**Structura de baza**

HTML e folosit pentru a structura continutul unei pagini.

Browserele folosesc documente HTML pentru a afisa pagina web

<!DOCTYPE html>

<html>

<head>

<title> </title>

</head>

<body> </body>

</html>

Tot ce se afiseaza in browser e scris doar intre <body>

Restul nu apare in browser.

Ceea ce este intre <head> este o descriere a fisierului.

**Elementele HTML**

!DOCTYPE html – inseamna ca e HTML5

<html> </html> - delimiteaza continutul paginii web,

<head> - se scrie descrierea documentului ca pagina sa fie cat mai bine cautat de motoarele de cautare

<body> - continutul, ceea ce apare in browser

Index.html e fisierul principal, cu el incepe orice proiect. Celelalte coduri, CSS, JS etc. se scriu in fisiere aparte.

* Fiecare tag poate avea attribute,elemente ce ofera informatii suplimentare despre un alt element, ce au forma

Nume=”valoare”

* **Atributele se specifica in tagul de start:**

<start atribut1=”…” atribut2=”…”>

**<Head> </Head>**

* Este pentru descrierea fisierului.
* Putem adauga titlul paginii cu <title></title>
* <meta charset="utf-8" /> - spune browserului sa foloseasca codificarea utf-8, care contine practic toate caracterele existente
* <link rel="icon" href="images\icon.png"> - adauga o imagine in browser la site. Link rel arata legatura dintre documentel actual si cel la care se face relatia. Href – HyperText referece arata tinta
* <meta name="keywords" content="business, money, teamwork" /> - cuvinte cheie pentru website
* <meta name="description" content="Acesta este un site in care se gasesc diferite informatii legate de activitati de Business" /> - descrierea websiteului
* <meta name="author" content="Mititiuc Eduard" /> - autor
* <meta name="viewport" content="width=device-width, initialscale=1.0" /> - seteaza zoom initial al paginii in browser

Width=device-width – stabileste latimea paginii pentru a urma latimea ecranului dispozitivului

Initialscale=1.0 – seteaza nivelul de zoom initial

* <meta name=”revised” content=”11/12/2022”/> - ultimul update
* <meta http-equiv=”refresh” content=”10;url=site”/>- se face redirect prin http-equiv=”refresh” la o alta pagina peste 10 secunde.
* <link rel="stylesheet" type="text/css" href="css\styles.css" /> - adauga un cod css la pagina
* <style type=”text/css”> cod CSS </style> adauga un cod CSS
* **../**fiolder1/folder2 – muta folderul actual unde se gaseste fisierul cu un folder in urma

**Taguri**

* Tot ce este intre <> sunt taguri
* <> - tag de inceput
* </> - tag de sfarsit
* <> </> folosite impreuna se numesc container tag. Nu toate elementele din HTML necesita folosirea amebelor in acelasi timp.
* Cea ce se gaseste intre tagurile <> element </> sunt elemente.
* <> - un tag care nu necesita tag de inchidere, ci doar de deschidere, se numeste tag gol.
* <p> </p> - Paragraf. Textul intre ele va incepe si se va termina din rand nou. Primul <p> indica primul rand nou lasat si </p> ultimul. Pot fi folosite si singure.
* <hNr> </hNr> - heading sau titlu(antent). 1 e cel mai mare, 6 cel mai mic.
* <img src=”locatie/link”> - afiseaza o imagine **src-source**
* <hr> - plaseaza o linie orizontala

Are atributele:

align – right,center,left

width – lungimea liniei

size – grosimea

noshade – fara umbra

color - culoare

* <br> - linie noua
* <address></address> - afiseaza o adresa(nu web)
* <blockquote></blockquote> - deplaseaza textul la dreapta de marginea stanga a paginii
* <q> </q> - ia textul intre “ ”
* <pre> </pre> - afiseaza textul exact asa cum il scriem, cu spatii, cu randuri noi
* <nobr> </nobr> - text afisat pe o singura linie
* <div> </div> - este folosit, deobicei, pentru a formata un text cu stiluri de tip CSS deja definite. El nu face nimic propiu zis.Poate folosi si attribute direct. De ex:

Avem un CSS:

Nume {

…..

