

## TEHNOLOGII WEB

### SUPORT DE CURS OPȚIONAL CA DISCIPLINĂ NOUĂ

Opțional elaborat în cadrul Proiectului de Accreditare Erasmus+

Nr. de referință 2022-1-RO01-KA121-SCH-000051974

finanțat de Uniunea Europeană în cadrul Programului ERASMUS+

**Aria curriculară:** Tehnologii

**Tipul de opțional:** opțional ca disciplină nouă

**Clasa a XI –a B** , profil *real*, specializare *Științe ale naturii*

**Durata:** 1 an (1 oră/săptămână)

**Propunător:** prof. Andrioea Viorel

## Cuprins

|                                                                          |    |
|--------------------------------------------------------------------------|----|
| I. Etica conținuturilor web.....                                         | 3  |
| II. Conținuturi web .....                                                | 7  |
| II.1. Text.....                                                          | 7  |
| II.2. Imagini – formatare și aranjare în pagină .....                    | 9  |
| II.3. Conținut animat .....                                              | 11 |
| II.4. Formulare.....                                                     | 13 |
| II.5. Sunete .....                                                       | 19 |
| II.6. Clipuri video .....                                                | 20 |
| II.7. Hărți de imagini .....                                             | 21 |
| III. Prelucrarea conținuturilor web .....                                | 24 |
| III.1. GoogleSites - Formatarea din interfața grafică a aplicației ..... | 24 |
| III.2. Utilizarea foilor de stil.....                                    | 27 |
| III.3. Foi de stil interne.....                                          | 28 |
| III.4. Foi de stil inline.....                                           | 29 |
| III.5. Foi de stil externe .....                                         | 30 |
| IV. Structurarea site-ului .....                                         | 35 |
| IV.1. Organizarea sub formă de pagină unică .....                        | 35 |
| IV.2. Organizarea în mai multe pagini .....                              | 38 |
| IV.3. Organizarea în mai multe foldere.....                              | 41 |
| IV.4. Harta site-ului.....                                               | 44 |
| V. Verificări înainte de publicare .....                                 | 48 |
| IV.1. Analiza rezultatului obținut.....                                  | 48 |
| VI. Publicarea.....                                                      | 50 |

## I.Etica conținuturilor web

Există o varietate de tipuri de informații care pot fi publicate în mediul online, iar acestea pot varia în funcție de platforma sau website-ul specific care urmează să fie folosit. În general, informațiile care pot fi publicate online includ:

- ✓ **text:** articole, știri, bloguri, eseuri, recenzii și alte forme de conținut textual
- ✓ **imagini:** fotografii, ilustrații, desene, precum și alte tipuri de imagini
- ✓ **video:** filme, videoclipuri, vloguri, tutoriale
- ✓ **audio:** podcasturi, melodii, înregistrări audio
- ✓ **documente:** documente text, prezentări, fișiere PDF sau alte formate
- ✓ **date și statistici:** informații statistice, rapoarte, seturi de date precum și alte informații de natură numerică
- ✓ **social media:** mesaje, comentarii, postări, actualizări de status și alte forme de conținut social

Există, totuși, anumite tipuri de informații care nu ar trebui să fie publicate în mediul online. Acestea includ:

- ✓ **informații personale sensibile:** informații precum CNP-ul, numărul cărții de credit, PIN-ul card-ului, adresa de domiciliu, datele personale ale minorilor sau alte informații personale sensibile deoarece ar putea fi expuse unor riscuri de fraudă sau abuz
- ✓ **date medicale confidențiale:** Informații despre sănătatea personală, istoricul medical sau alte informații confidențiale legate de sănătate ar trebui să fie protejate și nu ar trebui publicate online din motive de confidențialitate și securitate
- ✓ **informații juridice confidențiale:** informații legate de cazuri juridice în curs, documente juridice confidențiale sau alte informații protejate de confidențialitatea avocat-client

- ✓ **conținut ilegal sau dăunător:** orice formă de conținut care este ilegal, inclusiv conținut pornografic, incitare la violență sau alte forme de conținut care ar putea provoca rău

Este importantă conștientizarea consecințelor publicării anumitor informații și respectarea confidențialității, drepturile de autor și a legilor în vigoare atunci când este utilizat mediul online. Este important să fie luate în considerare impactul social al informațiilor publicate și evitarea răspândirii de informații false sau prejudiciabile.

De asemenea, este întotdeauna recomandată atenție și responsabilitate în ceea ce privește informațiile publicate în mediul online. Trebuie acordată atenție setărilor de confidențialitate și politici de utilizare a platformei respective pentru a proteja confidențialitatea și securitatea propriei persoane, precum și pentru a respecta drepturile și regulile impuse de platforma respectivă.

Respectarea dreptului de autor în mediul online este crucială pentru a proteja creațiile și drepturile autorilor. Câteva dintre principalele reguli în ceea ce privește dreptul de autor în mediul online:

- ✓ utilizarea conținutului cu drept de autor: nu poate fi utilizat sau republicat conținutul (text, imagini, video, muzică etc.) creat de alți autori fără permisiunea acestora sau fără a respecta regulile privind utilizarea justă sau alte excepții legale specifice. Pentru a utiliza conținutul cu drept de autor, este necesar să fie obținută permisiunea explicită a autorului sau să existe sprijinul licențelor creative sau alte tipuri de licențe care permit utilizarea anumitor tipuri de conținut în anumite condiții
- ✓ citații și referințe corecte: atunci când sunt utilizate informații / fragmente dintr-o sursă cu drept de autor, este importantă citarea și referențierea sursei corect, astfel încât să se recunoască dreptul de autor și să se ofere credit autorului original
- ✓ respectarea termenilor și condițiilor platformelor online: este importantă citirea și respectarea acestor termene și condiții atunci când este utilizată o platformă

online și obținerea permisiunii corespunzătoare înainte de a fi utilizat sau partajat conținutul creat de alții

- ✓ evitarea încălcării drepturilor de autor: nu trebuie copiat / modificat sau distribuit conținut protejat de drepturi de autor fără permisiunea autorului. Acest lucru include și evitarea pirateriei software sau descărcarea ilegală a materialelor protejate de drepturi de autor, cum ar fi filme, muzică sau cărți
- ✓ responsabilitatea pentru conținutul propriu: ca autor online, ești responsabil de conținutul pe care îl creezi și îl publici. Asigură-te că ai drepturile necesare pentru a-ți publica propriul conținut și că nu sunt încălcate drepturile de autor ale altor persoane.

Regulile și legislația privind dreptul de autor pot varia în funcție de țară și pot exista excepții specifice în anumite situații. În cazul în care apar nelămuriri specifice legate de dreptul de autor, este întotdeauna recomandată consultarea legilor și a reglementărilor locale ori solicitarea unei consultanțe juridice.

Câteva aspecte generale legate de legislația privind încălcarea dreptului de autor în mediul online:

- ✓ drepturile de autor: legislația protejează creațiile originale, cum ar fi texte, imagini, muzică, filme și software, prin drepturile de autor. Aceste drepturi conferă autorilor controlul asupra modului în care lucrările lor sunt utilizate, reproduse sau distribuite.
- ✓ încălcarea dreptului de autor: încălcarea dreptului de autor are loc atunci când cineva utilizează, reproduce, distribuie sau modifică o lucrare protejată fără permisiunea autorului sau fără a se încadra într-o excepție legală. Acest lucru poate include încărcarea ilegală a conținutului protejat de drepturi de autor pe platforme de partajare sau distribuire ilegală a acestuia.
- ✓ consecințe legale: încălcarea dreptului de autor poate duce la consecințe legale. Autorii sau titularii drepturilor de autor pot lua măsuri legale împotriva celor care încalcă drepturile lor, inclusiv prin trimiterea de notificări de încetare și dezinvestire, formularea de acuzații de încălcare sau intentarea de acțiuni în

justiție. În cazul unei condamnări, sancțiunile pot include amenzi, despăgubiri și chiar pedepse penale în unele cazuri.

- ✓ excepții și utilizarea justă: legislația privind dreptul de autor poate include și anumite excepții sau utilizări considerate justificate fără a obține permisiunea autorului. Aceste excepții pot include utilizarea în scopuri educaționale, critică, recenzie sau știri, în funcție de legislația specifică a fiecărei țări.

## II. Conținuturi web

### II.1. Text

Pentru a insera text într-o pagină HTML, se folosește elementul `<p>` (paragraf) pentru a crea un nou paragraf și a scrie textul în interiorul acestuia. De exemplu:

```
<p>Acesta este un paragraf de text.</p>
```

Pentru text îngroșat, se poate utiliza elementul `<strong>` ca în exemplu:

```
<p>Aceasta este un <strong>text îngroșat</strong> </p>
```

Pentru text italic, se poate folosi elementul `<em>` ca în exemplul următor:

```
<p>Exemplu de <em>text italic</em> </p>
```

Pentru a sublinia textul, se utilizează elementul `<u>` astfel:

```
<p>Aceasta este un <u>text subliniat</u>.</p>
```

Pentru a crea o listă cu marcatori, se folosesc elementele `<ul>` și `<li>`:

```
<ul>  
  <li>Element 1</li>  
  <li>Element 2</li>  
  <li>Element 3</li>  
</ul>
```

Poziționarea textului în pagină poate fi realizată utilizând stiluri CSS (Cascading Style Sheets). Se pot utiliza proprietățile CSS, cum ar fi ***text-align*** pentru alinierea textului, ***padding*** și ***margin*** pentru a crea spații între elementele de text sau ***float*** pentru a poziționa textul în raport cu alte elemente din pagină. Spre exemplu, pentru a alinia textul la centru:

```
<p style="text-align: center;">Acesta este un paragraf aliniat la centru.</p>
```

Pentru a specifica un font pentru un text se poate folosi tagul `<span>` și stilurile CSS. Se pot utiliza proprietățile CSS *font-family* și *font-size* ca în următorul exemplu:

```
<span style="font-family: Arial, sans-serif;"> text afișat cu fontul Arial</span>
```

Stilul CSS specifică fontul *Arial* ca primă opțiune; în cazul în care acesta nu este disponibil, browserul va utiliza un font *sans-serif* ca înlocuitor.

Important de reținut că utilizarea stilurilor CSS separate este preferabilă în comparație cu stilurile în linie, așa cum este prezentat în exemplul de mai sus. Astfel, se recomandă să fie definite stilurile CSS într-un fișier CSS separat și să fie aplicate utilizând clase sau selecții CSS. În următorul exemplu se definește un stil CSS într-un fișier extern și se aplică folosind o clasă:

în fișierul HTML se va scrie:

```
<span class="custom-font">Acesta este un text cu fontul  
Arial.</span>
```

iar în fișierul CSS extern:

```
.custom-font {  
  font-family: Arial, sans-serif;  
}
```

Utilizarea stilurilor CSS separate oferă un control mai mare asupra aspectului textului și permite o mai bună separare a conținutului de aspectul vizual.

