Cuprins

Table of Contents

[**Despre proiect** 2](#_Toc40180397)

[**I. Generalitati** 3](#_Toc40180398)

[**II. HTML** 8](#_Toc40180399)

[**III. Descrierea Aplicatiei** 19](#_Toc40180400)

[**IV.Bibliografie** 23](#_Toc40180401)

# 

# **Despre proiect**

Acest site este creat cu scopul de a informa cititorul despre cei doisprezece Olimpieni din Mitologia Greaca,o Mitologie foarte cunoscuta de-a lungul lumii.

Site-ul contine informatii si imagini specifice pentru formarea unei idei generale cand vine vorba de aceste zeitati.

.

# **I. Generalitati**

**Tehnologia informației** sau și **Tehnologia informației și a comunicațiilor**, abreviat (cel mai adesea **IT**) **TI** respectiv **TIC**, este tehnologia necesară pentru prelucrarea (procurarea, procesarea, stocarea, convertirea și transmiterea) informației, în particular prin folosirea [computerelor](https://ro.wikipedia.org/wiki/Computer) (calculatoarelor electronice) .

Termenii [englezești](https://ro.wikipedia.org/wiki/Englez%C4%83) corespunzători sunt **Information Technology** și **Information and Communication Technology**, abreviați **IT** și respectiv **ICT**. Prescurtarea **IT**, care deseori se folosește și în română, se pronunță /‿ai ti/.

Termenul a fost folosit pentru prima dată în 1958 într-un articol publicat în revista [*Harvard Business Review*](https://ro.wikipedia.org/w/index.php?title=Harvard_Business_Review&action=edit&redlink=1), articol în care autorii Leavitt și Whisler afirmă: «Noua tehnologie încă nu are un nume bine stabilit; noi o vom denumi „tehnologia informației”».

Tehnologia informației este un domeniu element de legătură între [electrotehnica](https://ro.wikipedia.org/wiki/Electrotehnic%C4%83) clasică și mult mai noua [informatică](https://ro.wikipedia.org/wiki/Informatic%C4%83), multe instituții profesionale profilate (inițial) electrotehnic extinzându-se cu timpul și spre domeniul de tehnologia informației. A nu se confunda cu Informatica (*Computer Science*) și nici cu [sistemele informatice](https://ro.wikipedia.org/wiki/Sistem_informatic).

Tehnologia informației găsește aplicare pe multiple domenii legate de date și informații, cum ar fi: [procesoare](https://ro.wikipedia.org/wiki/Procesor), [calculatoare](https://ro.wikipedia.org/wiki/Calculator), [hardware](https://ro.wikipedia.org/wiki/Hardware) și [software](https://ro.wikipedia.org/wiki/Software), [limbaje de programare](https://ro.wikipedia.org/wiki/Limbaj_de_programare), structuri de [date](https://ro.wikipedia.org/wiki/Dat%C4%83) și altele. Sunt considerate ca făcând parte din largul domeniu IT toate elementele care prelucrează, într-un fel sau altul, date, informații sau cunoștințe.

Ocupațiile specialiștilor din acest domeniu sunt foarte variate, de la instalarea de software aplicație și până la proiectarea unor [rețele de calculatoare](https://ro.wikipedia.org/wiki/Re%C8%9Bea_de_calculatoare) complexe și a [bazelor de date](https://ro.wikipedia.org/wiki/Baz%C4%83_de_date) de [informații](https://ro.wikipedia.org/wiki/Informa%C8%9Bie). Câteva sarcini tipice sunt: managementul datelor, construcția de hardware pentru calculatoare, proiectarea de software, administrarea unor întregi sisteme informaționale. În ultimii ani sfera de răspândire s-a lărgit mult, cuprinzând acum nu numai calculatoare și rețelele lor, dar și telefoane mobile celulare, intelifoane (smartphones), televizoare cuplabile la Internet, automatizarea automobilelor, aplicații militare și multe altele. De aceea, cererea de specialiști de vârf pe piața mâinii de lucru a domeniului este constant mare.

De aceea și instituțiile superioare de învățământ au introdus noi catedre, cursuri, profile și examene corespunzătoare, pentru a ține pasul cu aceste tendințe din cadrul IT, pe lângă domeniile [Informatică](https://ro.wikipedia.org/wiki/Informatic%C4%83) și [Sisteme informatice](https://ro.wikipedia.org/wiki/Sistem_informatic).

În 2009 piața mondială de servicii de IT a totalizat 763 miliarde de dolari ([USD](https://ro.wikipedia.org/wiki/USD)).

|  |
| --- |
|  |

Principala provocare a TIC o reprezinta crearea unui mediu propice pentru invatare, deschis invatarii, implicit noile tehnologii joaca un rol esential in trecerea de la mediul de invatare centrat pe predare, pe profesor, la mediul de invatare centrat pe elev.

Marea provocare a pedagogiei, reliefata de catre George Vaideanu in volumul „Educatia la frontiera dintre milenii” (Vaideanu, G., 1988) si de catre Miron Ionescu in  lucrarea „Demersuri creative in predare si invatare” (Ionescu, M., 2000), si anume mutarea accentului de pe predare pe invatare, de pe informativ pe formativ, de pe instructie pe educatie, devine implicit si tinta principala a mediilor electronice de invatare. Schimbarea rolului profesorului din acela de a fi sursa principala de transmitere a cunostintelor si informatiilor pentru elevi, in acela de a deveni un colaborator al elevului, un coleg implicat si el in procesul cunoasterii se coreleaza cu trecerea treptata a elevilor din simpli receptori pasivi ai unor informatii si cunostinte, in constructori activi ai propriei lor formari.

Calculatorul electronic, Internetul, posta electronica reprezinta realitati pe care elevii le experimenteaza in fiecare zi. Cadrul didactic este astfel pus in situatia de a opta: prefera continuarea demersului sau educational in mod traditional, ignorand tendintele de schimbare de paradigma care se prefigureaza, sau accepta provocarea, implementand noile tehnologii in activitatea sa didactica. Pentru a face fata insa cu succes provocarilor tehnologice actuale, sunt necesare eforturi de formare profesionala atat din partea cadrelor didactice, cat si din partea formatorilor lor, in perioadele de formare initiala si continua.

Pornind de la aceste considerente, vom analiza mai sistematic cateva tendinte actuale in educatie, tendinte care sustin importanta integrarii noilor tehnologii ale informatiei si comunicarii in scoala si in formarea individuala si profesionala, dar mai cu seama in formarea complexului de competente necesare profilului profesional al cadrului didactic. Tehnologiile informatie si comunicarii ofera un mediu favorabil pentru restructurarea intregului complex de competente necesare cadrului didactic, iar tendintele urmatoare sustin aceasta premisa.

