.

Principalul avantaj al CSS este cascada, adică faptul că orice modificare aplicată unui element este aplicată tuturor elementelor conținute de acest element

# CSS

```
html {  
  font-family: sans-serif;  
}  
  
main {  
  display: flex;  
  width: 80%;  
  margin: 0 auto;  
}  
  
section {  
  padding: 1em;  
}  
  
ul {  
  list-style: none;  
  padding: 0;  
}  
  
section > * + div {  
  margin-top: 2em;  
  border-top: 1px solid;  
}
```





## Prenume Nume

**Telefon:** 07xx xxx xxx

**Email:** prenume.nume@email.com

**LinkedIn:** <https://ro.linkedin.com/in/prenume-nume>

**Adresa:** strada Principala nr 0, sector 7,  
Bucuresti, Romania

---

## Despre mine

## Experienta

Perioada	Compania	Responsabilitati
----------	----------	------------------

2018 - 2021	<a href="#">Google</a>	Logo, algoritm de cautare, fundal de pagina
----------------	------------------------	--

2016 - 2018	<a href="#">facebook</a>	Nuanta de albastru, emoticon
----------------	--------------------------	------------------------------------

2012 - 2016	<a href="#">hi5</a>	Selfie-uri facute in baie
----------------	---------------------	------------------------------

---

## Educatie

Perioada	Unitate de invatamant
----------	-----------------------

2008	Universitatea Fictiva
------	-----------------------

Exemplu live  
CSS

...

# CSS

CSS folosește o sintaxă limitată

```
selector {  
    atribut-1: valoare;  
    atribut-2: valoare;  
}
```

# CSS

Selectorii identifică elementul căruia i se aplica stilurile specificate

- Element: orice nume al unui tag HTML (exp. `body`, `h1`, `a`)
- Clasă: un identificator care se poate repeta pentru mai multe elemente; aceeași clasă se poate atribui indiferent de element (exp. `.fundal`, `p.fundal`)
- ID: un identificator unic pe pagină; orice element poate avea un ID, atâta vreme cât acesta e unic (exp. `#logo`, `div#continut`)

# CSS

Selectorii se pot combina, ierarhia lor fiind aplicată de la stânga la dreapta

- `body div` = toate elementele `div` care sunt conținute în `body`
- `div > p` = toate paragrafele conținute într-un `div`
- `section + section` = toate secțiunile care urmează imediat după alta secțiune
- `ul ~ ul` = toate listele neordonate care sunt copii ai aceluiași element părinte

# CSS

Selectorii pot deveni foarte complecși:

`body.dark-mode article > h2:nth-child(even) a:hover` = toate linkurile pe care se afla cursorul mouseului, care linkuri sunt descendenți ai unui titlu, numai dacă acest titlu este unul par și descendent direct al unui articol, dacă articolul este într-un document al cărui element body are clasa “dark-mode”

# Exemplu live

## SELECTORI CSS

...

# Așezarea în pagină

Ierarhia documentului nu este strict necesară, dar este utilă pentru a delimita logic secțiunile de interes; în funcție de ele, putem specifica aspectul paginii



# Așezarea în pagină

## 1. `table`

- Tot conținutul paginii este încadrat într-unul sau mai multe tabele
- Se poate specifica exact ordinea și aranjarea în pagină a tuturor elementelor
- Ordinea și aranjarea în pagină sunt constante
- Nu se mai folosește de ~20 ani

# Așezarea în pagină

## 2. `float`

- Combinația dintre lățimea / înălțimea elementelor și fluxul lor în pagină determină ordinea și aranjarea elementelor
- Toate dimensiunile trebuie specificate exact
- Capătul de linie este un caz special
- Nu se mai folosește de ~ 15 ani

# Așezarea în pagină

## 3. flex

- Tehnologia “flex” (sau “flexbox”) permite crearea de elemente “flexibile”, ale căror dimensiuni pot fi specificate mult mai general
- Elementele ocupă tot spațiul disponibil, cu opțiunea de a le modifica fluxul pentru a ocupa mai multe rânduri
- Tehnologie modernă, suportată de majoritatea browserelor

# Așezarea în pagină

## 4. grid

- Cea mai recentă tehnologie când vine vorba de aranjarea în pagină
- Folosită concomitent cu flex
- Specifică aspectul paginii în rânduri și coloane

# Exemplu live

## FLEXBOX

...