**Temă de laborator:** Realizați în <http://www.canva.com> o prezentare în care se va explica modul de creare și utilizare a listelor în cadrul unui fișier HTML. Pentru a crea prezentarea se va folosi pentru conectare contul personal de gmail. Prezentarea creată va fi partajată pe Classroom. Se va folosi ca model, prezentarea partajată la adresa [https://www.canva.com/design/DAFoJ01HgFo/3O5mFrlhZy-IFHp39SMqwQ/edit?utm\\_content=DAFoJ01HgFo&utm\\_campaign=designshare&utm\\_medium=link2&utm\\_source=sharebutton](https://www.canva.com/design/DAFoJ01HgFo/3O5mFrlhZy-IFHp39SMqwQ/edit?utm_content=DAFoJ01HgFo&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton) .





## II.2. Imagini – formatare și aranjare în pagină

Pentru a putea fi inserată, imaginea trebuie să fie disponibilă local pe calculator sau să existe acces la ea prin intermediul unui URL de pe internet.

Se deschide editorul de cod / editorul HTML pentru a crea sau edita fișierul HTML în care urmează să fie inserată imaginea. Se poate folosi sintaxa specifică pentru a insera o imagine folosind o cale locală (de pe calculator) ca în acest exemplu:

```

```

în care **calea\_catre\_image.jpg** va fi înlocuită cu calea absolută sau relativă către imaginea ce urmează a fi inserată. De asemenea, se înlocuiește "**Descrierea imaginii**" cu o descriere relevantă a imaginii pentru a oferi informații alternative în cazul în care imaginea nu poate fi afișată. Se realizează apoi salvarea fișierului HTML și se deschide într-un browser web pentru a vedea imaginea afișată pe pagină.

Dacă se dorește inserarea unei imagini folosind un URL (de pe internet), se înlocuiește calea către imagine cu adresa URL completă ca în acest exemplu:

```

```

Utilizatorul trebuie să se asigure că are permisiunea de a utiliza și afișa imaginea, deoarece unele imagini pot fi protejate de drepturi de autor sau pot necesita aprobare. Aceasta este modalitatea de bază de a insera o imagine într-un document HTML.

Pot exista și alte atribute ce pot fi adăugate pentru a controla dimensiunile, stilul, linkurile și altele asociate cu imaginea.

Spre exemplu, pentru lățimea (**width**) și înălțimea (**height**) imaginii se folosesc atributele ca în exemplul următor în care valorile sunt exprimate în pixeli:

```

```

Se pot utiliza stiluri CSS inline pentru a aplica stiluri direct asupra imaginii folosind atributul **style** ca în exemplul următor:

```

```

În exemplul anterior, imaginea va avea un chenar roșu de grosime 1 pixel și un decalaj (față de margine) de 10 pixeli în jurul ei.

Adăugarea unui link la o imagine se realizează încapsulând imaginea într-un element **<a>** (ancoră) pentru a crea un link către o altă pagină sau o altă resursă.

Exemplu:

```
<a href="pagina_destinatie.html">  
    
</a>
```

În exemplul de mai sus, atunci când un utilizator vor face clic pe imagine, va fi direcționat către pagina **pagina\_destinatie.html**. Se poate înlocui **pagina\_destinatie.html** cu adresa URL sau calea relativă către locația dorită.

Pentru a alinia o imagine se folosește atributul **align** pentru a alinia imaginea în raport cu textul înconjurător.

```

```

În acest exemplu, imaginea va fi aliniată în partea stângă a textului înconjurător. Se poate folosi și valoarea **right** pentru a alinia imaginea în dreapta.

Se pot utiliza stiluri CSS pentru a alinia imaginea folosind atributul **style**.

```

```

În acest exemplu, imaginea va fi aliniată central orizontal în cadrul elementului care o conține. Folosind **margin-left: auto** și **margin-right: auto**, imaginea va avea margini egale în stânga și dreapta, ceea ce va duce la centrarea ei.

Se poate defini o clasă CSS și apoi aplicarea acesteia pe eticheta imaginii ca în exemplul următor:

```
<style>
```

```

.aliniere-stanga {
  float: left;
}
.aliniere-dreapta {
  float: right;
}
.aliniere-centrala {
  display: block;
  margin-left: auto;
  margin-right: auto;
}
</style>




```

În acest exemplu, s-au definit trei clase CSS (aliniere-stanga, aliniere-dreapta, aliniere-centrală) care aplică stiluri specifice de aliniere. Acestea pot fi aplicate pe eticheta imaginii pentru a o alinia în funcție de nevoi.

**Temă de laborator:** Realizați în <http://www.canva.com> o prezentare în care va fi prezentat pas cu pas cum poate fi realizată o galerie foto pentru un magazin online în HTML. Toți elevii clasei vor avea acces la editarea aceleiași prezentări. După ce vă conectați la contul vostru de gmail, puteți accesa prezentarea (în vederea completării) la adresa:

[https://www.canva.com/design/DAFoJ22YTrg/tBz8DJFeGNXDGqLwsnFiuQ/edit?utm\\_content=DAFoJ22YTrg&utm\\_campaign=designshare&utm\\_medium=link2&utm\\_source=sharebutton](https://www.canva.com/design/DAFoJ22YTrg/tBz8DJFeGNXDGqLwsnFiuQ/edit?utm_content=DAFoJ22YTrg&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton)

### II.3. Conținut animat

Se pot crea animații folosind chei de animație și tranzații CSS. Se definește o clasă CSS care descrie animația dorită și apoi se aplică această clasă la elementul HTML ce urmează a fi animat.

```

<style>
@keyframes slide {

```

```

    0% { transform: translateX(0); }
    100% { transform: translateX(100px); }
  }
  .animated-element {
    animation: slide 2s linear infinite;
  }
</style>
<div class="animated-element">Conținut animat</div>

```

În acest exemplu, am definit o animație CSS numită **slide** care utilizează **transform: translateX()** pentru a deplasa conținutul în dreapta. Apoi, am aplicat această animație la un element **<div>** cu clasa **.animated-element**. Elementul va fi animat într-un ciclu continuu timp de 2 secunde.

Se poate utiliza JavaScript pentru a crea și controla animațiile în HTML folosind biblioteci sau framework-uri JavaScript precum jQuery, GreenSock sau CSS Animations API pentru a simplifica procesul de animare.

```

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
  $(document).ready(function() {
    $('.animated-element').animate({marginLeft: '100px'}, 2000);
  });
</script>
<div class="animated-element">Conținut animat</div>

```

În acest exemplu, am folosit jQuery pentru a anima elementul **.animated-element**. Prin intermediul funcției **animate()**, am specificat o animație care va deplasa conținutul cu 100 de pixeli spre dreapta în 2 secunde.

**Temă de laborator:** Realizați în <https://coggle.it/> o hartă conceptuală în care va fi introdus în centru cuvântul ANIMAȚIE iar pe ramuri domenii / tipuri de activități în care utilizarea animațiilor în paginile lor web ar aduce avantaje. În continuarea ramurilor prezentați cel puțin un avantaj pentru fiecare domeniu identificat (vor fi minimum 5 domenii de activitate). Pentru a crea o hartă conceptuală pe această platformă vă veți conecta cu contul de gmail. Harta conceptuală va fi partajată pe Classroom. Se va folosi ca model, exemplul postat la adresa:

<https://coggle.it/diagram/ZKrUFZDbJNK99bqD/t/-/1df2d1abde1fa2c5d0560efd78abda5156dd4e2f6c1a42adbf0607295db0942e>



## II.4. Formulare

Într-un formular HTML, se pot introduce diferite obiecte pentru a permite utilizatorilor să completeze și să trimită informații către server. Iată câteva dintre principalele obiecte care pot fi inserate într-un formular HTML:

1) Casetă text (***input type="text"***): utilizată pentru introducerea de texte sau șiruri de caractere scurte.

Sintaxa: ***<input type="text" name="nume\_camp" id="id\_camp">***

Se înlocuiește "***nume\_camp***" cu un nume unic pentru câmpul respectiv și "***id\_camp***" cu un identificator unic (opțional) pentru manipularea câmpului utilizând JavaScript.

2) Zonă de text (textarea): folosită pentru a introduce de texte sau șiruri de caractere lungi.

Sintaxa:

***<textarea name="nume\_camp" id="id\_camp" rows="4" cols="40"></textarea>***

Se înlocuiește "***nume\_camp***" și "***id\_camp***" cu valorile corespunzătoare. Atributele "***rows***" și "***cols***" controlează înălțimea și lățimea zonei de text în rânduri și coloane.

3) Câmp pentru selectare (select): este utilizat pentru a permite utilizatorului să selecteze una sau mai multe opțiuni dintr-o listă.

Sintaxa:

```
<select name="nume_camp" id="id_camp">
  <option value="valoare1">Opțiune 1</option>
  <option value="valoare2">Opțiune 2</option>
  <option value="valoare3">Opțiune 3</option>
</select>
```

În care se înlocuiește "**nume\_camp**" și "**id\_camp**" cu valorile corespunzătoare. În interiorul etichetei **<select>**, utilizează etichetele **<option>** pentru a defini opțiunile disponibile.

4) Câmp pentru selectare multiplă (select cu atributul **multiple**): este similar cu câmpul pentru selectare, dar permite utilizatorului să selecteze mai multe opțiuni.

Sintaxa:

```
<select name="nume_camp" id="id_camp" multiple>
  <option value="valoare1">Opțiune 1</option>
  <option value="valoare2">Opțiune 2</option>
  <option value="valoare3">Opțiune 3</option>
</select>
```

Adaugă atributul "**multiple**" la eticheta **<select>** pentru a permite selecția multiplă.