}

<div class=”Nume”> Text formatat de Nume </div>

* <abbr> </abbr> - este folosit pentru prescurtari
* <acronym> </acronym> - este folosit pentru a arata prima litera a unor cuvinte ce creaza un tot intreg.

abbr si acronym folosesc atributul title=”Titlu” ca atunci cand se va plasa mouse pe ele sa aratae cuvantul/ele in forma lor intreaga.

* <del> </del> - text sters. E afisat cu o linie, de parca e taiata, ca si <s>, doar ca e tratat diferit de browser/ moto de cautare.
* <ins> </ins> - text inserat. E subliniat ca si <u>, doar ca e tratat diferit de browser si motoare de cautare
* <mark> </mark> - text marcat(cu coloare din urma lui)
* <font> </font> - modifica fontul, dar si alte caracteristici ale textului

Are atributele:

color – culoare text

face – font

size – marime (intre 1 si 7). Daca nu este setat, browser alege o marime automata, la fel cum face si cu fontul. Daca punem +2, size va fi cu 2 mai mare ca cel implicit, iar -2 cu 2 mai mic ca cel implicit.

weight – grosime

* <b> </b> - bold
* <strong> </strong> - evidentiat. Arata exact ca bold, doar ca e accentuat de motoarele de cautare cand se cauta ceva.
* <i> </i> - italic
* <u> </u> - subliniat
* <big> </big> - text mai mare
* <small> </small> - text mai mic
* <strike> </strike> sau <s> </s> - text taiat
* <sub> </sub> - text ca putere jos
* <sup> </sup> - text cu putere sus

**Liste**

* **Unordered List(lista neordonata) –** lista ale carui elemente nu sunt numerotate sau ordonate.

Incepe si se termina cu <ul> </ul>(default apar cerculete)

Fiecare element incepe si se termina cu <li> </li>

li – list item

* putem folosi si atributul **type = tip** pentru a nu arata doar cifre.

Type = square – patratel

Type = disc – cerc plin

Type = circle – cerc gol

<ul type=square> .. </ul>

* **Ordered List** – lista ale carui elemente sunt ordonate cu indexi.

Incepe si se termina cu <ol> </ol>(default apar numere)

Fiecare element incepe si se termina cu <li> </li>

li – list item

* putem folosi atributul **start=nr** pentru a incepe lista de la acel nr, nu de la 1, asa <ol start=5> … </ol>
* putem folosi si atributul **type = tip** pentru a nu arata doar cifre.

Type = A – arata litere

Type = I – cifre romane

<ol type=A> .. </ol>

* Pentru a arata numerele in ordine inversa, folosim atributul **reverse**

<ol reverse> </ol>

* **Description list** – este o lista cu termeni unde fiecare termen are o descriere.

Incepe si e termina cu <dl> </dl>

Fiecare termen e pus intre <dt> </dt>

dt – description termen

Dupa fiecare termen urmeaza descrierea intre <dd> </dd>(incepe automat din aliniat cu o -)

dd – description description

**Tabele**

* Pentru crearea lui se foloseste atributul **<table></table>**
* <tr> </tr> - defineste o linie (tabel row)
* Un tabel nu are in mod default un border.
* Intre **<tr> </tr>** se utilizeaza **<td> </td>** (tabel data) pentru a insera date in linia data, adica pe orizontal.
* Pentru a oferi un titlu unui tabel, se foloseste tagul **<caption>** si poate sa nu fie inchis! Dupa el se pune direct titlul. Trebuie pus dupa **<table>** neaparat!