Importanta utilizarii tehnologiilor pentru dezvoltarea personala si institutionala a fost si mai mult scoasa in evidenta prin introducerea termenului: Tehnologie a informatiei si invatarii (Information and Learning Technology (ILT)), termen utilizat in mod deosebit in programele de educatie continua, pentru referirile la aplicatiile IT si TIC in desfasurarea eficienta a procesului de predare si invatare.

**Securitatea informatiilor**

Una dintre problemele care au aparut in urma prelucrarii informatiilor folosind instrumentele TIC este cea a securitatii informatiilor. Aceasta problema implica urmatoarele aspecte:

a)securitatea fizica a datelor;

b)securitatea impotriva furtului sau a indiscretiilor;

c)protectia impotriva virusilor informatici.

***a)Securitatea fizica a datelor***

Datele sunt pastrate pe suporturi materiale care pot fi supuse distrugerii. Pentru protejarea lor trebuie respectate urmatoarele reguli:

incintele in care se gaseste tehnica de calcul trebuie protejate impotriva accesului persoanelor neautorizate si impotriva incendiilor;

trebuie sa se asigure calculatorului un mediu ambiant corespunzator din punct de vedere al temperaturii (temperatura optima este 21°C), fara umiditate mare, fara praf si fum. La o temperatura prea scazuta sau prea ridicata, componentele calculatorului nu mai functioneaza bine si pot sa apara erori in exploatarea programelor sau blocarea acestora. Umiditatea, praful si fumul afecteaza contactele electrice si capetele de citire-scriere ale unitatilor de memorii externe, putand duce la scrierea sau citirea eronata a datelor;

calculatorul trebuie sa fie protejat la supratensiuni, care pot distruge unele componente;

persoanele care folosesc suporturile de informatie trebuie sa cunoasca modul in care trebuie sa le utilizeze. Suporturile de informatie trebuie pastrate in cutiile lor si nu se atinge suprafata lor nici cu mana, nici cu obiecte contondente. Trebuie tinute departe de surse de caldura foarte puternice si nu se varsa lichide pe ele. Suporturile magnetice trebuie tinute departe de campurile electromagnetice puternice.

daca se intrerupe brusc alimentarea cu tensiune electrica, datele care se gasesc in memoria interna se pierd. Pentru prevenirea acestor pierderi, este bine ca operatiile de salvare sa se faca foarte des, iar daca produsul software permite, sa se configureze aplicatia astfel incat salvarea sa se faca automat, dupa un numar de minute.

***b)Securitatea impotriva furtului si a indiscretiilor***

Cand se lucreaza intr-o retea de calculatoare, la datele de pe un calculator oarecare poate sa aiba acces foarte multe persoane. In acest caz, apar doua categorii de probleme: pot sa aiba acces la date persoane neautorizate sau pot sa aiba acte de distrugere a datelor.

Rezolvarea acestor probleme se poate face prin:

construirea unui **sistem de parole**, pe mai multe niveluri, care sa controleze accesul diferentiat la date;

**criptarea** datelor (codificarea datelor dupa un anumit algoritm).

Un pericol pentru securitatea datelor dintr-o retea de calculatoare il reprezinta asa numitii „**hackers**” si „**crackers**”. Sunt persoane care, prin intermediul unui calculator, reusesc sa treaca de nivelurile de securitate ale unei retele pentru a ajunge la informatii. Hackers o fac pentru a se distra iar crackers – fie pentru a obtine informatii pe care sa le foloseasca in interes personal, fie pentru a introduce in retea programe care sa distruga datele.

***c)Protectia impotriva virusilor informatici***

**Virusul** este un mic program executabil care are doua caracteristici:

este capabil sa realizeze copii dupa el insusi si sa includa acest cod executabil, obtinut in urma multiplicarii, intr-un alt program, care devine gazda si pe care il infecteaza;

este capabil sa execute o actiune bine definita, care uneori poate sa fie o actiune de distrugere a altor programe sau a datelor.

In momentul in care ati sesizat anomalii in functionarea calculatorului, este posibil ca ele sa se datoreze unui virus. Exista semne care pot da de gandit utilizatorului, cum ar fi: incetinerea vitezei de lucru a calculatorului, scaderea spatiului liber de pe disc sau cresterea numarului de blocuri defecte.

Virusul este o boala a calculatorului. Pentru a trata un calculator virusat este nevoie de un program antivirus.

Exemple de programe antivirus: Norton Antivirus, AVG, Avast, NOD32.

Programele moderne antivirus contin trei module:

un modul care asigura recunoasterea virusului;

un modul care selecteaza fisierele contaminate si le curata;

un modul care monitorizeaza functionarea calculatorului si transmite mesaje de avertizare atunci cand un virus incearca sa patrunda in sistem.

Programele antivirus:

pot sa verifice,la cererea utilizatorului, daca exista virusi in sistem sau

pot sa fie rezidente in memorie, pentru a supraveghea in permanenta sistemul (in acest caz, scade viteza de lucru a calculatorului).

**Drepturi de autor si legislatie**

**Licenta software** este un document in care producatorul de software specifica drepturile utilizatorului cu privire la programul achizitionat si este singura proba legala pentru dreptul de folosire.

Utilizatorul are nevoie de cate o licenta pentru fiecare produs software pe care-l foloseste. Pentru protejarea programelor cu licenta impotriva utilizarii ilegale, unii producatori folosesc **cheia de licenta**. Aceasta este un sir de caractere care are rolul de parola pentru instalarea unui produs software comercial.

„**Pirateria software**” inseamna folosirea unui produs software fara licenta, adica fie instalarea pe un calculator a unui software pentru care nu s-a platit licenta, fie folosirea pe mai multe calculatoare a aceluiasi produs software pentru care s-a platit o singura licenta, fie multiplicarea pe suporturi de informatie si comercializarea neautorizata a unui produs software.

Legislatia din Romania referitoare la produsul software:

**Legea 8/1996** – garanteaza dreptul de autor asupra oricarei opere de creatie intelectuala (opere literare, artistice, stiintifice, **inclusiv programele de calculator)**;

**Ordonanta Guvernului 124/2000** – prevede ca persoanele fizice sau juridice care produc, distribuie, comercializeaza sau inchiriaza programe pentru calculator pe teritoriul Romaniei au obligatia sa se inscrie in **Registrul Programelor pentru Calculator (RPC)**, prin intermediul Oficiului Roman pentru Drepturile de Autor (ORDA).

**Legea 677/2001** – operatorii (orice persoana fizica sau juridica) care prelucreaza date cu caracter personal au obligatia de a asigura securitatea acestor date. Activitatea operatorilor este monitorizata si controlata de **Autoritatea de supraveghere**.