5) Butoane radio (**input type="radio"**): sunt utilizate pentru a permite utilizatorului să selecteze o singură opțiune din mai multe. Exemplu complet de utilizare a acestor butoane:

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemplu de butoane radio</title>
</head>
<body>
  <h1>Culoarea preferată</h1>

  <form>
    <input type="radio" id="rosu" name="culoare" value="rosu">
    <label for="rosu">Roșu</label><br>

    <input type="radio" id="verde" name="culoare" value="verde">
    <label for="verde">Verde</label><br>
```

```

    <input type="radio" id="albastru" name="culoare"
value="albastru">
    <label for="albastru">Albastru</label><br>

    <input type="submit" value="Trimite">
</form>

<script>
    const form = document.querySelector('form');
    form.addEventListener('submit', function(event) {
        event.preventDefault();

        const selectedColor =
document.querySelector('input[name="culoare"]:checked').value;
        console.log("Culoarea selectată este: " + selectedColor);
    });
</script>
</body>
</html>

```

6) Butoane de selectare multiplă (***input type="checkbox"***) sunt utilizate pentru a permite utilizatorului să selecteze una sau mai multe opțiuni dintr-un set de opțiuni.

Sintaxa:

```

<input type="checkbox" name="nume_camp" value="valoare1"
id="id_camp1">
<label for="id_camp1">Opțiune 1</label>

```

În care se înlocuiește "***nume\_camp***" și "***id\_camp1***" cu valorile corespunzătoare. Eticheta ***<label>*** este utilizată pentru a oferi o etichetă descriptivă asociată cu butonul de selectare multiplă.

7) Buton de trimitere (***input type="submit"***): este utilizat pentru a permite utilizatorului să trimită formularul către server.

Sintaxa: ***<input type="submit" value="Trimite">***

8) Buton de resetare (***input type="reset"***): utilizat pentru a reseta valorile formularului la starea inițială.

Sintaxa: **<input type="reset" value="Resetare">**

Atributul "value" specifică textul afișat pe butonul de resetare.

9) Inputul de tip "file" (**input type="file"**) în HTML permite utilizatorului să selecteze un fișier de pe dispozitivul lor local.

Sintaxa: **<input type="file" name="nume\_camp" id="id\_camp">**

Unde "**nume\_camp**" reprezintă numele unic al câmpului și va fi utilizat pentru a identifica fișierul pe server după trimiterea formularului; "**id\_camp**" este un identificator unic (opțional) care poate fi utilizat pentru a manipula câmpul utilizând JavaScript.

Exemplu de utilizare:

```
<form action="/upload" method="POST" enctype="multipart/form-data">  
  <label for="file">Selectează un fișier:</label>  
  <input type="file" name="file" id="file">  
  <input type="submit" value="Încarcă">  
</form>
```

În acest exemplu, am creat un formular care va fi trimis la adresa **"/upload"** prin metoda HTTP POST. Eticheta **<label>** este utilizată pentru a afișa o etichetă descriptivă asociată cu inputul de tip "file". Inputul de tip **"file"** are atributul **"name"** setat la **"file"**, ceea ce înseamnă că numele fișierului va fi trimis cu această denumire pe server.

După completarea formularului, utilizatorul poate apăsa butonul **"Încarcă"** pentru a trimite formularul împreună cu fișierul selectat.

Trebuie specificat atributul **"enctype"** al formularului setat la **"multipart/form-data"** pentru a permite încărcarea fișierelor. Aceasta permite serverului să primească și să proceseze fișierele încărcate prin formular.

10) Inputul de tip "date" (**input type="date"**) în HTML permite utilizatorului să selecteze o dată dintr-un calendar interactiv. Inputul de tip **"date"** permite utilizatorului să aleagă o dată specifică dintr-un calendar interactiv. Formatul datelor afișat în calendar poate varia în funcție de implementarea browserului și setările locale ale utilizatorului. Datele selectate vor fi trimise la server în formatul standardizat YYYY-MM-DD.

Sintaxa: **<input type="date" name="nume\_camp" id="id\_camp">**



În care: **"nume\_camp"** reprezintă numele unic al câmpului și va fi utilizat pentru a identifica data selectată pe server după trimiterea formularului iar **"id\_camp"** este un identificator unic (opțional) care poate fi utilizat pentru a manipula câmpul utilizând JavaScript.

Exemplu:

```
<form>
  <label for="date">Selectează o dată:</label>
  <input type="date" name="date" id="date">
  <input type="submit" value="Trimite">
</form>
```

În acest exemplu, s-a creat un formular simplu cu un input de tip "date". Eticheta **<label>** este utilizată pentru a afișa o etichetă descriptivă asociată cu inputul de tip **"date"**. Inputul de tip **"date"** are atributul **"name"** setat la **"date"**, ceea ce înseamnă că data selectată va fi trimisă cu această denumire pe server. După selectarea unei date din calendar, utilizatorul poate apăsa butonul **"Trimite"** pentru a trimite formularul împreună cu data selectată. Este important să menționez că afișarea și suportul pentru inputul de tip "date" pot varia între browsere. Unele browsere pot utiliza un calendar interactiv, în timp ce altele pot utiliza un câmp de text în care utilizatorul trebuie să introducă manual data într-un anumit format.

11) Inputul de tip "time" (input type="time") în HTML permite utilizatorului să selecteze o valoare pentru timp.

Sintaxa: **<input type="time" name="nume\_camp" id="id\_camp">**

În care **"nume\_camp"** reprezintă numele unic al câmpului și va fi utilizat pentru a identifica timpul selectat pe server după trimiterea formularului iar **"id\_camp"** este un identificator unic (opțional) care poate fi utilizat pentru a manipula câmpul utilizând JavaScript.

Exemplu:

```
<form>
  <label for="time">Selectează un timp:</label>
  <input type="time" name="time" id="time">
  <input type="submit" value="Trimite">
</form>
```

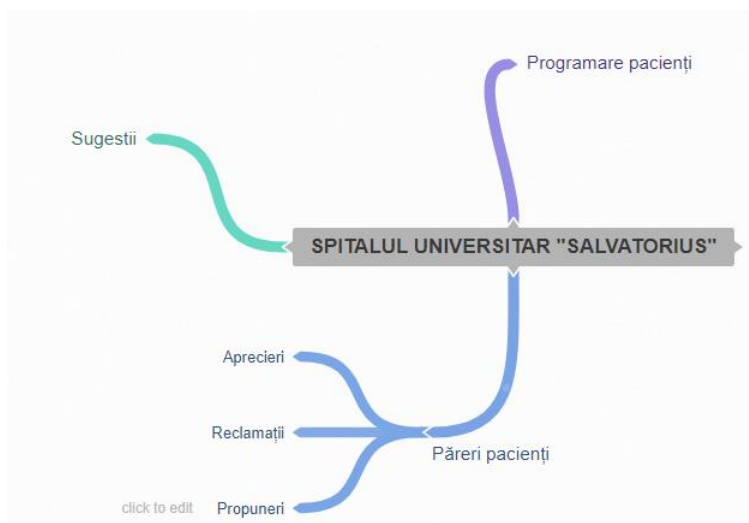
În acest exemplu, a fost creat un formular simplu cu un input de tip **"time"**. Eticheta **<label>** este utilizată pentru a afișa o etichetă descriptivă asociată cu inputul de tip **"time"**. Inputul de tip **"time"** are atributul **"name"** setat la **"time"**, ceea ce înseamnă că timpul selectat va fi trimis cu această denumire pe server.

După selectarea unei valori de timp, utilizatorul poate apăsa butonul **"Trimite"** pentru a trimite formularul împreună cu timpul selectat.

Este important de menționat că afișarea și suportul pentru inputul de tip **"time"** pot varia între browsere. Unele browsere pot utiliza un selector interactiv pentru ore și minute, în timp ce altele pot utiliza un câmp text în care utilizatorul trebuie să introducă manual timpul într-un anumit format.

**Temă de laborator:** Realizați în <https://coggle.it/> o hartă conceptuală în care prezentați minimum 8 utilități/ beneficii ale formularelor în cadrul unui site web al unei companii de produse IT. Pentru a crea o hartă conceptuală pe această platformă vă veți conecta cu contul de gmail. Harta conceptuală va fi partajată pe Classroom împreună cu fișierul HTML de prezentare a acesteia. În centrul hărții conceptuale va fi introdus numele companiei (în stabiliți voi). Se va folosi ca model, exemplul postat la adresa și se va utiliza un cod asemănător pentru a insera harta în pagina HTML creată de voi:

```
<iframe width='853' height='480'  
src='https://embed.coggle.it/diagram/ZKrX8JnX8RXLJRh_/950e7df84  
5007b734ce84aed6c751e6f5f4fb1914dbf8506cedf12ae8e082e75'  
frameborder='0' allowfullscreen></iframe>
```



## II.5. Sunete

Pentru a insera o melodie pe fundalul unei pagini web folosind cod HTML, se poate utiliza elementul **<audio>**.

Elementul **<audio>** este utilizat pentru a încorpora și reda fișiere audio într-o pagină web.

Atributul principal este **"src"**, care specifică locația și numele fișierului audio.

Alte atribute opționale pot fi utilizate pentru a controla redarea și stilizarea elementului audio.

Sintaxa:

```
<audio src="calea_catre_fisier" controls autoplay loop>  
Mesaj de înștiințare pentru browserele care nu suportă elementul  
audio.  
</audio>
```

În care **"calea\_catre\_fisier"** reprezintă calea către fișierul audio ce urmează a fi redat. Poate fi un URL absolut sau relativ către fișierul audio. Atributul **"controls"** afișează controalele de redare, cum ar fi butoanele de redare/pauză, controlul volumului și bara de progres.

Atributul **"autoplay"** face ca fișierul audio să se redă automat imediat ce pagina se încarcă.

Atributul **"loop"** permite repetarea continuă a fișierului audio.

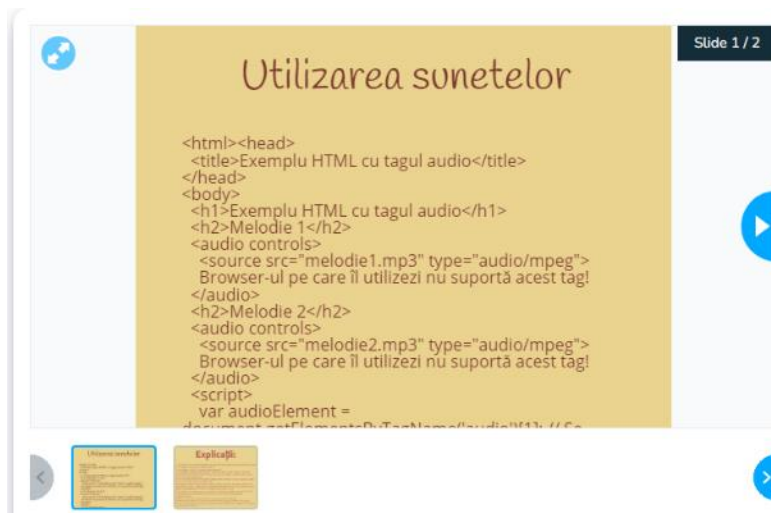
Exemplu:

```
<audio src="cantece/melodie.mp3" controls autoplay loop>  
Din păcate, browserul tău nu suportă elementul audio.  
</audio>
```

În exemplul de mai sus, fișierul audio **"melodie.mp3"** se află într-un director numit **"cantece"** în cadrul proiectului web. Elementul **<audio>** include atributele **"controls"**, **"autoplay"** și **"loop"** pentru a permite controlul redării, redarea automată și repetarea continuă a melodiei. În cazul în care browserul utilizatorului nu suportă elementul audio, se va afișa mesajul specificat între etichetele **<audio>**.

Trebuie să verificăm că fișierul audio specificat există în directorul corect și este compatibil cu formatele audio suportate de browserele utilizatorilor.

**Temă de laborator:** Realizați o pagină HTML care să redea pe rând două melodii. Cele două melodii vor fi descărcate de pe Youtube (melodiile trebuie să fie instrumentale cu o durată de 1-4 minute). În realizarea paginii HTML se va pleca de la exemplul prezentat și descris pe cele două diapozitive ale prezentării de la adresa: <https://nearpod.com/library/preview/lesson-L135994216>



## II.6. Clipuri video

Pentru a vizualiza un clip video într-o pagină web folosind cod HTML, se poate utiliza elementul **<video>**.

Elementul **<video>** este utilizat pentru a încorpora și reda fișiere video într-o pagină web. Atributul principal este **"src"**, care specifică locația și numele fișierului video. Alte atribute opționale pot fi utilizate pentru a controla redarea, dimensiunile și stilizarea elementului video.

Sintaxa:

**<video src="calea\_catre\_fisier" width="lățime" height="înălțime" controls autoplay loop>**

**Mesaj de înștiințare pentru browserele care nu suportă elementul video.**

**</video>**

În care **"calea\_catre\_fisier"** reprezintă calea către fișierul video ce urmează a fi redat. Poate fi un URL absolut sau relativ către fișierul video. Atributele **"width"** și **"height"** specifică dimensiunile în pixeli ale elementului video. Acestea sunt opționale și pot fi ajustate în funcție de necesități. Atributul **"controls"** afișează controalele de redare, cum ar fi butoanele de redare/pauză, controlul volumului și bara de progres. Atributul **"autoplay"** face ca fișierul video să fie redat automat imediat ce pagina se încarcă iar atributul **"loop"** permite repetarea continuă a fișierului video.

Exemplu:

```
<video src="video/clip.mp4" width="640" height="480" controls  
autoplay loop>
```

*Din păcate, browserul tău nu suportă elementul video.*

```
</video>
```

În acest exemplu, fișierul video numit **"clip.mp4"** se află într-un director cu numele **"video"** în cadrul proiectului web. Elementul **<video>** include atributele **"src"**, **"width"**, **"height"**, **"controls"**, **"autoplay"** și **"loop"** pentru a specifica sursa videoclipului, dimensiunile, controalele de redare, redarea automată și repetarea continuă. În cazul în care browserul utilizatorului nu suportă elementul video, se va afișa mesajul specificat între etichetele **<video>**.

Trebuie verificat dacă fișierul video specificat există în directorul corect și este compatibil cu formatele video suportate de browserele utilizatorilor tăi.

**Temă de laborator:** Realizați un fișier HTML care să fie redat un videoclip cu prezentarea unui tutorial de JavaScript. Tutorialul va fi preluat de pe Youtube / Google video. . În realizarea paginii HTML se va pleca de la exemplul prezentat și descris pe cele două diapozitive ale prezentării de la adresa: <https://nearpod.com/library/preview/lesson-L135994528>



## II.7. Hărți de imagini

Hărțile de imagini în HTML permit asocierea de regiuni specifice ale unei imagini cu link-uri sau acțiuni. Acestea sunt definite folosind tag-ul HTML **<map>** și tag-urile **<area>**. Un exemplu de utilizare:

```
<html>
```

```

<head>
  <title>Exemplu hartă de imagine</title>
</head>
<body>
  <h1>Hartă de imagine</h1>
  

  <map name="harta">
    <area shape="rect" coords="0,0,100,100" href="pagina1.html"
alt="Regiune 1">
    <area shape="circle" coords="150,150,50" href="pagina2.html"
alt="Regiune 2">
    <area shape="poly" coords="200,200,250,250,200,300,150,250"
href="pagina3.html" alt="Regiune 3">
  </map>
</body>
</html>

```

Se utilizează tag-ul **<img>** pentru a afișa imaginea de referință, *image.jpg*, în acest caz. Se utilizează atributul *usemap* al tag-ului **<img>** pentru a specifica numele hărții de imagini. În acest exemplu, numele hărții este *harta*. Se utilizează tag-ul **<map>** pentru a începe definirea hărții de imagini și se specifică numele hărții folosind atributul *name*.

Se utilizează tag-ul **<area>** pentru a defini regiunile hărții de imagini. Atributul *shape* specifică forma regiunii, iar atributul *coords* specifică coordonatele acesteia.

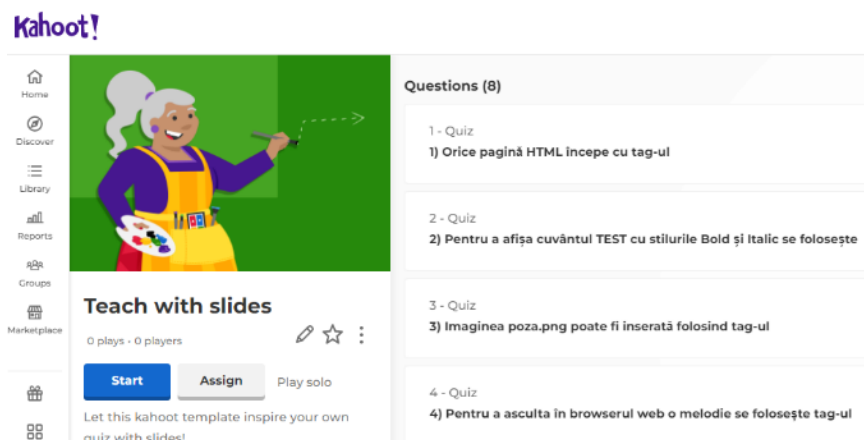
În exemplul de mai sus, am definit trei regiuni: o regiune dreptunghiulară cu colțul stângasus la coordonatele (0,0) și colțul dreapta-jos la coordonatele (100,100); o regiune circulară cu centrul la coordonatele (150,150) și rază de 50 de pixeli și o regiune poligonală definită prin perechi de coordonate (x, y).

Pentru fiecare regiune, s-a specificat prin atributul *href* legătura către pagina care se va naviga atunci când utilizatorul dă clic pe regiune. S-a folosit atributul *alt* pentru a oferi un text alternativ pentru accesibilitate.

Este important de menționat că, coordonatele *coords* sunt specificate în funcție de dimensiunile imaginii. În exemplul de mai sus, imaginea *image.jpg* trebuie să fie prezentă în același director cu fișierul HTML. Trebuie înlocuite *image.jpg*, *pagina1.html*, *pagina2.html*, *pagina3.html* și orice alte valori necesare pentru a se potrivi cu propriile fișiere și coordonate.

**Temă de laborator:** Conectați-vă cu contul de gmail și rezolvați următorul chestionar HTML:

<https://create.kahoot.it/share/teach-with-slides/8ae81e75-6080-49ae-aeefa-bca426bbf27c>



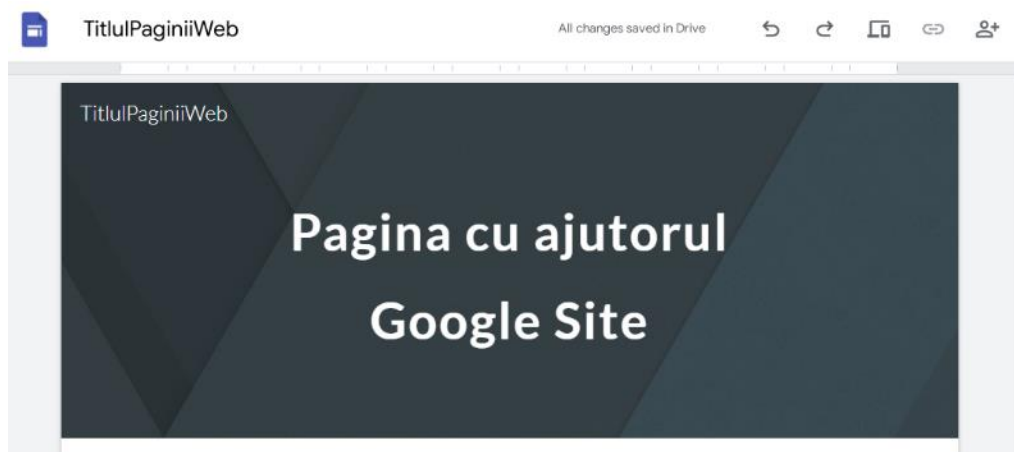
The screenshot shows the Kahoot! interface for a quiz titled "Teach with slides". On the left is a sidebar with navigation icons for Home, Discover, Library, Reports, Groups, and Marketplace. The main area features a green background with a cartoon character of an elderly woman in a purple shirt and yellow overalls, holding a paintbrush and pointing at a green square. Below the character, the title "Teach with slides" is displayed, along with "0 plays · 0 players" and icons for editing, favoriting, and sharing. There are three buttons: "Start" (blue), "Assign" (grey), and "Play solo" (text). A description reads: "Let this kahoot template inspire your own quiz with slides!". On the right, a "Questions (8)" panel lists four quiz questions:

- 1 - Quiz  
1) Orice pagină HTML începe cu tag-ul
- 2 - Quiz  
2) Pentru a afișa cuvântul TEST cu stilurile Bold și Italic se folosește
- 3 - Quiz  
3) Imaginea poza.png poate fi inserată folosind tag-ul
- 4 - Quiz  
4) Pentru a asculta în browserul web o melodie se folosește tag-ul

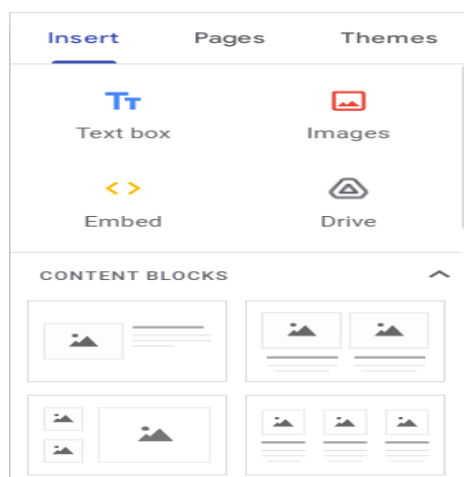
### III. Prelucrarea conținuturilor web

#### III.1. GoogleSites - Formatarea din interfața grafică a aplicației

Interfața aplicației online GoogleSites este formată din două secțiuni, cea din stânga în care se afișează conținutul paginii, ca în exemplul de mai jos



Și cea din dreapta care conține elementele ce pot fi introduse și formate, secțiune care la rândul ei este structurată în 3 grupuri.



Bara de instrumente din partea de sus a platformei:



permite anularea și refacerea; previzualizarea paginii; partajarea cu alte persoane; stabilirea setărilor specifice paginii și publicarea acesteia.

Din interfața aplicației pot fi inserate următoarele elemente (folosind opțiunea **Insert**): casetă text, imagine, obiecte încorporate (conținut multimedia, cum ar fi fișiere audio, video, animații sau alte aplicații), fișiere din Google Drive, blocuri predefinite cu diferite moduri de aranjare a textelor și

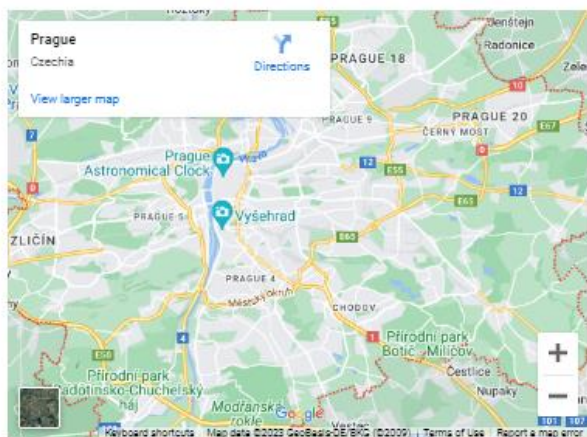


imaginilor, grup pliabil, cuprinsuri, carusel de imagini, buton, separator, spațiator, legături către rețele sociale, substituiete, calendare, hărți, documente și multe altele.

Prin intermediul opțiunii **Pages** pot fi create subpagini iar prin intermediul opțiunii Themes se alege una din temele prefedinite de pagini puse la dispoziție.

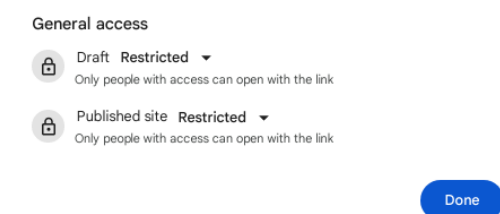
În doar câteva minute se poate obține o pagină asemănătoare cu:



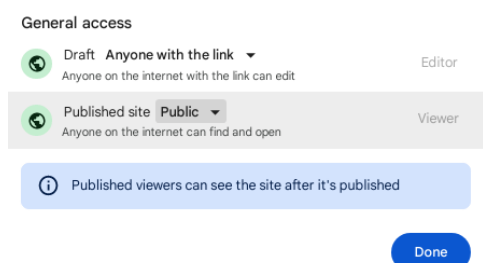


Când se dorește publicarea site-ului creat, se apasă pe butonul  și sunt oferite următoarele posibilități:

✓ cu acces restricționat



✓ cu acces public pentru oricine are link-ul către site



GoogleSites pune la dispoziție și o opțiune prin care unii utilizatori pot edita și face modificări sau completări ale site-ului (practic devin administratori ai acestuia). În acest caz se face partajare cu specificarea adreselor de e-mail ale celorlalți administratori și cu specificarea că utilizatorul este de tip **Editor**:

← Share "TitlulPaginiiWeb" ? ⚙

invitat1@gmail.com ✕

invitat2@gmail.com

Editor ▾

☒ Notify people

Message

Puteți realiza modificări ale acestui site după bunul plac.

Cancel Send

### III.2. Utilizarea foilor de stil

Foile de stil în limbajul HTML sunt utilizate pentru a controla aspectul și prezentarea elementelor unei pagini web. Acestea permit definirea proprietăților vizuale, cum ar fi culori, dimensiuni, stiluri de text, stiluri de fundal și multe altele. Utilizarea foilor de stil este utilă întrucât oferă un nivel ridicat de flexibilitate și control asupra aspectului paginilor web, permițând separarea clară între conținutul HTML și stilizarea acestuia. Aceasta duce la un cod mai organizat și mai ușor de întreținut.

Foile de stil pot fi incluse în documentul HTML în mai multe moduri:

- ✓ Stiluri în linie (inline styles)
- ✓ Stiluri în bloc (embedded styles)
- ✓ Fișiere separate de stil (external style sheets)

Avantajele utilizării foilor de stil în limbajul HTML includ:

- ✓ Separarea prezentării de conținut: Permite păstrarea structurii și semanticii corecte ale documentului HTML, în timp ce stilurile sunt separate și gestionate într-un fișier CSS separat. Aceasta face ca paginile să fie mai ușor de înțeles, de întreținut și de actualizat.
- ✓ Reutilizarea stilurilor: Stilurile definite într-un fișier CSS separat pot fi folosite pentru mai multe pagini, ceea ce reduce duplicarea codului și facilitează actualizările globale.
- ✓ Flexibilitate și consistență: Folosind stiluri, poți controla cu ușurință aspectul global al tuturor elementelor de același tip, asigurând o prezentare consecventă pe întreaga pagină web sau pe întregul site.
- ✓ Capacitatea de a gestiona aspectul într-un singur loc: Prin modificarea unui singur fișier CSS, se poate schimba aspectul întregului site, fără a fi nevoie să fie modificată fiecare pagină individual.

- ✓ Suport pentru media și responsivitate: Folosind media queries și alte tehnici CSS, se pot adapta stilurile la diferite dispozitive și dimensiuni de ecran, asigurând site-ul să facă afișarea corect.
- ✓ Extensibilitate și posibilități avansate de stilizare: CSS oferă o gamă largă de funcționalități și opțiuni de stilizare, cum ar fi animații, tranziții, efecte speciale și multe altele. Acestea permit personalizarea avansată a aspectului și crearea de interfețe web atractive și interactive.

### III.3. Foi de stil interne

Se pot utiliza foi de stil interne pentru a aplica stiluri specifice unei pagini web individuale. Aceste foi de stil sunt definite direct în cadrul elementului **<style>** din secțiunea **<head>** a documentului HTML.

Sintaxa pentru a crea o foaie de stil internă:

```
<html>
<head>
  <title>Exemplu de foaie de stil internă</title>
  <style>
    /* Reguli de stil */
  </style>
</head>
<body>
  <!-- Conținutul paginii -->
</body>
</html>
```

În cadrul secțiunii **<style>**, se pot scrie reguli de stil folosind sintaxa CSS obișnuită. Aceste reguli de stil vor fi aplicate elementelor din pagina HTML curentă. În continuare va fi analizat un exemplu detaliat pentru a înțelege modul de folosire a foilor de stil interne:

```
<html>
<head>
  <title>Exemplu de foaie de stil internă</title>
  <style>
    /* Reguli de stil */
    h1 {
      color: blue;
      font-size: 24px;
    }
    p {
      color: red;
    }
  </style>
</head>
<body>
  <!-- Conținutul paginii -->
</body>
</html>
```

```

    font-size: 16px;
  }
</style>
</head>
<body>
  <h1>Titlu</h1>
  <p>Acesta este un paragraf.</p>
</body>
</html>

```

În acest exemplu, s-au definit reguli de stil pentru selectorii **h1** și **p** în interiorul elementului **<style>**. Stilurile specificate sunt aplicate automat elementelor corespunzătoare în cadrul paginii. Astfel, titlul **<h1>** va avea culoarea albastră și dimensiunea de 24 de pixeli, în timp ce paragraful **<p>** va avea culoarea roșie și dimensiunea de 16 pixeli.

Foaia de stil internă este utilă în cazurile în care se dorește să se aplice stiluri specifice unei singure pagini, fără a afecta alte pagini web din același site. Totuși, pentru a aplica stiluri care trebuie partajate între mai multe pagini, este mai eficient să fie utilizate foi de stil externe (CSS extern).

### III.4. Foi de stil inline

În HTML, se pot utiliza stiluri "**inline**" pentru a aplica stiluri direct la nivelul unui element HTML individual. Aceste stiluri sunt specificate prin utilizarea atributului **style** pe elementul respectiv. Sintaxa pentru a folosi stiluri "**inline**":

```

<html>
<head>
  <title>Exemplu de stil inline</title>
</head>
<body>
  <h1 style="color: blue; font-size: 24px;">Titlu</h1>
  <p style="color: red; font-size: 16px;">Acesta este un paragraf.</p>
</body>
</html>

```

În exemplul de mai sus, s-a adăugat atributul **style** pe elementele **<h1>** și **<p>**. Valorile atributului **style** reprezintă regulile de stil CSS, separate prin punct și virgulă.

Această abordare permite specificarea stilurilor individuale pentru fiecare element. În exemplul de mai sus, titlul **<h1>** va avea culoarea albastră și dimensiunea de 24 de pixeli, în timp ce paragraful **<p>** va avea culoarea roșie și dimensiunea de 16 pixeli.

Stilurile "*inline*" sunt utile atunci când se dorește a se aplica stiluri specifice doar unui singur element și nu este nevoie să fie partajate cu alte elemente sau pagini web. Totuși, în general, este mai recomandat să fie utilizate foi de stil interne sau externe pentru a separa stilurile de conținutul HTML, deoarece oferă o gestionare mai eficientă și mai flexibilă a stilurilor.

### III.5. Foi de stil externe

Pentru a utiliza foi de stil externe în HTML, se poate folosi elementul `<link>` în secțiunea `<head>` a documentului HTML. Acest element `<link>` stabilește legătura dintre documentul HTML și fișierul de stil extern.

Sintaxa de bază a elementului `<link>` pentru atașarea unei foi de stil externe:

```
<link rel="stylesheet" type="text/css" href="calea/la/fisierul-de-stil.css">
```

Explicarea atributelor utilizate:

**rel:** Acesta specifică relația dintre documentul HTML și fișierul de stil extern. În cazul utilizării unei foi de stil, valoarea atributului rel trebuie să fie setată la "*stylesheet*".

**type:** Acest atribut specifică tipul de conținut al fișierului legat. În cazul foi de stil CSS, valoarea atributului type trebuie să fie setată la "*text/css*".

**href:** Acest atribut specifică calea către fișierul de stil extern. Aceasta poate fi o cale relativă sau absolută către fișierul CSS.

Va fi prezentat un exemplu detaliat de utilizare a unei foi de stil externe într-un document HTML: Se creează mai întâi un fișier HTML numit "*pagina.html*" și un fișier CSS numit "*stil.css*" în același director.

#### Exemplul 1:

Se deschide fișierul "*pagina.html*" într-un editor de text și se adaugă următorul cod:

```
<html>
<head>
  <title>Exemplu de folosire a foii de stil externe</title>
  <link rel="stylesheet" type="text/css" href="stil.css">
</head>
<body>
  <h1>Acesta este un exemplu</h1>
  <p>Acesta este un paragraf de text.</p>
</body>
</html>
```

Se deschide apoi fișierul "stil.css" în același editor de text și adăugați următorul cod:



```

h1 {
    color: blue;
}
p {
    font-size: 18px;
}

```

În acest exemplu, fișierul HTML (***pagina.html***) include fișierul CSS (***stil.css***) utilizând elementul ***<link>***. Conținutul CSS din fișierul "***stil.css***" va fi aplicat elementelor HTML corespunzătoare din fișierul "***pagina.html***". Astfel, ***<h1>*** va avea culoarea albastră, iar ***<p>*** va avea dimensiunea de 18px.

Acum poate fi deschis fișierul HTML într-un browser web și se va vedea că stilurile CSS sunt aplicate corespunzător. Acesta este un exemplu simplu de utilizare a unei foi de stil externe în HTML.

Utilizarea acestor stiluri poate duce la caracteristici ale tag-urilor din cadrul fișierelor HTML mult mai detaliate. Vom analiza un al doilea exemplu pentru acest lucru.

### **Exemplul 2:**

Se va crea un fișier HTML numit "***pagina.html***" și un fișier CSS numit "***stil.css***" în același director. Se deschide apoi fișierul "***pagina.html***" într-un editor de text și se adaugă următorul cod:

```

<html>
<head>
    <title>Exemplu de folosire a foilor de stil externe</title>
    <link rel="stylesheet" type="text/css" href="stil.css">
    <link rel="stylesheet" type="text/css" href="alte-stiluri.css">
</head>
<body>
    <header>
        <h1>Titlul paginii</h1>
        <nav>
            <ul>
                <li><a href="#">Acasă</a></li>
                <li><a href="#">Despre</a></li>
                <li><a href="#">Servicii</a></li>
                <li><a href="#">Contact</a></li>
            </ul>
        </nav>
    </header>

```

```

<section>
  <h2>Despre noi</h2>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed feugiat, leo id aliquet tincidunt, est nisi tristique lectus, ut facilisis purus urna in mi.</p>
</section>

<section>
  <h2>Servicii</h2>
  <ul>
    <li>Web design</li>
    <li>Dezvoltare web</li>
    <li>Gazduire</li>
  </ul>
</section>

<footer>
  <p>&copy; 2023 Numele Companiei. Toate drepturile rezervate.</p>
</footer>
</body>
</html>

```

Se deschide apoi fișierul "**stil.css**" cu un editor de text (poate fi același cu cel utilizat la crearea fișierului **pagina.html**) și se adaugă următoarea secvență de cod:

```

body {
  font-family: Arial, sans-serif;
}

header {
  background-color: #333;
  color: #fff;
  padding: 20px;
}

nav ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
}

```



```

nav ul li {
    display: inline-block;
    margin-right: 10px;
}

nav ul li a {
    color: #fff;
    text-decoration: none;
}

section {
    margin-bottom: 20px;
}

footer {
    background-color: #333;
    color: #fff;
    padding: 10px;
    text-align: center;
}

```

Se va crea un fișier CSS suplimentar numit "**alte-stiluri.css**" în același director; se va deschide în editorul de text și se va adăuga următorul cod:

```

header h1 {
    font-size: 24px;
}

section h2 {
    color: #333;
    border-bottom: 1px solid #333;
}

section p {
    line-height: 1.5;
}

ul {
    padding-left: 20px;
}

```

În acest exemplu, fișierul HTML (***pagina.html***) include două foi de stil externe (***stil.css*** și ***alte-stiluri.css***) utilizând elementul **`<link>`**. Prima foaie de stil (***stil.css***) definește stilurile de bază pentru elemente precum corpul paginii, antetul, secțiunile și subsolul. A doua foaie de stil (***alte-stiluri.css***) conține stiluri suplimentare pentru anumite elemente, cum ar fi dimensiunea fontului pentru titlul antetului, stilurile pentru secțiuni și stilurile pentru liste.

Atunci când va fi deschis fișierul HTML într-un browser web, se va observa că stilurile CSS sunt aplicate corespunzător. Fonturile, culorile, dimensiunile și alte stiluri definite în foile de stil externe vor fi aplicate elementelor HTML corespunzătoare.

Acest exemplu ilustrează modul în care se pot utiliza mai multe foi de stil externe pentru a gestiona stilurile pentru diferite componente și secțiuni ale unei pagini web.

**Temă de laborator:** Intrați pe padlet-ul partajat la adresa

[https://padlet.com/viorelinfo/tehnologii-web-stiluri\\_html-5ms9yw7klsnmf670](https://padlet.com/viorelinfo/tehnologii-web-stiluri_html-5ms9yw7klsnmf670) și creați o singură postare (fiecare dintre voi) în care prezentați un caz practic de utilizare a stilurilor HTML.

Viorel Andrișae • mai puțin de un minut

### Tehnologii web-Stiluri\_HTML

#### Foi de stil externe

Pentru a utiliza foi de stil externe în HTML, se poate folosi elementul **`<link>`** în secțiunea **`<head>`** a documentului HTML. Acest element **`<link>`** stabilește legătura dintre documentul HTML și fișierul de stil extern. Sintaxa de bază a elementului **`<link>`** pentru atașarea unei foi de stil externe:

```
<link rel="stylesheet" type="text/css" href="calea/la/fisierul-de-stil.css">
```

Atributelor utilizate:

- rel:** Acesta specifică relația dintre documentul HTML și fișierul de stil extern. În cazul utilizării unei foi de stil, valoarea atributului **rel** trebuie să fie setată la **"stylesheet"**.
- type:** Acest atribut specifică tipul de conținut al fișierului legat. În cazul foilor de stil CSS, valoarea atributului trebuie să fie **"text/css"**.

#### Foi de stil inline

În HTML, se pot utiliza stiluri **"inline"** pentru a aplica stiluri direct la nivelul unui element HTML individual. Aceste stiluri sunt specificate prin utilizarea atributului **style** pe elementul respectiv. Sintaxa pentru a folosi stiluri **"inline"**:

```
<html>
<head>
<title>Exemplu de stil inline</title>
</head>
<body>
<h1 style="color: blue; font-size: 24px;">Titlu</h1>
<p style="color: red; font-size: 16px;">Acesta este un paragraf.
</p>
</body>
</html>
```

#### Foi de stil interne

Se pot utiliza foi de stil interne pentru a aplica stiluri specifice unei pagini web individuale. Aceste foi de stil sunt definite direct în cadrul elementului **`<style>`** din secțiunea **`<head>`** a documentului HTML. Sintaxa pentru a crea o foaie de stil internă:

```
<html>
<head>
<title>Exemplu de foaie de stil internă</title>
<style>
/* Reguli de stil */
</style>
</head>
<body>
<!-- Conținutul paginii -->
</body>
</html>
```

#### Avantajele utilizării foilor de stil în limbajul HTML:

- Separarea prezentării de conținut: Permite păstrarea structurii și semanticii corecte ale documentului HTML, în timp ce stilurile sunt separate și gestionate într-un fișier CSS separat. Aceasta face ca paginile să fie mai ușor de înțeles, de întreținut și de actualizat.
- Reutilizarea stilurilor: Stilurile definite într-un fișier CSS separat pot fi folosite pentru mai multe pagini, ceea ce reduce duplicarea codului și facilitează actualizările globale.
- Flexibilitate și consistență: Folosind stiluri, poți controla cu ușurință aspectul global al tuturor elementelor de același tip, asigurând o prezentare consecventă pe întreaga pagină web sau pe întregul site.
- Capacitatea de a gestiona

#### Foi de stil

Foile de stil în limbajul HTML sunt utilizate pentru a controla aspectul și prezentarea elementelor unei pagini web. Acestea permit definirea proprietăților vizuale, cum ar fi culori, dimensiuni, stiluri de text, stiluri de fundal și multe altele. Utilizarea foilor de stil este utilă întrucât oferă un nivel ridicat de flexibilitate și control asupra aspectului paginilor web, permițând separarea clară între conținutul HTML și stilizarea acestuia. Aceasta duce la un cod mai organizat și mai ușor de întreținut.

## IV. Structurarea site-ului

### IV.1. Organizarea sub formă de pagină unică

Vom analiza un exemplu de pagină HTML complexă în care tot codul, inclusiv partea de stilizare, se află într-un singur fișier. Vom folosi stilurile CSS inline pentru a defini stilurile direct în codul HTML. Această metodă poate fi utilă pentru pagini mici sau exemple simple, dar pentru proiecte mai mari și mai complexe, se recomandă utilizarea fișierelor de stil externe.

```
<html>
<head>
  <title>Pagina HTML Complexă</title>
</head>
<body>
  <header style="background-color: #333; color: #fff; padding:
20px;">
    <h1 style="font-size: 24px;">Titlul paginii</h1>
    <nav>
      <ul style="list-style-type: none; margin: 0; padding: 0;">
        <li style="display: inline-block; margin-right: 10px;">
          <a href="#" style="color: #fff; text-decoration:
none;">Acasă</a>
        </li>
        <li style="display: inline-block; margin-right: 10px;">
          <a href="#" style="color: #fff; text-decoration:
none;">Despre</a>
        </li>
        <li style="display: inline-block; margin-right: 10px;">
          <a href="#" style="color: #fff; text-decoration:
none;">Servicii</a>
        </li>
        <li style="display: inline-block; margin-right: 10px;">
          <a href="#" style="color: #fff; text-decoration:
none;">Contact</a>
        </li>
      </ul>
    </nav>
  </header>

  <section style="margin-bottom: 20px;">
    <h2 style="color: #333; border-bottom: 1px solid #333;">Despre
noi</h2>
```

```
<p style="line-height: 1.5;">Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Sed feugiat, leo id aliquet tincidunt, est
nisi tristique lectus, ut facilisis purus urna in mi.</p>
</section>
```

```
<section style="margin-bottom: 20px;">
  <h2 style="color: #333; border-bottom: 1px solid
#333;">Servicii</h2>
  <ul style="padding-left: 20px;">
    <li>Web design</li>
    <li>Dezvoltare web</li>
    <li>Gazduire</li>
  </ul>
</section>
```

```
<table style="border-collapse: collapse; width: 100%;">
  <thead>
    <tr>
      <th style="border: 1px solid #333; padding:
10px;">Nume</th>
      <th style="border: 1px solid #333; padding:
10px;">Prenume</th>
      <th style="border: 1px solid #333; padding:
10px;">Vârsta</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td style="border: 1px solid #333; padding:
10px;">Popescu</td>
      <td style="border: 1px solid #333; padding: 10px;">Ion</td>
      <td style="border: 1px solid #333; padding: 10px;">30</td>
    </tr>
    <tr>
      <td style="border: 1px solid #333; padding:
10px;">Ionescu</td>
      <td style="border: 1px solid #333; padding:
10px;">Maria</td>
      <td style="border: 1px solid #333; padding: 10px;">25</td>
    </tr>
  </tbody>
</table>
```

```
<button style="background-color: #4CAF50; color: #fff; padding: 10px 20px; border: none; border-radius: 4px; cursor: pointer;">Apasă-mă</button>
```