**<table> <caption>Titlu tabel </table>**

* **Caption** poate lua atributul **align=locatie**, unde locatie poate fi right,left,top,bottom. Automat el ia buttom si center. Are si in CSS **caption{ }**si aliniearea se va face cu **caption-side:**
* Putem crea linii si coloane de atribute, adica care sa arate ce anume vor fi valorile. Aceste celule se numesc celule cap(tabel header) si au tagul **<th> atribut </th>.** Textul e automat in bold si central. Ele sunt ca si <td> doar ca sunt automat centrale si in bold.
* Putem adauga si imagini cu **<img src=”poza.png”>.** Acest tag are atributele:

**Src** – source

**Alt** – alternativ(adica un text care apare in locul imaginii, daca aceasta nu a putut fi incarcata)

**Hspace** – horizontal space, care arata spatiul din partea stanga si dreapta a imaginii

**Vspace** – vertical space, arata spatiul din partea de sus si jos a imaginii

**Atributele Tabelelor**

**Tabel:**

* **Border** = latime sau doar **Border**(default = 1)

**<table border=2> ... </table>**

Acest atribut este prezent si in CSS, doar ca dupa marimea in px se pune **solid! Mai mult ca atat, daca va fi folosit doar de table{} se va crea doar un chenar in jurul la tabel, nu propriu zis linii la celule.**

* **Align** = directie

Left – tabelul e in stanga si textul plasat dupa el va fi in dreapta lui

Right – tabelul e in dreapta si textul plasat dupa el va fi in stanga

Center – tabelul e in centru si textul dupa el va fi mai jos de el, la stanga.

* **Width** si **height** care exprima lungimea si latimea tabelului in px sau %. Pot fi aplicate si pentru **<td>,** dar daca de ex prima linie are o celulta de 100px, deci si urmatoarea linie va avea celulta de desupt de aceeasi marime automat. Nu pot avea lungimi sau latimi diferite celulele una sub alta! Se ia automat cea mai lunga.
* **Cellspacing =** marime – distanta dintre fiecare celula din tabel
* **Cellpadding –** distanta dintre continutul celulei si marginile ei
* **Bgcolor**
* **Background = image.png –** pune o imagine pe fundal la tabel
* Pentru a crea o celula sub care vor 3 celule,pe orizontala, de ex, ce vor avea toate 3 in total lungimea ei, se foloseste **colspan=numar** la <td> sau <th>, de aici si numele, adica col de la coloane si arata ca se vor gasi mai multe coloane sub o celula
* **Rowspan**- face ca celula sa se intinda(sa ocupe) pe x randuri, in jos deci
* **Colspan** – face ca celula sa se intinda(sa ocupe) pe x coloane, pe orizontala deci, in stanga anume.
* Tagul **<colgroup> </colgroup>** creaza un grup de coloane. Are atributele:

**Span** = numarColoane

**Width –** lungimea fiecarei coloane

**Align**

Putem sa editam si fiecare coloana aparte prin folosirea in <colgroup> </colgroup> a tagului **<col>**

Dupa, cand vom folosi <tr> si <td>, ele vor lua automat ce am

setat in <cologroup>, dar numai daca sunt atatea cate sunt setate. Daca adaugam mai multe coloane, ele vor fi cu valori default, insa toate randurile din coloane vor lua atributele din <colorgroup>

<col> poate fi folosit si fara <colgroup> si tot poate folosi span.

<col span=3> se creaza 3 coloane, nu grupuri!

**Continut**

* **Width** si **height** pot fi aplicate si pentru **<td>,** dar daca de ex prima linie are o celulta de 100px, deci si urmatoarea linie va avea celulta de desupt de aceeasi marime automat. Nu pot avea lungimi sau latimi diferite celulele una sub alta! Se ia automat cea mai lunga.
* **Align** poate fi folosit si pentru <tr> si <td> , si el aliniaza pe orizontala.Ia atributele

**Left**

**Right**

**center**

* **Valign –** alinieaza pe verticala, dar deja ia atributele:

**Baseline –** nu aliniaza dupa celula, ci dupa linia unde se amplaseaza textul in mod normal in celula.

**Bottom**

**Top**

**Middle**

* **Bgcolor**
* **Background = image.png –** pune o imagine pe fundal la <tr> sau <td>

**Entitati**

* Sunt folosite pentru a afisa caractere pe care HTML le foloseste si are nevoie de ele.