# Tipografie

Textul unui document poate avea unul sau mai multe fonturi; în general, se preferă un font pentru conținut și unul “opus” pentru titluri

# Tipografie

Fonturile sans-serif (Arial, Helvetica, Futura) tind sa fie drepte, geometrice, optime pentru titluri

# Tipografie

Fonturile serif (Times New Roman, Garamond, Palatino) au serife (codițele de la capătul elementelor drepte), și sunt optime pentru conținut



# Tipografie

Fonturile monospace (Courier, Consolas) sunt cele ale căror caractere au toate o lăţime identică; sunt folosite pentru exemple de cod

# *Tipografie*

Fonturile caligrafice (Copperplate, Papyrus, Lobster) sunt destinate pentru a atrage atenția asupra conținutului

# Tipografie

În general, nu se recomandă folosirea a mai mult de 2 fonturi pe pagină

# Exemplu live

# TIPOGRAFIE

...

# Culori

Browsersle moderne suportă un număr foarte mare de culori--peste 16 milioane.

Culorile pot fi specificate în mai multe feluri: după nume, după codul hexadecimal asociat culorii, sau după nuanță și luminozitate

# Culori

## Nume

140 de culori au nume speciale; de exemplu, `orangered`

# Culori

## Cod hexadecimal

Culorile pot fi specificate după un cod de forma `#000000`; acestea reprezintă combinațiile de roșu, verde și albastru, de la 0 la 255, în funcție de luminozitate; astfel, culoarea portocaliu poate fi reprezentată ca `#FF4500`, adică roșu la 255 (`FF` în hexadecimal), verde la 69 (`45` în hex) și albastru deloc (`00` în hex); dacă intensitatea culorii este un multiplu de 16, fiecare a 2a literă poate fi omisă (exp `#F50`)

# Culori

## Cod numeric în roșu-verde-albastru

Culorile pot fi specificate sub forma `rgb (255, 69, 0)`; valorile sunt aceleași ca în exemplul anterior, dar reprezentarea numerică e în baza 10, nu 16.

Numele funcției vine de la numele Red-Green-Blue



# Culori

## Cod numeric în funcție de nuanță și luminozitate

Culorile pot fi specificate sub forma `hsl(16deg 100% 50%)`; aceasta înseamnă 160 de grade în spectrul vizibil (între 0 și 360 grade, ROGVAIVR -- 360 și 0 corespund ambele culorii roșu), saturație de 100% și luminozitate de 50%

Numele funcției vine de la Hue-Saturation-Lightness

# Culori

Pentru transparență, se pot specifica 4 numere--portocaliu semi-transparent poate avea oricare din aceste forme:

- `#FF450080`
- `rgb(255, 69, 0 / 50%)`
- `hsl(16deg 100% 50% / 50%)`

Exemplu live  
CULORI

...

# Pagini responsive

Atunci cand o pagină web are un element care să-i specifice o anume proprietate (meta-date, informații despre pagină, nu despre conținutul său), putem modifica aspectul ei în funcție de dimensiunea și orientarea dispozitivului de pe care o accesăm; Aceste proprietăți se numesc “media queries”, adică “întrebări despre mediul de vizualizare”

# Pagini responsive

Se pot specifica:

- lățimea dispozitivului
- orientarea (verticală sau orizontală)
- mediul în sine (web sau print)
- setări ale browserului și/sau ale sistemului de operare (dark mode, reduced motion)

# Pagini responsive

Media Queries sunt suportate de majoritatea browserelor moderne, dar nu de toate; tactica cea mai eficientă e să optimizăm o pagină pentru mobil, schimbând stilurile pentru dispozitive mai mari

# Exemplu live

## MEDIA QUERIES

...

# Exemplu practic

...