```
<footer style="background-color: #333; color: #fff; padding: 10px; text-align: center;">
  <p>&copy; 2023 Numele Companiei. Toate drepturile rezervate.</p>
</footer>
</body>
</html>
```

Codul de mai sus creează următoarea pagină:

**Titlul paginii**
Acasa Despre Servicii Contact

## Despre noi

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed feugiat, leo id aliquet tincidunt, est nisi tristique lectus, ut facilisis purus urna in mi.

## Servicii

- Web design
- Dezvoltare web
- Gazduire

| Nume    | Prenume | Vârsta |
|---------|---------|--------|
| Popescu | Ion     | 30     |
| Ionescu | Maria   | 25     |

Apasa-ma

© 2023 Numele Companiei. Toate drepturile rezervate.

În acest exemplu, toate stilurile sunt specificate *inline*, folosind atributul *style* al fiecărui element HTML. Acest atribut conține perechi cheie-valoare care definesc stilurile pentru fiecare element în parte. Iată o scurtă explicație a stilurilor utilizate în codul HTML:

Stilurile pentru antet (*<header>*):

*background-color, color, padding* - specifică culoarea fundalului, culoarea textului și spațierea interioară.

Stilurile pentru meniu (*<nav>* și *<ul>*):

*list-style-type, margin, padding* - elimină stilul implicit al listei, elimină marginile și padding-ul.

Stilurile pentru secțiuni (*<section>*):

*margin-bottom* - specifică o margine inferioară între secțiuni.

*color, border-bottom* - specifică culoarea textului și o linie subtitlu.

Stilurile pentru tabel (*<table>*, *<th>*, *<td>*):

***border-collapse*** - specifică modul de colapsare a chenarelor celulelor tabelului.

***width*** - specifică lăţimea tabelului.

***border, padding*** - specifică stilul şi spaţiarea celulelor.

Stilurile pentru buton (***<button>***):

***background-color, color, padding, border, border-radius, cursor*** - specifică culoarea fundalului, culoarea textului, spaţiarea internă, stilul chenarului, raza de curbură a chenarului şi cursorul la interacţiunea cu butonul.

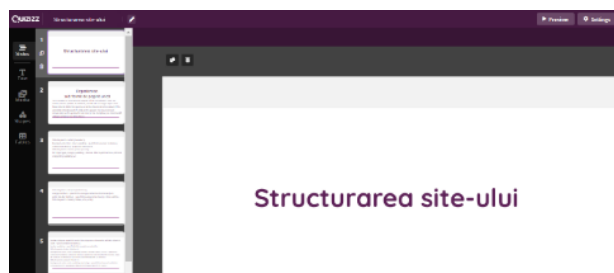
Stilurile pentru subsol (***<footer>***):

***background-color, color, padding, text-align*** - specifică culoarea fundalului, culoarea textului, spaţiarea internă şi alinierea textului în centru.

Este important de menţionat că utilizarea stilurilor ***inline*** poate deveni dificilă de gestionat pe măsură ce pagina devine mai complexă şi mai mare. În astfel de cazuri, este mai recomandat să fie utilizate foi de stil externe separate pentru a organiza şi gestiona mai eficient stilurile.

**Temă de laborator:** Urmăriţi materialul de la adresa

[https://quizizz.com/admin/presentation/64aec865942063001db76a59?source=lesson\\_share](https://quizizz.com/admin/presentation/64aec865942063001db76a59?source=lesson_share) şi realizaţi o structură de site (pentru pagina voastră personală care să conţină 3 subpagini).



## IV.2. Organizarea în mai multe pagini

Vom analiza structura unui site format din mai multe fişiere. Acest site este format din trei pagini HTML: "***acasa.html***", "***servicii.html***" şi "***contact.html***". Vom utiliza foi de stil externe pentru a separa codul HTML de stilizare în fişiere separate.

Mai întâi trebuie create cele trei fişiere HTML separate: "***acasa.html***", "***servicii.html***" şi "***contact.html***" în acelaşi director.

În fișierul "**acasa.html**", se va adăuga următorul cod:

```
<html>
<head>
  <title>Acasă</title>
  <link rel="stylesheet" type="text/css" href="stil.css">
</head>
<body>
  <h1>Bun venit pe pagina Acasă!</h1>
  <p>Aici veți găsi informații despre compania noastră și serviciile
pe care le oferim.</p>
  <a href="servicii.html">Vezi Servicii</a>
  <a href="contact.html">Contactează-ne</a>
</body>
</html>
```

În fișierul "**servicii.html**", adăugați următorul cod:

```
<html>
<head>
  <title>Servicii</title>
  <link rel="stylesheet" type="text/css" href="stil.css">
</head>
<body>
  <h1>Serviciile noastre</h1>
  <ul>
    <li>Web design</li>
    <li>Dezvoltare web</li>
    <li>Gazduire</li>
  </ul>
  <a href="acasa.html">Înapoi la Acasă</a>
  <a href="contact.html">Contactează-ne</a>
</body>
</html>
```

În fișierul "**contact.html**", va fi inserat următorul cod:

```
<html>
<head>
  <title>Contact</title>
  <link rel="stylesheet" type="text/css" href="stil.css">
</head>
<body>
  <h1>Contactați-ne</h1>
```

**<p>Puteți să ne contactați folosind următoarele informații de contact:</p>**

**<ul>**

**<li>Email: [contact@companie.com](mailto:contact@companie.com)</li>**

**<li>Telefon: 1234567890</li>**

**<li>Adresă: Strada Exemplu, Orașul Tau, 12345</li>**

**</ul>**

**<a href="acasa.html">Înapoi la Acasă</a>**

**<a href="servicii.html">Vezi Servicii</a>**

**</body>**

**</html>**

Se va crea apoi un fișier CSS numit "**stil.css**" în același director, se va deschide acest fișier în editorul de text și se vor adăuga stilurile CSS pentru site:

```
body {  
    font-family: Arial, sans-serif;  
    background-color: #f2f2f2;  
    margin: 0;  
    padding: 0;  
}
```

```
h1 {  
    color: #333;  
}
```

```
p {  
    color: #555;  
}
```

```
a {  
    color: #007bff;  
    text-decoration: none;  
}
```

```
ul {  
    list-style-type: none;  
    padding: 0;  
}
```

```
li {
```



```
margin-bottom: 10px;  
}
```

În acest exemplu, fiecare fișier HTML ("**acasa.html**", "**servicii.html**" și "**contact.html**") conține codul HTML corespunzător fiecărei pagini. Toate cele trei fișiere HTML includ aceeași foaie de stil externă "**stil.css**" folosind elementul **<link>**. Această foaie de stil definește stilurile comune pentru toate paginile, cum ar fi fontul, culorile și stilurile legăturilor și listelor.

Când se deschide oricare dintre fișierele HTML într-un browser web, se va observa că stilurile CSS din "**stil.css**" sunt aplicate corespunzător.

Utilizarea foilor de stil externe separate în acest mod oferă o abordare modulară și ușor de gestionat pentru stilizarea paginilor individuale. Modificarea stilurilor într-un singur fișier "**stil.css**" va afecta toate paginile care îl utilizează, facilitând actualizarea și menținerea stilurilor pe întregul site.

### IV.3. Organizarea în mai multe foldere

În continuare vom analiza un exemplu de structură de site formată din trei pagini HTML: "**acasa.html**", "**servicii.html**" și "**contact.html**", în care codul și resursele sunt organizate în foldere separate. Structura folderelor arată astfel:

```
site/  
├── imagini/  
│   └── logo.png  
├── stil/  
│   └── stil.css  
├── acasa/  
│   └── acasa.html  
├── servicii/  
│   └── servicii.html  
└── contact/  
    └── contact.html
```

În folderul "**site/imagini/**", se va adăuga o imagine numită "**logo.png**".

În folderul "**site/stil/**", se va salva un fișier CSS numit "**stil.css**", apoi acesta va fi deschis în editorul de text. Se completează (se adaugă) stilurile CSS pentru site:

```
body {  
    font-family: Arial, sans-serif;  
    background-color: #f2f2f2;  
    margin: 0;  
    padding: 0;  
}
```

```
h1 {  
    color: #333;  
}
```

```
p {  
    color: #555;  
}
```

```
a {  
    color: #007bff;  
    text-decoration: none;  
}
```

```
ul {  
    list-style-type: none;  
    padding: 0;  
}
```

```
li {  
    margin-bottom: 10px;  
}
```

În folderul "**site/acasa/**", se crează fișierul "**acasa.html**" și se adaugă următorul cod:

```
<html>  
<head>  
    <title>Acasă</title>  
    <link rel="stylesheet" type="text/css" href="../stil/stil.css">  
</head>  
<body>  
    <h1>Bun venit pe pagina Acasă!</h1>  
    <p>Aici veți găsi informații despre compania noastră și serviciile  
    pe care le oferim.</p>
```

```

    <a href="../servicii/servicii.html">Vezi Servicii</a>
    <a href="../contact/contact.html">Contactează-ne</a>
    
</body>
</html>

```

În folderul "**site/servicii**", va fi creat fișierul "**servicii.html**" și se va adăuga următorul cod:

```

<html>
<head>
    <title>Servicii</title>
    <link rel="stylesheet" type="text/css" href="../stil/stil.css">
</head>
<body>
    <h1>Serviciile noastre</h1>
    <ul>
        <li>Web design</li>
        <li>Dezvoltare web</li>
        <li>Gazduire</li>
    </ul>
    <a href="../acasa/acasa.html">Înapoi la Acasă</a>
    <a href="../contact/contact.html">Contactează-ne</a>
    
</body>
</html>

```

În folderul "**site/contact**", va fi creat fișierul "**contact.html**" în care se va insera următorul cod:

```

<html>
<head>
    <title>Contact</title>
    <link rel="stylesheet" type="text/css" href="../stil/stil.css">
</head>
<body>
    <h1>Contactați-ne</h1>
    <p>Puteți să ne contactați folosind următoarele informații de contact:</p>
    <ul>
        <li>Email: contact@companie.com</li>
        <li>Telefon: 1234567890</li>
    </ul>

```

```

        <li>Adresă: Strada Exemplu, Orașul Tau, 12345</li>
    </ul>
    <a href="../acasa/acasa.html">Înapoi la Acasă</a>
    <a href="../servicii/servicii.html">Vezi Servicii</a>
    
</body>
</html>

```

În acest exemplu, fișierele HTML ("**acasa.html**", "**servicii.html**" și "**contact.html**") se află în directoare separate corespunzătoare paginilor. Fișierele CSS și imagini sunt plasate în directoare separate pentru stiluri și imagini.

Fiecare fișier HTML include calea corectă către foaia de stil externă folosind elementul **<link>** și atributul **href**. Calea este relativă la locația fișierului HTML curent. De exemplu, în "**acasa.html**", calea "**../stil/stil.css**" indică faptul că fișierul "**stil.css**" se află în directorul părinte ("**../**") al directorului "**acasa/**". Același principiu se aplică și altor resurse, cum ar fi imaginile.

Când se deschide oricare dintre fișierele HTML într-un browser web, se va observa că stilurile CSS și imaginile sunt încărcate și afișate corect, datorită utilizării căilor corecte către aceste resurse.

Utilizarea unei astfel de structuri organizate în foldere separate facilitează gestionarea și menținerea resurselor site-ului într-un mod ordonat și clar.

#### IV.4. Harta site-ului

Vom analiza un exemplu de structură a site-ului cu cele patru pagini HTML (*Sediu*, *Agentie*, *Magazin*, *Prezentare*) și fișiere separate pentru codul HTML și stilizare.

Structura fișierelor:

```

index.html (pagina principală)
sediu.html
agentie.html
magazin.html

```

*prezentare.html*

*style.css* (fișierul de stilizare CSS)

Codul din fișierul ***index.html***:

```
<html>
<head>
  <title>Site-ul meu</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <h1>Bine ați venit pe site-ul meu!</h1>
  <nav>
    <ul>
      <li><a href="sediu.html">Sediu</a></li>
      <li><a href="agentie.html">Agentie</a></li>
      <li><a href="magazin.html">Magazin</a></li>
      <li><a href="prezentare.html">Prezentare</a></li>
    </ul>
  </nav>
</body>
</html>
```

Codul din fișierul ***sediu.html*** (și același format pentru celelalte pagini):