Delictele cu ajutorul calculatorului pot lua urmatoarele forme:

**distrugerea informatiei** – se manifesta prin stergerea voluntara a datelor sau a programelor, fie prin producerea si raspandirea intentionata a virusilor;

**manipularea** – se manifesta prin transmiterea prin intermediul retelelor publice a unor informatii false;

**furtul** – se manifesta prin sustragerea de echipamente, de programe sau de date si prin practicarea pirateriei software (comercializarea si folosirea neautorizata a programelor de calculator).

**Probleme de ergonomie si protectie a muncii**

Unul dintre scopurile pentru care a fost creata si dezvoltata tehnologia informatiei si a comunicatiilor a fost cresterea productivitatii muncii. Dar s-a constatat, in cazul persoanelor care lucreaza intensiv in pozitia de introducere a datelor ca, de multe ori, productivitatea muncii nu este foarte crescuta, din cauza unei pozitii inadecvate in timpul lucrului cu calculatorul. O data cu cresterea gradului de folosire a calculatorului in activitatile zilnice si a cresterii numarului de persoane care isi petrec foarte mult timp lucrand cu calculatoare personale, a crescut si interesul pentru studiile ergonomice asupra factorului uman, in interactiune cu calculatorul. In urma studiilor ergonomice, s-a constatat ca pozitia in fata calculatorului este foarte importanta deoarece, pe langa aparitia unor stari de disconfort sau de oboseala (migrene sau dureri de spate sau de ceafa), pot fi provocate si diferite afectiuni. Monitorul joaca si el un rol important in afectiunile care pot fi dobandite atunci cand lucrati cu calculatorul, caci el poate afecta vederea, auzul si poate crea efecte neplacute datorita radiatiilor electromagnetice. Si tastatura poate genera probleme de sanatate, prin executarea repetata a acelarasi miscari. De aceea trebuie respectate urmatoarele indicatii:

***Mobilierul ergonomic si incaperea:***

**Biroul –**inaltimea biroului trebuie sa fie de 64 – 74 cm si ajustabila.

**Scaunul –** spatele si inaltimea scaunului trebuie sa fie ajustabile.

**Dispozitivul pe care se fixeaza documentul in lucru –**trebuie sa fie ajustabil.

**Sursa de lumina –** trebuie pozitionata deassupra monitorului, astfel incat sa asigure o iluminare buna.

**Suporturi de sprijin –** pentru picioare si incheieturile mainilor se pot folosi suporturi care sa le asigure o pozitie cat mai relaxata.

**Amplasarea calculatoarelor in incapere** – pentru a se evita efectul negativ al radiatiilor electromagnetice, calculatoarele trebuie amplasate astfel incat distanta dintre o persoana care lucreaza si orice calculator vecin sa fie de cel putin 1 m.

# **II. HTML**

Una dintre cele mai importante tehnologii de pe internet este HTML-ul. El este una din componentele de baza pe care orice website sau aplicatie web se bazeaza.

Toate acestea la un loc formeaza structura de baza a oricarui website si de aceea sunt foarte importante in web design.

**Diferenta dintre tehnologii**

Diferenta dintre ele este ca HTML in principiu, ne arata structura si continutul unei pagini web. CSS ne va spune cum acea pagina va fi stilizata (cum va arata, sau ce culori va folosi) iar Javascript ne ofera interactivitatatea dintr-o pagina.

Fara nici o indoiala HTML-ul este cea mai importanta componenta dintre cele 3, pentru ca fara el, celelalte 2 nu au rost.

Putem chiar sa construim un website folosind doar HTML. Fara css, nu va arata, vizual foarte bine, sau fara JS, nu va fi interactiv, dar continutul si structura site-ului, va fi acolo.

**Cum functioneaza HTML-ul**

HTML este un limbaj de tip **MARKUP**. Aceasta inseamna ca rolul lui principal este de a oferi un comportament special unei anumite parti dintr-o pagina. Si pentru a face asta trebuie ca acel continut sa fie cuprins intre doua **TAG**-uri.

Asadar, componenta de baza din HTML este un tag care doar ne spune ce functie are acel continut.

De exemplu, tag-ul “p” ne spune ca in interiorul lui avem un paragraf. Sau tagul “a”, ne spune ca avem o legatura catre o alta pagina, fie ea in acelasi website sau in exteriorul lui.

Html-ul are multe taguri, si ce semnifica fiecare in parte este mai putin important in acest moment, important acum este sa intelegem ca ideea de baza este foarte simpla:

aceea de a Iua un continut si de a ii oferi un comportament special

In 1990 **Tim Berners-Lee** impreuna cu **Robert Cailliau** a scris o propunere pentru crearea **World Wide Web**. Desi purta acel nume, la inceput acesta a fost folosit doar de cercetatori de la diferite universitati din SUA si Europa pentru a crea legaturi intre documente.

Dar motivele pentru care HTML a devenit foarte popular au fost ca acesta a fost oferit gratuit si pentru ca la baza, avea o idee foarte simpla prin care se puteau crea documente destul de complexe.

O pagina HTML nu este altceva decat un fisier text scris intr-un limbaj special. Un browser este un program care interpreteaza acest fisier si ii arata continutul.

Limbajul HTML este folosit pentru a defini continutul si structura unei pagini dar nu este responsabil de cum arata pagina.

**Ce este un element**

O pagina HTML este alcatuita din elemente. Iar acele elemente sunt alcatuite din alte elemente.

In limbajul HTML, un elemente este format din continutul acestuia si unul sau doua **TAG**-uri.

Un **TAG** este o serie de simboluri ce arata inceputul sau sfarsitul unui element

Pentru a scrie un tag vom folosi simbolul <, apoi numele tagulului (de exemplu html) si apoi simbolul >. Toate la un loc vor arata astfel: <html>

Acesta este tag-ul de deschidere, sau cel ce arata ca incepe un element HTML. Iar pentru unele elemente, trebuie sa spunem si unde se termina. Pentru aceasta vom folosi tag-ul de inchidere care seamana foarte mult cu primul doar ca are si simbolul /, de exemplu: </html>.

Iar elementul final va arata astfel:

<html></html>

Acest element este primul folosit intr-o pagina HTML. Este un element esential si de fapt defineste ca acea pagina este una HTML.

Imediat in interiorul lui vom avea inca doua elemente (despre care vom discuta pe larg in capitolul urmator): <head></head> si <body></body>. Toate elementele vizibile dintr-o pagina sunt incluse in <body></body>.

**Un paragraf**

Daca vrem sa aratam in pagina un **paragraf**, vom folosi tag-urile <p></p>:

<p>continutul unui paragraf</p>

Asa ca vedem cum continutul paragrafului este cuprins intre tagul de inceput si cel de sfarsit.