&lt este < (lesser than)

&gt - >

&amp - & (ampersand)

&apos – ‘

&quot – “

&nbsp – spatiu alb

**In HTML spatiile mai multe tot sunt tratate ca unul!**

**Afisarea literelor romanesti**

* &#cod\_unicod\_in\_baza\_10

Sau

* <meta http-equiv=”Content-type” content=”text/html; charset=utf-8”/>

**Atribute**

**Se pun direct dupa elementul din <>**

**<hr align=”center”> </hr<**

**“” sunt optionale, dar obligatorii cand valoarea e formata din mai multe cuvinte**

**px este optional, de ex width = 150 => width = 150px**

**<hr topmargin=50> </hr>**

* bgcolor – culoarea fundalului
* text – culoarea textului
* link – culoarea linkurilor
* alink – culoarea linkului active
* leftmargin – distanta dintre marginea stanga a ferestrei browserului si marginea stanga a continului paginii
* topmargin - distanta dintre marginea de sus a ferestrei browserului si marginea de sus a continului paginii
* align

**Referinte**

* Referintele mai sunt numite si linkuri. Linkurile reprezinta conexiuni ale anumitor parti din pagini. Ele sunt folosite cu **<A> </A>,** ce trebuie neaparat sa contina atributul **href**

Se foloseste id pentru orice tag, ca sa indicam ca catre el se va ajunge de altundeva.

<A href = #IDdestinatie id=nume></A> - si A poate avea id!

<p id=nume></p>

Asa aratam unde va sari:

* <A href=**#ID**destinatie>Click</A> - sarim la un element din aceeasi pagina, asta anume si arata #
* <A href=Site></A> - daca sarim la un site, **nu mai folosim #,** ci indicam adresa!
* <A href=fisier.html></A> - sarim la o pagina din acelasi folder
* <A href=fisier.html#ID> </A> - sarim la alt fisier la id al elementului indicat.
* Linkurile pot fi de 2 tipuri:
* Interne – se sare undeva in aceeasi pagina
* Externe – se sare undeva in alta pagina.
* Poate fi folosit si atributul target=”mod” pentru a specifica cum se va deschide linkul.

\_top – incarca pagina intr-o fereastra peste cele existente

\_self – incarca pagina in cadrul curent

\_parent – incarca pagina in cea parinte

\_blank – intr-o fereastra noua, fara nume

**ID si class**

* ID

Este folosit mai mult pentru elemente care apar o singura data in pagina.

In CSS se defieste mereu cu # la inceput

#idexemplu { }

href la fel mereu va folosi # la inceput de id

<p href=”#unid”> </p>

Pentru a adauga un id la o referinta, nu folosim #

<a id=”faraDiez”>Aici se va ajunge </a>

* Class

Este folosita pentru elemente care apar odata sau de mai multe ori in pagina

In CSS incep mereu cu .

.numeclasa { }

Poate fi folosit in CSS si pentru anumite taguri, si va avea forma

tag.clasa {}

De exemplu:

p.nume { }

si apoi va fi folosit asa <p class=”clasa”></p>, fara . !

Daca are doar . si nu incepe cu vreun tag, poate fi folosit de oricare tag

**Clase Combinate**

* Un tag poate folosi si stilurile unei clase si ale sale caracteristice. De ex:

p{

color: red;

}

.clasa{

color: yellow;

background-color: black;

}

<p class=”clasa”>text</p>

Textul dat va fi yellow, caci se ia ultimul stil mereu, dar va avea si backgrounf black, caci se combina ci stilul p din CSS, cat si cel al clasei.

.clasa p{

color=black;

Un asa mod de a declara o clasa va intra in forta cand intai se preiau proprietatile clasei, si apoi se defineste un p:

<div class="clasa"><p>Text</p></div>

Aici se vor prelua intai stilurile clasei, apoi ale lui p si apoi ale lui .clasa p

**Harti din imagini**

Permit declararea unor zone din imagine ca legaturi catre alte pagini web.