```
<html>
<head>
  <title>Sediul nostru</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <h1>Sediul nostru</h1>
  <p>Acesta este sediul nostru, situat într-un loc minunat.</p>
  <a href="index.html">Înapoi la pagina principală</a>
</body>
</html>
```

Codul din fișierul ***style.css***:

```
/* Stilizare pentru întregul site */

body {
  font-family: Arial, sans-serif;
```

```

margin: 0;
padding: 0;
}

h1 {
color: #333;
}

nav ul {
list-style-type: none;
margin: 0;
padding: 0;
}

nav ul li {
display: inline;
margin-right: 10px;
}

nav ul li a {
text-decoration: none;
color: #333;
}

/* Stilizare specifică pentru fiecare pagină */
/* Reguli CSS specifice pentru fiecare pagină pot fi adăugate aici */

```

Harta site-ului (sitemap) reprezintă o structură ierarhică a paginilor de pe un site. Aceasta facilitează navigarea și înțelegerea relațiilor dintre pagini. În cazul acestui exemplu, harta site-ului arată astfel:

Pagina principală (***index.html***)

Sediu (***sediu.html***)

Agentie (***agentie.html***)

Magazin (***magazin.html***)

Prezentare (***prezentare.html***)

Harta site-ului indică faptul că pagina principală este punctul de intrare în site și că există patru pagini accesibile de la pagina principală. Utilizatorii pot accesa oricare dintre aceste pagini făcând clic pe link-urile corespunzătoare din meniul de navigare (nav). Aceasta oferă o structură clară și ușor de urmărit a site-ului, făcând mai ușoară explorarea și găsirea informațiilor dorite.

## V. Verificări înainte de publicare

### IV.1. Analiza rezultatului obținut

Înainte de a încărca paginile unui site pe un server, este important să fie verificate mai multe aspecte pentru a ne asigura că conținutul este pregătit și funcțional. Trebuie avut în vedere câteva aspecte cheie care ar trebui verificate:

- ✓ *Corectitudinea codului HTML*: se verifică dacă toate paginile HTML sunt valide din punct de vedere sintactic. Se pot utiliza instrumente precum W3C Markup Validation Service pentru a verifica dacă nu există erori sau avertismente în codul HTML.
- ✓ *Stilizarea și aspectul vizual*: se verifică dacă stilurile CSS sunt aplicate corect și dacă pagina arată așa cum a fost proiectată inițial. Se verifică culorile, fonturile, dimensiunile și alți parametri de stilizare pentru a ne asigura că paginile arată coerent și profesionist.
- ✓ *Responsivitate și adaptabilitate la diferite dispozitive*: se verifică cum arată și funcționează site-ul pe diferite dispozitive și dimensiuni de ecran, cum ar fi desktop, tabletă și telefon mobil. Se verifică dacă design-ul este responsiv și dacă elementele se redimensionează și se plasează corespunzător în funcție de ecran.
- ✓ *Link-urile și navigarea*: se verifică dacă toate link-urile interne și externe din pagină funcționează corect și îndreaptă utilizatorii către destinația potrivită. Trebuie să ne asigurăm că meniurile și hiperlink-urile de navigare sunt configurate corespunzător și că toate paginile sunt accesibile.
- ✓ *Imaginile și multimedia*: se verifică dacă toate imaginile și elementele multimedia sunt încărcate corect și afișate în mod corespunzător. Se verifică dimensiunile, calitatea și formatul fișierelor pentru a asigura o experiență vizuală plăcută pentru utilizatori.
- ✓ *Formulare și funcționalitate interactivă*: se verifică dacă site-ul conține formulare sau alte elemente interactive, se verifică dacă acestea funcționează corect și dacă datele introduse de utilizatori sunt procesate în mod adecvat. Se testează funcționalitatea butoanelor, validarea datelor și funcțiile de trimitere a formularului.
- ✓ *Optimizarea performanței*: trebuie să ne asigurăm că paginile sunt optimizate pentru încărcare rapidă. Se verifică dimensiunea fișierelor, se minimizează și comprimă codul HTML, CSS și JavaScript pentru a reduce timpul de încărcare și pentru a îmbunătăți performanța site-ului.
- ✓ *Verificare a compatibilității cu diferite browsere*: se testează site-ul pe diferite browsere populare, cum ar fi Google Chrome, Mozilla Firefox, Safari, Microsoft Edge și altele, pentru a ne asigura că afișarea și funcționalitatea rămân consistente pe diferite platforme.



- ✓ *Verificare a conținutului și a gramaticii:* se revizuieste conținutul paginilor pentru a ne asigura că nu există erori gramaticale sau de tastare. Se verifică dacă informațiile sunt actualizate și corecte.
- ✓ *Testarea generală a site-ului:* se efectuează o testare generală a întregului site pentru a verifica toate funcționalitățile și pentru a ne asigura că nu există erori sau probleme neașteptate. Se explorează fiecare pagină și se verifică interacțiunile și funcțiile specifice.

Verificarea tuturor acestor aspecte înainte de a încărca paginile pe server ne va ajuta să ne asigurăm că site-ul este funcțional, coerent și pregătit pentru a fi accesat de către utilizatori. Este recomandat să fie efectuate teste în mai multe etape și pe mai multe dispozitive pentru a identifica și remedia eventuale probleme înainte de lansarea oficială a site-ului.

## VI. Publicarea

Pentru a publica un site și a încărca fișierele pe un server, vor fi urmați acești pași de bază:

- ✓ Va fi achiziționat un nume de domeniu și un serviciu de găzduire web: Un nume de domeniu reprezintă adresa pe care oamenii o vor utiliza pentru a accesa site-ul, iar serviciul de găzduire web oferă spațiu pe un server pentru a stoca fișierele site-ului.
- ✓ Configurare a DNS-ului: Dacă numele de domeniu și serviciul de găzduire web sunt achiziționate separat, trebuie configurate setările DNS pentru a redirecționa traficul către serverul de găzduire. Aceasta implică adăugarea unor înregistrări DNS care conectează numele de domeniu cu adresa IP a serverului de găzduire web.
- ✓ Trebuie create / pregătite fișierele site-ului. Este necesar să fie pregătite fișierele HTML, CSS, JavaScript și alte resurse necesare pentru site. Trebuie verificat că toate fișierele sunt organizate într-o structură de directoare logică.
- ✓ Se stabilește conexiunea cu serverul de găzduire web utilizând un Client FTP: pentru a încărca fișierele pe server, este nevoie de un client FTP (File Transfer Protocol) pentru a stabili o conexiune cu serverul. Există numeroase aplicații de tip client FTP disponibile. Iată trei exemple: a.) *FileZilla*: FileZilla este un client FTP gratuit și popular, disponibil pentru Windows, macOS și Linux. Acesta oferă o interfață intuitivă, suport pentru transferuri sigure (SFTP) și funcții avansate de gestionare a fișierelor. b.) *Cyberduck*: Cyberduck este un client FTP gratuit și open-source disponibil pentru Windows și macOS. Suportă o varietate de protocoale de transfer de fișiere, inclusiv FTP, SFTP și WebDAV, și oferă o interfață prietenoasă. c.) *WinSCP*: WinSCP este un client FTP gratuit pentru Windows, care permite transferuri de fișiere sigure prin protocoalele FTP, SFTP și SCP. Are o interfață duală care afișează atât directorul local, cât și cel remote, pentru o gestionare ușoară a fișierelor.
- ✓ Trebuie configurată conexiunea FTP: în clientul FTP, se introduce adresa serverului, numele utilizatorului și parola furnizate de serviciul de găzduire web. Aceste informații ar trebui să fie disponibile în contul de găzduire web sau în e-mail-ul de bun venit primit de la furnizorul de servicii.
- ✓ Se încarcă fișierele pe server: se va utiliza clientul FTP pentru a naviga prin directoarele de pe server și a încărca fișierele site-ului. De obicei, există o zonă desemnată pentru fișierele site-ului (de exemplu, un director "**public\_html**" sau "**www**") în care ar trebui să fie copiate fișierele.
- ✓ Se verifică apoi publicarea site-ului: după ce fișierele sunt încărcate pe server, se deschide un browser web și se accesează numele de domeniu al site-ului pentru a verifica dacă acesta

a fost publicat corect. Trebuie verificat dacă toate paginile și resursele sunt afișate așa cum s-a stabilit înainte.

Acești pași vor acoperi procesul de bază pentru publicarea unui site și încărcarea fișierelor pe un server. Este posibil ca anumite detalii și setări să difere în funcție de furnizorul de servicii de găzduire web ales și de aplicația client FTP folosită.

Editarea fișierelor direct pe serverul web poate fi realizată folosind un editor de texte integrat în interfața serverului sau prin utilizarea unui client FTP care suportă funcționalitatea de editare a fișierelor pe server.

Printre cele mai utilizate două modalități comune de a edita fișierele unui site direct pe serverul web se enumeră:

- ✓ *Editorul de texte integrat în interfața serverului web:* Unele panouri de control ale serverelor web oferă opțiuni de editare a fișierelor direct în interfața lor. Aceasta poate fi o opțiune disponibilă în panourile de control populare, cum ar fi **cPanel** sau **Plesk**. De obicei, aceste panouri oferă un editor de texte simplu în care putem deschide și edita fișierele direct pe server. Se accesează panoul de control al serverului și se caută opțiunea de editare a fișierelor pentru a începe editarea direct pe serverul web.
- ✓ *Utilizarea unui client FTP cu suport pentru editare:* unele aplicații de tip client FTP oferă funcționalitate de editare a fișierelor direct pe serverul web. Iată câteva exemple de astfel de aplicații:
  - a.) *Notepad++:* este un editor de texte gratuit și open-source pentru Windows. Poate fi configurat pentru a se conecta la serverul web folosind protocolul FTP și oferă opțiuni de editare a fișierelor direct pe server.
  - b.) *Sublime Text:* este un editor de texte popular și puternic disponibil pentru Windows, macOS și Linux. Poate fi utilizat împreună cu un plugin, cum ar fi SFTP, pentru a se conecta la serverul web și a permite editarea directă a fișierelor.
  - c.) *Atom:* este un editor de texte gratuit și extensibil dezvoltat de GitHub. Cu ajutorul unui plugin precum Remote FTP, se poate configura Atom pentru a se conecta la serverul web și a permite editarea fișierelor direct pe server.

Aceste aplicații de editare folosesc protocoalele de transfer de fișiere (FTP, SFTP) pentru a se conecta la serverul web și a descărca fișierele pe care le dorești să le editezi. După ce modificările sunt făcute, fișierele pot fi salvate și încărcate înapoi pe server pentru a le publica pe site.

Trebuie să ne asigurăm că avem o copie de rezervă a fișierelor înainte de a le edita direct pe serverul web. Modificările directe pe server pot avea consecințe ireversibile și este important să fim atenți și să facem backup regulat al fișierelor pentru a evita pierderi de date.