Prima pagina

Aproape toate elementele in HTML se scriu in acest fel.

Dar indiferent de ce vom avea intr-o pagina, aceasta va trebui sa contina cel putin 3 elemente: <html></html>, <head></head> si <body></body>

Aceste 3 elemente sunt esentiale pentru a creea o pagina noua. Pentru moment nu vom discuta despre ele, vom face asta pe larg in capitolul urmator, dar este important sa stim ca acestea nu trebuie sa lipseasca din orice pagina.

Asadar, pentru a scrie un element HTML avem nevoie de un tag de deschidere si unul de inchidere.

Pentru un paragraf avem tagul <p> pentru a arata unde incepe paragraful si </p> pentru a spune ca s-a terminat

**Elemente care nu se inchid**

Nu tooate elementele au un tag de inchidere.

Nu toate elementele trebuie inchise, majoriatea da, dar sunt cateva elemente in HTML care nu au un continut propriu-zis asa ca nu e nevoie sa le inchidem.

**O noua linie**

De exemplu, browser-ul poate ignora o linie noua (atunci cand apasam ENTER) pe care noi o punem in mijlocul unui paragraf. Dar pentru a-l forta sa o arate vom folosi un elemente special, si anume: <br />

Acest elemnt nu are continut si nici un tag de inchidere, iar pentru a arata asta folosim semnul / dupa numele tag-ului.

**Imaginile**

Cred ca cel mai folosit element care nu are un tag de inchidere este cel pentru a arata o imagine in pagina.

Atunci cand vrem sa introducem o imagine in pagina noastra, vom folosi tag-ul: <img />

Nu vom include imaginea intre doua tag-uri ci ii vom spune elementului de unde sa citeasca aceasta image.

Ii vom comunica elementului calea catre imagine folosind un **atribut**, despre care vom discuta in continuare.

**Elemente cuibarite**

**Cum se pot combina elementele intre ele pentru a crea structuri mai avansate**

**Element**

Asa cum am vazut in cazul elementului <html></html>, includerea unor elemente in alte elemente este unul dintre cele primele lucruri pe care le vom face cand construim pagini web.

Pentru a crea pagini mai complexe vom fi nevoiti sa combinam elementele intre ele, nu doar sa le scriem consecutive.

**Un cuvant special**

De exemplu, am vrea intr-un paragraf ca primul cuvant sa iasa in evidenta.

Pentru aceasta, vom cuprinde primul cuvant din paragraf intr-un element special:

<p>

<strong>acest</strong> cuvant va fi ingrosat

</p>

**Iar rezultatul va fi: acest** cuvant va fi ingrosat

Tag-ul pentru a scoate un cuvand in evidenta este: <strong></strong>, iar vizual este similar cu ceea ce face butonul **Bold** dintr-un editor de text.

Ce este important cand folosim un element in interiorul altui element, este sa il inchidem inainte ca parintele lui sa se inchida.

De exemplu, daca vom folosi aceste tag-uri si pentru ultimul cuvant din paragraf, dar vom scrie tag-ul de inchidere </strong> dupa </p>, codul ar arata astfel:

<p>acest cuvant va fi <strong>ingrosat</p></strong>

Textul tot va arata bine:

acest cuvant va fi **ingrosat**

Si aceasta se intampla pentru ca HTML este un limbaj permisiv. El incearca sa interpreteze ce am vrut noi sa spunem, desi nu este corect semantic.

Este ca si cum am pune punctul de sfarsit de propozitie inainte de ultimul cuvant. In acest caz poate ca arata cat de cat bine, dar nu stim cum se va comporta acest cod in alte browsere.

Asta nu inseamna ca trebuie sa ne bazam pe acest lucru si sa scriem ordinea tag-urilor gresit.

**Copiii unui element**

In urmatorul caz, elementul <p></p> se numeste parinte:

<p>

<b>acest</b> cuvant va fi <strong>ingrosat</strong>

</p>

Iar elementele <strong></strong> sunt copii lui.

Pentru ca acest cod sa fie corect, este foarte important ca aceste elemente sa fie incluse in intregime in parintele lor.

Asta inseamna sa inceapa dupa tag-ul de deschidere al parintelui si sa se inchida inainte de tag-ul de inchidere al lui.

**Atribute**

Atributele se folosesc pentru a adauga functii noi unui element HTML

**Atributul unui tag HTML**

De exemplu, pentru tagul de imagine <img /> putem avea un atribut care sa precizeze sursa imaginii pe care vrem sa o afisam:

<img src="o-imagine.png" />

In acest caz src="o-imagine.png" este atributul tag-ului.

Un atribut are urmatoarea structura:

1. numele atributului src (prescurtare de la source)
2. semnul egal =
3. si apoi valoarea lui cuprinsa intre ghilimele "o-imagine.png" (un URL catre locul unde se afla imaginea)

In acest caz, atributul are un rol **functional**, pentru ca influenteaza modul in care se comporta elementul.

**Atributele unui element care se inchide**

Daca am vrea sa aplicam un atribut unui element care se inchide, precum unui paragraf, acesta ar arata astfel:

<p lang="ro">

aici avem un paragraf

</p>

Asadar, atributele pentru elementele care se inchid se trec in **primul** tag al lui.

In acest caz atributul este lang si este **de tipul informativ** pentru ca nu influenteaza cu nimic continutul paragrafului, ci doar ne ofera informatii suplimentare despre acesta.

**Atribute multiple**

Un tag poate avea mai multe atribute care ii ofera un comportament special:

<p id="un-id" class="o-clasa">

aici avem un paragraf

</p>

Elementul de mai sus contine doua atribute noi, pe care le vom folosi foarte des: id si class.

Acestea doua vor fi foarte folositoare cand vom discuta despre CSS si JavaScript.

**Tipuri de atribute**

Dupa cum am spus exista atribute **functionale**, care influenteaza comportamentul elementului HTML si atribute **informative**, care doar ofera detalii despre acel element.

Exista cateva atribute care se pot intalni la orice element (precum class si id) dar exista si atribute care sunt specifice doar unui element.

De exemplu pentru elementul <img />, avem atributul src.

Acesta este folosit de foarte putine elemente si in nici un caz de un paragraf. Asadar putem spune ca acest atribut este oarecum spcific elementului <img />.

O serie de elemente importante in HTML sunt cele folosite pentru a crea titluri si subtitluri. Ele sunt folositoare pentru ca ne ajuta sa delimitam anumite sectiuni ale unei pagini.

In functie de importanta lor aceste elemente au numele de la <h1> la <h6>

1. <h1></h1>
2. <h2></h2>
3. <h3></h3>
4. <h4></h4>
5. <h5></h5>
6. <h5></h6>

Ele se numesc **headings** si vom folosi <h1> atunci cand vrem sa precizam care este titlul principal intr-o pagina.

