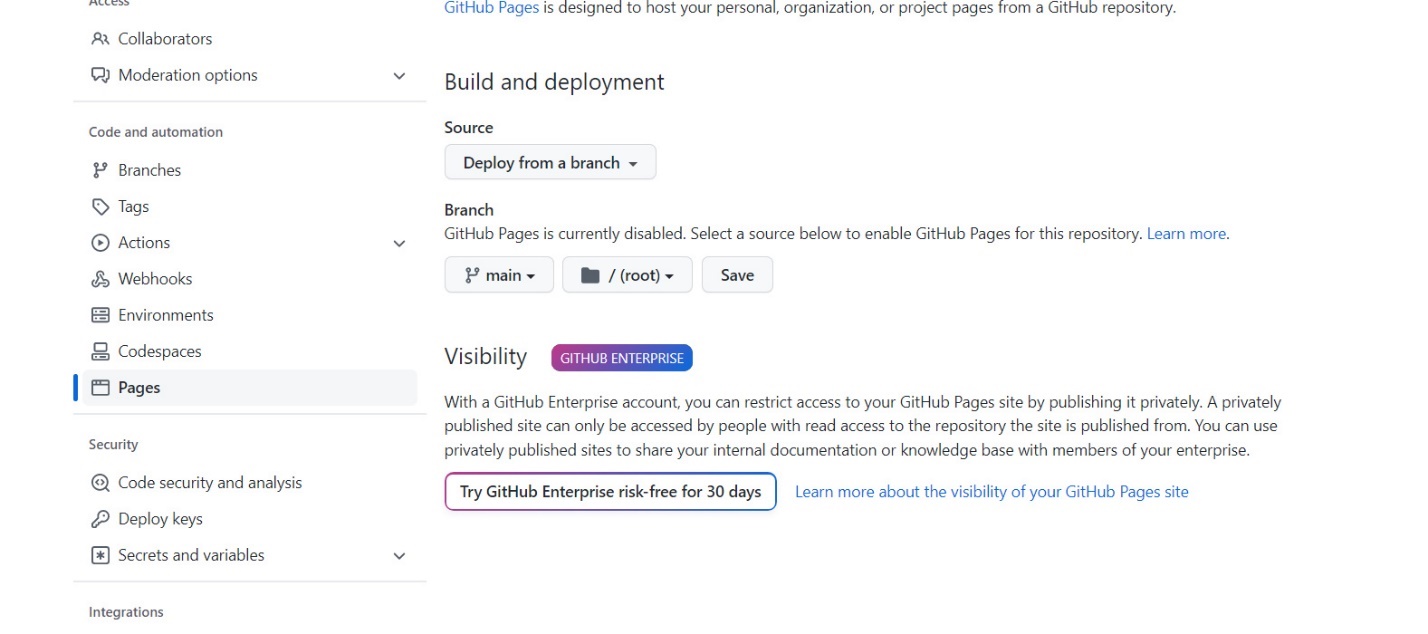
rxxxxxPentru a instala browser sync:

npm install -g browser-sync

**Host Site on GitHub**

1. Intai trebuie sa avem ceva continut in repo pentru a putea hosta un site
2. 
3. Pagina va avea adresa:

[My Page (kerbecs.github.io)](https://kerbecs.github.io/MyWeb/)

**Auto-Sync**

* Auto sync ne va ajuta prin aceea ca daca o resursa, de ex o pagina html deschisa in browser, se va modifica, nu va trebui sa dam manual reloa la pagina, ci se va da automat
* Pentru a o activa, dam cd la folderul cu paginile apoi deschidem CMD in admin mod si scriem:

browser-sync start --server --directory --files "\*.html"

--server – asta spune se va crea un local http server

--directory – spunem ca se va lua directory in care suntem acum cu cd

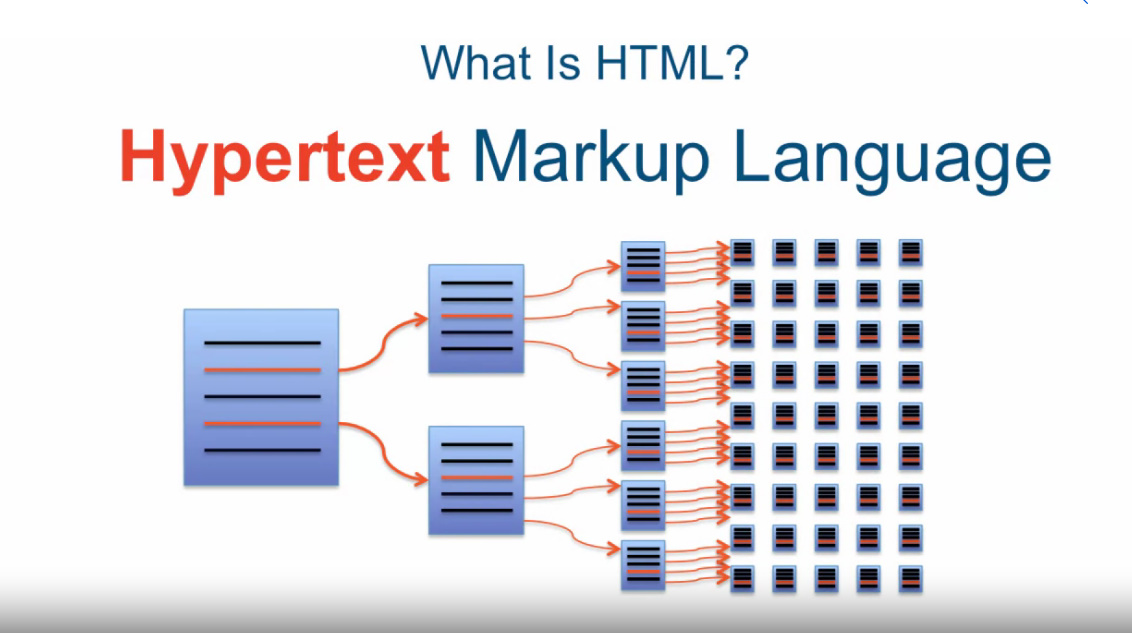
--files “\*.html” – spune ca orice tip de file .html va fi urmarit daca se fac schimbari