* Atribute pentru <img> :

**Src** – sursa imaginii

**Usemap** =”#nume” – e ca un id pentru imaginea data.

* **<map> </map>** - defineste o harta intr-o imagine. Harta este o zona cu clickuri ce fac ceva, dar se pune deobicei peste imagini.
* Atribute:

**Name=”nume”** – fara #! Creaza o legatura cu imaginea peste care se doreste crearea hartii.

* In map se foloseste tagul **<area> (Fara tag de inchis!)** pentru a crea propriu zis zona de harta. Are atributele:
* **Shape** = “tip”, cu tip ce poate fi **rect**(reptunghi),**circle**(cerc),**poly**(poligonala)
* **Cords** = “,…,” – indica coordonatele zonei harta(coordonatele din imagine, nu pagina!)

Pentru rect, se indica x,y,lung,latime

Pentru circle se indica x,y,raza

Pentru poly se indica pentru fiecare latura coordonatele

* **Href** = link catre pagina

**Cadre**

Reprezinta parti din alte pagini ce sunt afisate pe pagina principala.

* Se introduc prin **<iframe> </iframe>**
* Se poate introduce un text intre ele ce va aparea daca browserul nu suporta
* **Atribute:**

src – fîșierul sau adresa acestuia

height – înălțimea cadrului

width – lățimea cadrului

frameborder - grosimea bordurii (chenarului) (1 – da 0 – nu)

scrolling - adaugă bare de defilare cu valorile yes, no și auto

align - aliniază cadrul (left, right, center, top, bottom, middle)

vspace – distanța peste și sub cadru

hspace – distanța în stânga și în dreapta de cadru

name – specifica un fel de id

**Audio**

* **<audio></audio>**

Are atributele;

**controls**, care adauga play,pause,volum

**loop –** stabileste daca sunetul va fi repetat la nesfarsit, si iar valorile **true** si **false**

**volume** – 0 la 100

* Intre <audion> </audion> se foloseste **<source src=”adresa”>**

Daca se specifica mai multe, browserul va alege primul suportat

* Se poate introduce un text intre ele ce va aparea daca browserul nu suporta
* Se pune mereu atribut type = “audio/mp3”

**Video**

* **<video></video>**

Are atributele;

**controls**, care adauga play,pause,volum

**autoplay –** la deschiderea paginii, se porneste automat

* Intre <video> </video> se foloseste **<source src=”adresa”>**

Daca se specifica mai multe, browserul va alege primul suportat

* Se poate introduce un text intre ele ce va aparea daca browserul nu suporta
* Se pune mereu ca atribut type = “video/mp4”

**Pseudo-Clase**

* Pseudo-clasele sunt folosite pentru a defini un statut special unui element, de exemplu sa il stilizam cand userul da click pe el, sa stilizam linkurile vizitate si nevizitate, un element cand e focusat.
* **Element:pseudoclasa{**

**}**

**Element –** poate fi selector,id,clasa

Putem avea si de genul: **p i:first-child** { }

* Pseudo-coduri:
* **Active** – cand tinem apasat pe link
* **Link** – pentru linkuri nevizitate
* **Visited** – link vizitat
* **Hover** – cand mousul se afla deasupra lui
* **First-child** – adauga stiluri elementului care e primu, fie din body,div sau p
* **Last-child**
* **Lang**(cuvant) – doar pentru acele element ce folosesc sue le lang, de ex:

.test:lang(ceva) { .. }

<ul>

<li class=test> nu se intampla nimic</li>

<li cilass=test lang=ceva> se va intampla ceva </li>

</ul>

* **Focus** – cand se afla cursoul de scriere in interiorul unui text
* Putem combina si cu clase,id:

.clasa:active{ }

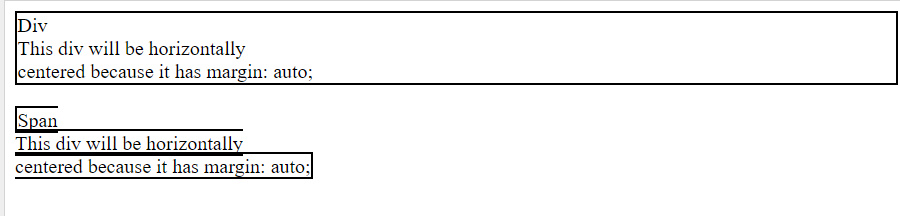
**Pseudo-element**

* Pseudo-elementele au rolul de a stiliza doar anumite parti dintr-un element.
* Au forma **selector::pseudoelement { }**
* **Atribute:**
* **After –** adauga continut dupa un element HTML
* **Before –** adauga continut inaintea unui element HTML
* **First-letter** – adauga primului caracter
* **First-line** – primei linii
* **Selection** - modificam elementul cand e selectat

Pseudo-clasele si pseudo-elementele pot fi combinate: **Obiectcss:Pseudo\_clasa::Pseudo-element**

**Div vs Span**

* **Div –** e folosit pentru a crea un bloc din pagina ce contine mai multe elemente in el. El include toate elementele sale intr-un bloc. La folosirea lui radius, va aparea un dreptunghi.
* **Span –** e folosit pentru a modifica un element cocnret dintr-un bloc, fiind deobicei in interioriul lui <div>. El nu creaza un border pentru tot continutul, ci fiecare linie noua din el va avea borderul ei.



**Machetare**

**Machetarea** este modul in care repartizam sau grupam elementele intr-o pagina.

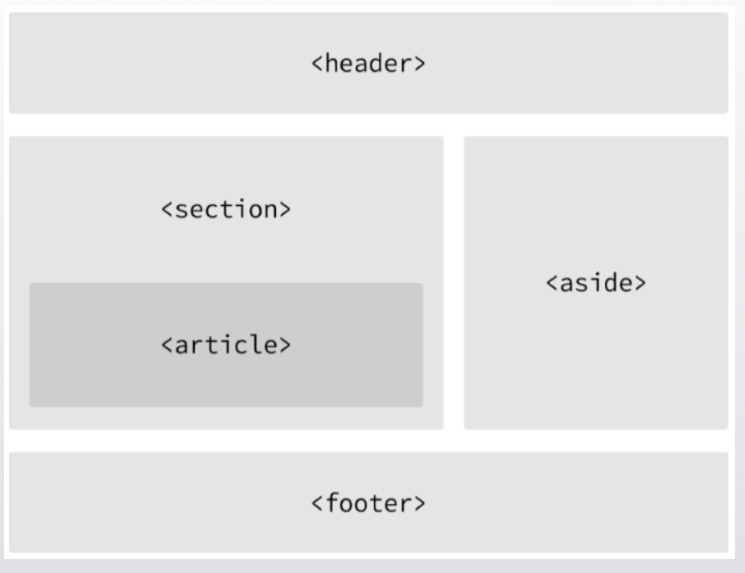
Cea mai buna machetare a paginii este prin folosirea tagului <div> pentru fiecare parte a paginii.

Principii:

1. Separarea continutului si designului – Codul HTML trebuie sa contina doar taguri, nu si cod CSS. Codul CSS e scos inafara.
2. Pagina WEB se descompune in blocuri prin <div>

**Machetare in HTML5**

In html5, structura unei pagini e impartita asa:



* Header - <header> </header> - este antetul paginii
* Nav - <nav> </nav> - e pentru sectiunea dedicata navigarii in pagina, ca butoane de navigare, inapoi etc... Are doar cele mai importante sectiuni in pagina
* Articol - <article> </article> - identifica o sectiune independenta e continut, ca postari, articole etc.
* Sectie - <section> </section> - indica o sectiune, adica o grupare de coontinut.
* Aside - <aside> </aside> - sunt bare laterale,ca autorul articolului.
* Footer - <footer> </footer> - finalul unei pagini, articol.