Apoi fiecare sectiune va fi delimitata folosind urmatorul element ca importanta <h2></h2>, apoi pentru fiecare subsectiune a lui vom folosi urmatorul element <h3></h3> si asa mai departe.

De exemplu, in aceata pagina titlul lectiei este incadrat in tag-urile <h1></h1>, iar fiecare subtitlul va folosi <h2></h2>.

**Cum se folosesc**

Aceste tag-uri se folosesc ca si cele pentru paragraf. Vom avea tag-ul de deschidere <h1>, apoi continutul lui (precum un titlu) si apoi tag-ul de inchidere </h1>:

<h1>Titlul unei pagini</h1>

**Diferenta vizuala dintre ele**

Dupa cum am spus aceste elemente arata importanta unui text si de aceea ele difera si vizual. Toate aceste tag-uri vor arata astfel:

Un text folosind tag-ul <h1>

Un text folosind tag-ul <h2>

Un text folosind tag-ul <h3>

Un text folosind tag-ul <h4>

Un text folosind tag-ul <h5>

Un text folosind tag-ul <h6>

Si primul lucru care va va trece prin minte este ca pentru a crea un text mai mare trebuie sa folosim un heading mai important, precum <h1></h1>, iar pentru unul mai mic putem folosi unul mai putin important.

Nu este o idee buna sa folosim un element HTML doar pentru a schimba modul in care arata un text. Pentru aceasta vom vorbi de limbajul CSS care ne ajuta sa stilizam orice element din pagina, si chiar sa facem un <h6> sa fie mai mare decat un <h1>.

Limbajul HTML trebuie folosit doar pentru a preciza care este rolul unui continut nu pentru a-l stiliza.

**Tipuri de elemente**

In HTML5 (care este ultima versiune de HTML) elementele sunt impartite in 7 categorii.

Eu as dori sa folosim o clasificare care a fost valabila pana la HTML4, dar pe care o consider mult mai folositoare cand cineva incepe sa invete despre HTML. Am putea vedea toatele elementele HTML impartite in doua categorii:

elemente de tip block

Pe scurt, inseamna ca elementele vor fi pe o singura linie. De fiecare data cand le folosim acestea se intind pe toata lungimea elementului parinte. Cateva elemente de tip block sunt: <p>, toate elementele de la <h1> pana la <h6>, <ul>, <li>.

De exemplu, am folosit elementul <p> pentru aceste doua paragrafe si vedem ca fiecare este pe linia lui si nu unul in continuarea celuilalt.

elemente de tip inline

Aceste tipuri de elemente vor aparea pe o singura linie si nu vor impinge continutul ce urmeaza dupa ele pe o linie noua.

De cele mai multe ori, ele apar in interiorul unui element de tip block.

De exemplu, in interiorul unui <p> cand vrem sa avem un cuvant ingrosat folosim elementul <strong>. El este unul de tip inline pentru ca nu forteaza textul de dupa el pe o linie noua.

In exemplul urmator, ambele paragrafe vor aparea pe o linie:

<p>aceste <strong>cuvinte</strong> vor aparea pe o singura linie</p><p>aici vom avea un rand separat</p>

Aceasta este marea diferenta dintre cele 2 tipuri de elemente, si desi suna simplu cauzeaza o groaza de probleme in web design.

Din fericire putem forta un element sa alterneze intre bloc si inline si pentru asta vom folosi CSS.

**Elementul DIV**

Un element general care poate fi folosit peste tot

Unul dintre cele mai folosite elemente in HTML este <div></div> (prescurtare de la document division).

Pana acum am vorbit despre elemente care dau un rost continutul, sau creaza o structura, dar nu si div-ul.

Div-ul este un element care nu are nici un rol semantic in HTML, exista doar pentru a grupa alte elemente.

De exemplu, inainte de HTML5 nu exista o metoda de a defini un element pentru un meniu, asa ca un web developer folosea un <div></div> care avea un atribut de tip id cu valoarea de meniu, si astfel definea rolul acestuia:

<div id="meniu">

...

</div>

In HTML5 (cea mai recente versiune a HTML) a aparut elementul de <nav></nav> care defineste clar ca o parte din web-site este un meniu. Si pe langa el au aparut o groaza de alte elemente care au inceput sa inlocuiasca functiile div-ului, asa ca importanta lui continua sa scada.

Dar in trecut, era unul dintre cele mai folosite elemente desi nu avea nici un rol. Acesta era foarte usor de folosit (si inca este) si aparea peste tot cand cineva vroia sa trateze cateva elemente impreuna.

Asa ca probabil div-ul inca este unul dintre cele mai intalnite elemente intr-o pagiona web.

**Cand ar trebui sa se foloseasca tag-ul <div>**

De exemplu, sa spunem ca vrem sa separam primele 2 paragrafe, pentru a le stiliza altfel. Asa ca le vom include intr-un div:

<div class="intro">

<h1>Elementul de tip div</h1>

<h2>Ce este si cand il folosim</h2>

</div>

<p>Elementul de tip div era pana de curand unul dintre cele mai des intalnite elemente intr-o pagina HTML.</p>

<p>Odata cu aparitia HTML 5, importanta lui a scazut, dar inca mai este foarte folosit pentru gruparea altor elemente in pagina.</p>

Acest <div> nu va schimba cu nimic modul in care este afisata pagina dar ne ajuta pe noi sa grupam cateva elemente impreuna (in acest caz primele doua paragrafe), pe care am vrea sa le tratam altfel.

Folosindu-ne de clasa intro putem stiliza cele doua paragrafe folosind CSS. Nu vom face asta acum, dar acesta este unul din rolurile acestui element.

Elementul <div> este unul de tip **block**, de aceea ar trebui folosit impreuna cu alte elemente de acelasi fel. Dupa cum am vazut mai devreme, cel mai adesea el este folosit pentru a grupa alte elemente block.

Nefiind un element de tip inline, el nu are cum sa apara in interiorul textului. Dar daca am vrea sa grupam doar cateva cuvinte impreuna putem folosi tag-ul <span>.

Tag-urile pentru Javascript si CSS

Cateva tag-uri care sunt foarte comune in partea de <head> a paginii

As vrea acum sa vorbim despre cateva tag-uri care se intalnesc foarte des in elementul <head></head>. Si anume cele pentru a incarca fisiere Javascript si CSS.

**Tag-ul pentru fisiere Javascript**

Pentru a incarca un cod Javascript intr-o pagina HTML vom folosi tag-ul: <script></script>

Continutul unui fisier extern Javascript poate fi incarcat folosind atributul src (care vine de la **source**):

<script src="un-fisier.js"></script>

In acest caz, browser-ul va cere fisierul de la acea adresa, il va citi si il va interpreta.

Indiferent de ce tag este folosit atributul src, acesta poate primi o cale in doua feluri:

* **absoluta**: prin care precizam exact adresa unde se afla fisierul: src="https://www.edumo.org/un-fisier.js", acesta include protocolul https/http, domeniul www.edumo.org si calea completa a fisierului /un-fisier.js
* **relativa**: daca fisierul Javascript se afla in acelasi folder cu fisierul HTML care il importa. Calea catre fisier va fi in functie fisierul HTML src="un-fisier.js"

Continutul unui fisier Javascript poate fi scris direct in pagina, intre cele doua tag-uri. In acest caz nu mai este nevoie sa precizam atributul src:

<script>

// codul javascript va fi scris direct aici

</script>

Veti vedea in unele pagini si atributul type folosit cu acest tag, dar acesta nu mai este cerut de nici un browser modern, pentru ca ele stiu acum cum sa interpreteze un astfel de fisier. Cel mai important atribut al tag-ului (si singurul) ramane src.

**Tag-ul pentru fisiere CSS**

Pentru a incarca fisiere CSS vom folosi tag-ul <link />. Acesta este un tag care nu se inchide iar atributul lui principal href primeste ca valoare calea catre fisierul pe care vrem sa il incarcam:

<link href="style.css" />

La fel ca in cazul atributului src folosit de tag-ul <script></script> acest atribut poate primi un link absolut (adresa completa a fisierului) sau unul relativ (in functie de fisierul HTML care il importa).

Si astfel browser-ul cere fisierul CSS de la acea adresa, il citeste si apoi il interpreteaza.

Alte atribute destul de intalnite ale acestui tag (dar oarecum optionale) sunt:

* type: acest atribut ii spune browser-ului ce tip de fisier va gasi la capatul link-ului.

Daca incarcam un fisier cu extensia **.css** el va presupune ca este un fisier de tipul "text/css", dar daca oferim un link catre un fisier fara o extensie ii poate veni un pic mai greu sa ghiceasca. Valoarea acestui atribut este valoarea [MIME type](https://developer.mozilla.org/en-US/docs/Glossary/MIME_type) al fisierului. Exemplu:

<link href="https://www.edumo.org/style.css" type="text/css" />

* rel: acest atribut ii spune browser-ului ce rol are acel fisier (rel vine de la **relationship**, ce legatura/rol are fisierul cu pagina).

De obicei va avea valoarea de "stylesheet", de exemplu:

<link href="https://www.edumo.org/style.css" type="text/css" rel="stylesheet" />

Mai exista o metoda de a folosi cod CSS, si anume de a-l scrie direct in pagina fara a-l incarca din alta parte. Pentru aceasta in HTML avem un tag special. Diferenta fata de tag-ul <link /> este ca acesta se inchide, iar tot continutul va fi pus intre aceste tag-uri:

<style>

/\* codul css va fi scris direct aici \*/

</style>

Concluzii

Asadar pentru Javascript avem un singur tag pentru importa un fisier in pagina sau pentru a scrie direct codul, si anume <script></script>.

Iar pentru CSS avem doua tag-uri: unul pentru a-l importa in pagina <link /> si unul pentru a-l scrie direct <style></style>

**<button>**

Unul dintre cele mai intalnite elemente intr-o pagina sunt butoanele. Dar nu toate dintre acestea sunt butoane propriu-zise ci sunt facute sa arate ca atare.

Un buton in HTML se creeaza astfel:

<button>Un buton</button>

Iar rezultatul este:

Un buton

Un buton are mai multe roluri printre care acela de a trimite un formular (atunci cand avem mai multe elemente de tip input grupate) sau acela de a permite alte actiuni (precum aceea de a accepta Termenii si conditii).

Butonul este un element destul de general si poate fi folosit pentru a crea interactivitate intr-un site prin faptul ca permite tot felul de actiuni.

Atribute

Pe langa atributele general de **id** si **class**, tag-ul mai accepta si:

Type

Acesta este cel mai important atribut pentru ca schimba cel mai mult comportamentul butonului. Ca valori, acest atribut poate primi:

* **submit**: aceasta este valoare prestabilita (default) si se foloste intr-un formular cand vrem sa il trimitem la server.

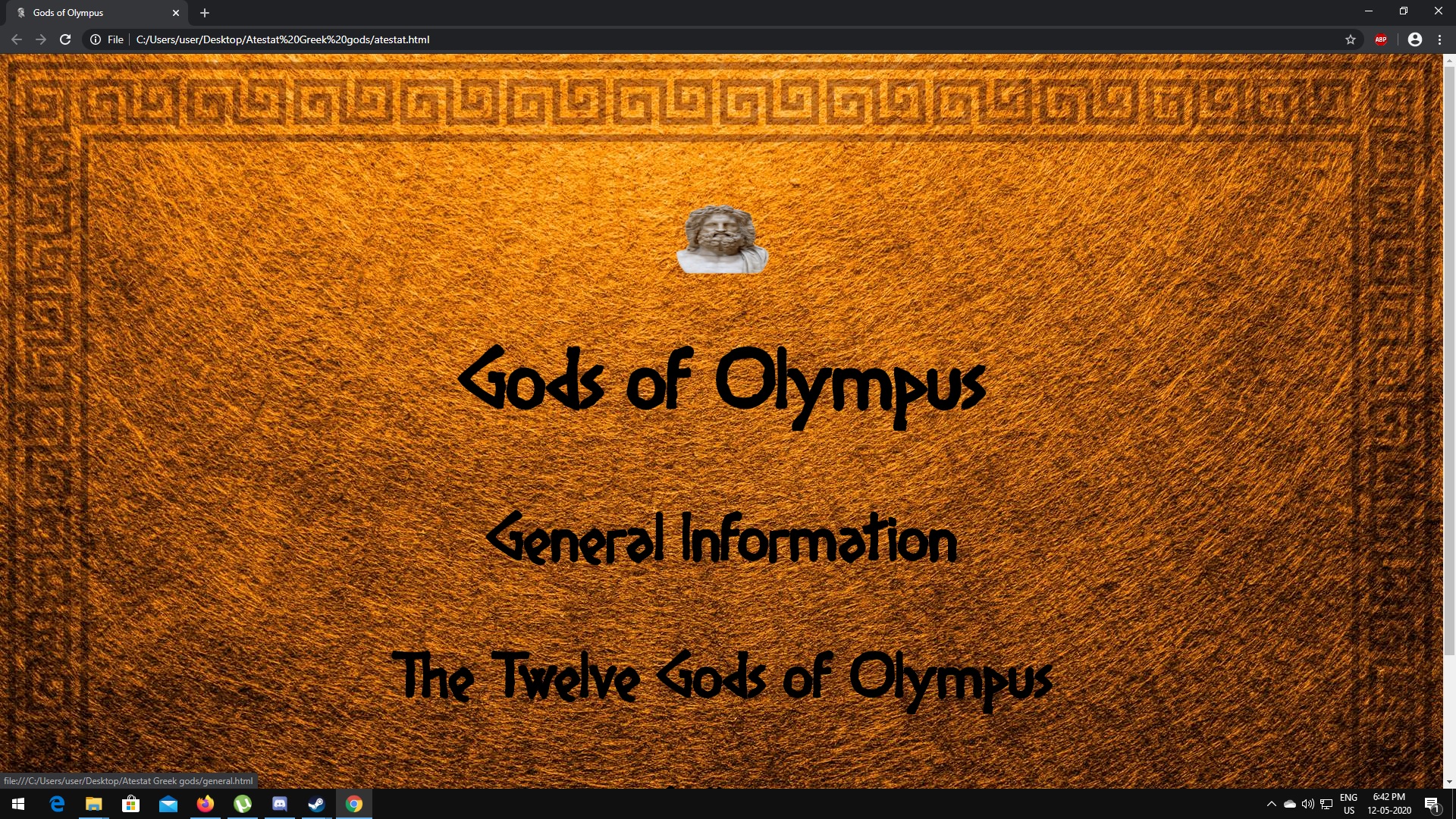
<button type="submit">Trimite</button>

* **reset**: un buton cu aceasta valoare pentru type se foloseste tot intr-un formular cand vrem ca cel ce-l completeaza sa stearga toate datele pe care le-a introdus. Nu este foarte des intalnit, dar folositor.

Acestea este cel mai intalnit atribut, lista completa o puteti citi pe [MDN](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/button).

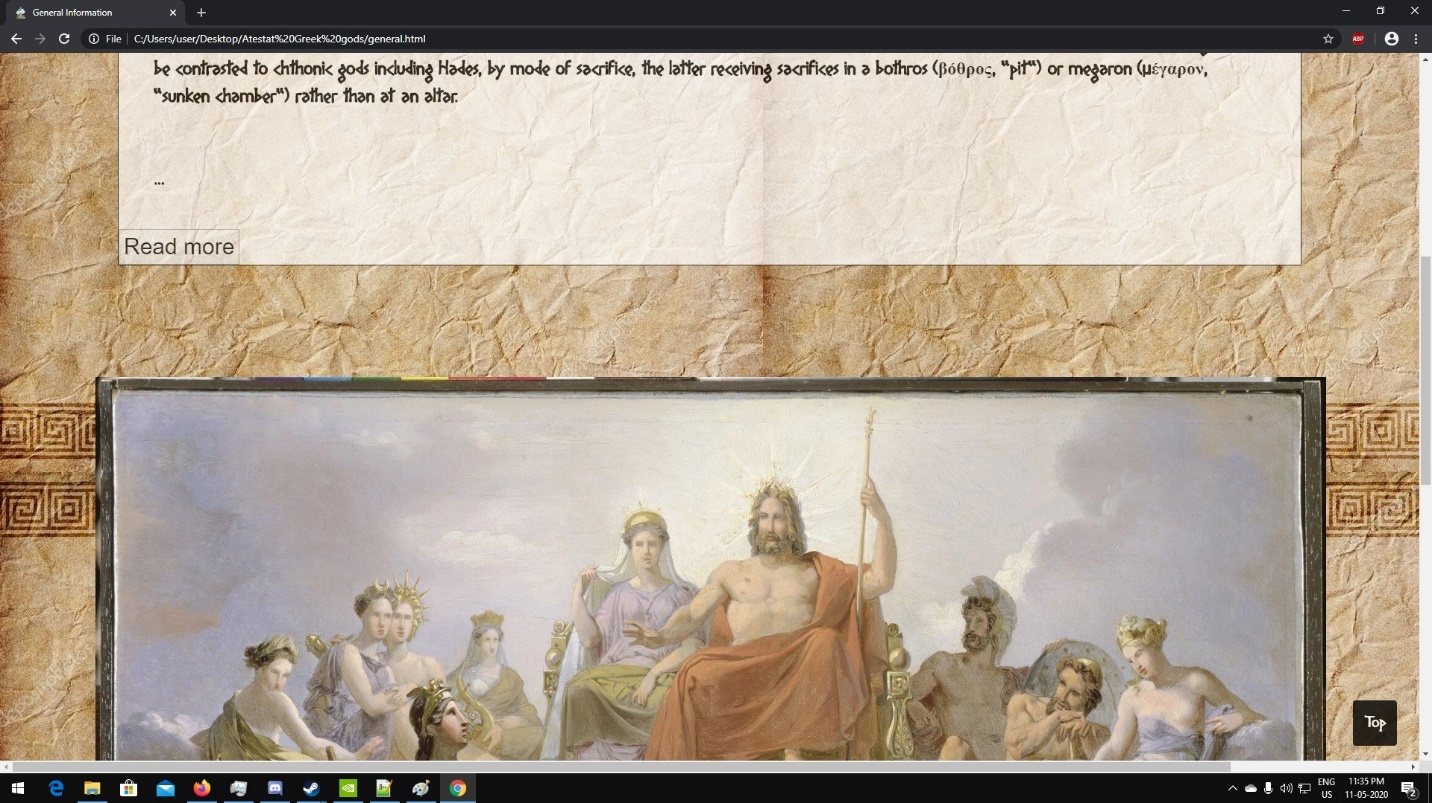
Asadar un buton cel mai des este intalnit intr-un formular, dar de fapt il putem folosi oriunde in pagina si folosind JavaScript ii putem schimba actiunea atunci cand se apasa pe el.

# **III. Descrierea Aplicatiei**

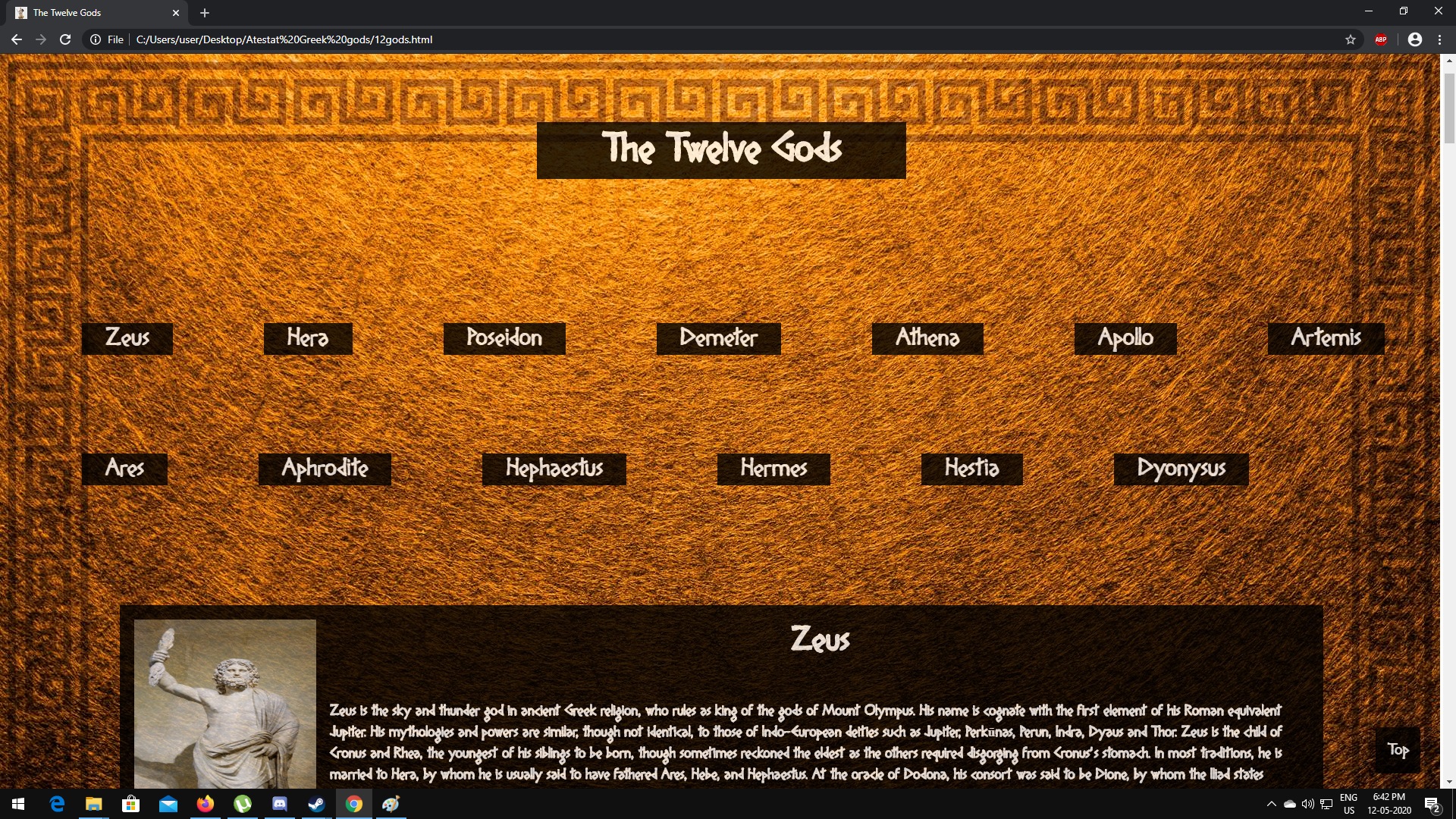


**Aceasta este pagina principala a site-ului in care exista posibilitatea de a alege urmatoarea pagina pe care doresti sa deschizi.**

**Daca apasati pe iconita care se afla deasupra de titlu,ve-ti fi acompaniat de o melodie greceasca traditionala**

****

**Aceasta pagina este General Information si ea ofera informatii generale legate de zeii care locuiau pe Muntele Olimp.**



**Pagina respectiva ofera informatii mai detaliate in legatura cu fiecare zeu in parte acompaniat de o statuie al zeului respectiv sau zeitei respective.**

****

**In aceasta sectiune sunt prezentate cateva imagini in care zeii au influentat arta in sine,fiind niste puncte de repere interesante si pline de mister**

# **IV.Bibliografie**

<https://en.wikipedia.org/wiki/Twelve_Olympians?fbclid=IwAR0ZSrE5H0fQeMf_Gs5J0ImmWgbx2ZpYzI2ZgtSbO3A3TBMJFBdb5_c_woU>

<https://en.wikipedia.org/wiki/Zeus?fbclid=IwAR34WJn6eGNFr13mRzrriACk-4QOpsOXGrZAVudpRDdsYFmBUeo1TcwClzU>

<https://en.wikipedia.org/wiki/Hera?fbclid=IwAR1GBRH5KSIEUVxebNlYG-QERnIdSGkO4zrQv4ct6AGnLyz6hZ5thT-aIWE>

<https://en.wikipedia.org/wiki/Poseidon?fbclid=IwAR0f5XY2aAX4Wu40fIk-paPlVAFJL_I6Y9inAhjIleVEbGkAwP8ArnUqePs>

<https://en.wikipedia.org/wiki/Demeter?fbclid=IwAR34WJn6eGNFr13mRzrriACk-4QOpsOXGrZAVudpRDdsYFmBUeo1TcwClzU>

<https://en.wikipedia.org/wiki/Athena?fbclid=IwAR1GBRH5KSIEUVxebNlYG-QERnIdSGkO4zrQv4ct6AGnLyz6hZ5thT-aIWE>

<https://en.wikipedia.org/wiki/Apollo?fbclid=IwAR3ZD5SQFecjDRduzs8fK_KfRvxB62hQ1ArCFFFFsXbvQNkJ1dluyteRG4w>

<https://en.wikipedia.org/wiki/Artemis?fbclid=IwAR3YmUJHiIJlZjMXbfFmlqRgK2B3Q581ol4aEO7uKgw0oDLDaqICrKrTREA>

<https://en.wikipedia.org/wiki/Ares?fbclid=IwAR3ZD5SQFecjDRduzs8fK_KfRvxB62hQ1ArCFFFFsXbvQNkJ1dluyteRG4w>

<https://en.wikipedia.org/wiki/Aphrodite?fbclid=IwAR1kL_f77NsKfzEfA5msW1d1NjAycVNG8d5wxB7WkilQQAXfMOG9uWqfc24>

<https://en.wikipedia.org/wiki/Hephaestus?fbclid=IwAR3gh8EML9Ww2vAgP4WEv5Qe2BmQK6XJMfIJC-bga2hrSbDD3Y-sGCsVxy8>

<https://en.wikipedia.org/wiki/Hermes?fbclid=IwAR3dM7rLsFU8Ikkojrs_GTLg0HaR7fh34hkhusp-nHSA2vNgoYxUpSmwoWI>

<https://en.wikipedia.org/wiki/Hestia?fbclid=IwAR2rlAGSS40uDeVGiNskSvdFY2bqf4dkxq_mtgvH7H4PegDhSQjmKLWQoxc>

<https://en.wikipedia.org/wiki/Dionysus?fbclid=IwAR2eW8JR3phhD9GvpI0ltugUCddG-dO6tfrk31A5YG4_a7hrefj9Vj1Odfg>