in el

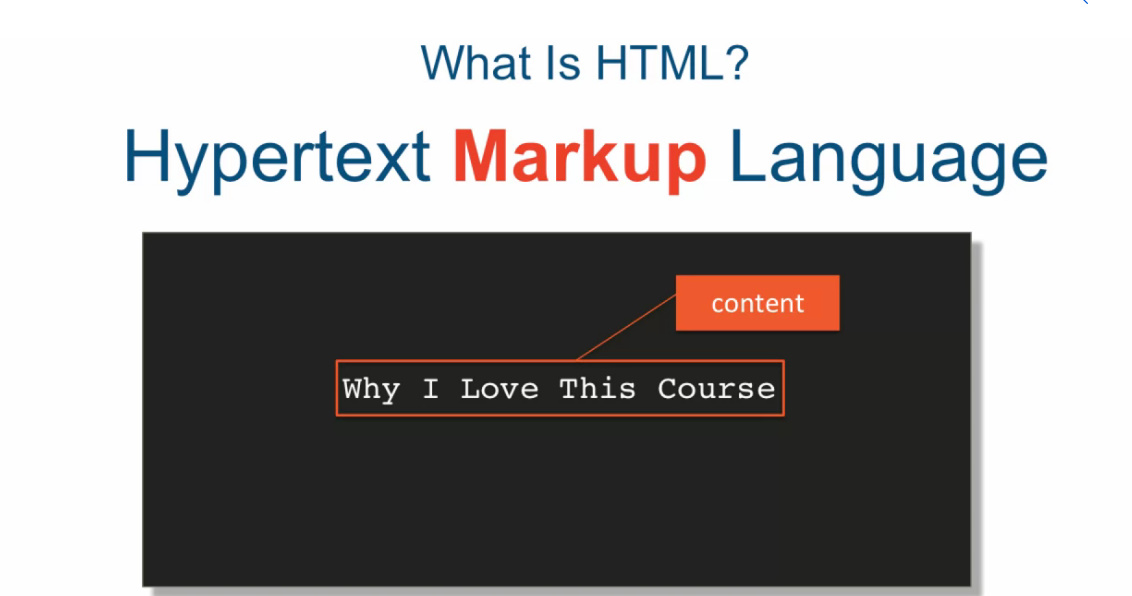
* Apoi pentru a descide un file, doar scriem /file.format la final de paina deschisa
* Serverul va rula dar cat e deschis cmd

**HTML Name**

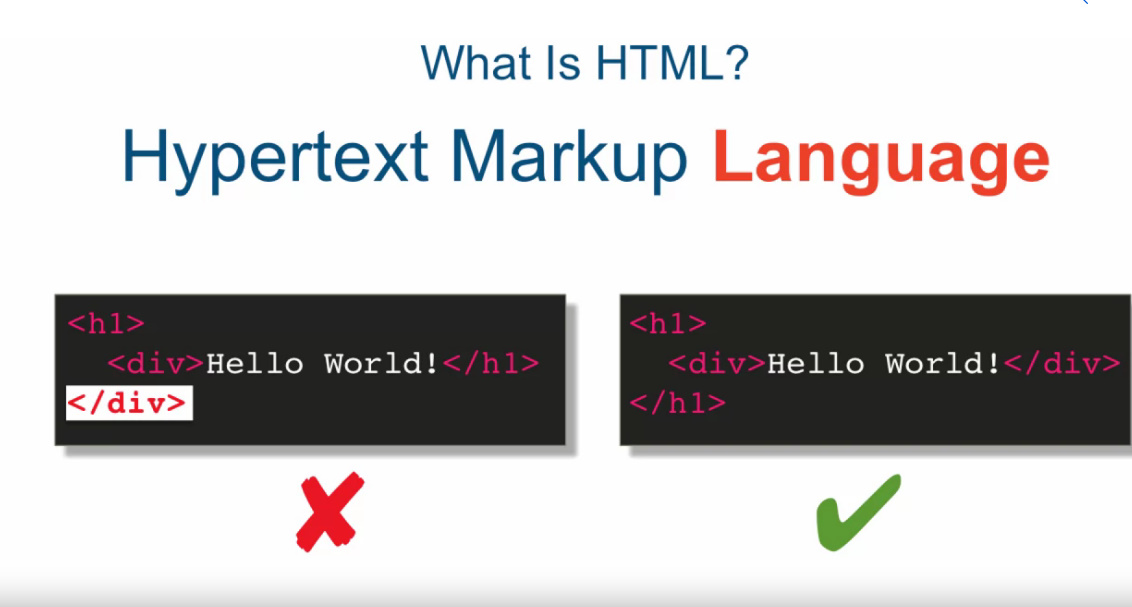
* **HTML** – HyperText MarkUp Language
* **HyperText** – text care contine linkuri catre alt text. Un document se refera la altul si acela tot la altul si tot asa. Nu se mai limiteza doar la text



* **Markup** – se traduce a anota. Se refera la continut(content) din document. Aume anotam cu HTML ce este continutul nostru pe care browserul sa il afiseze

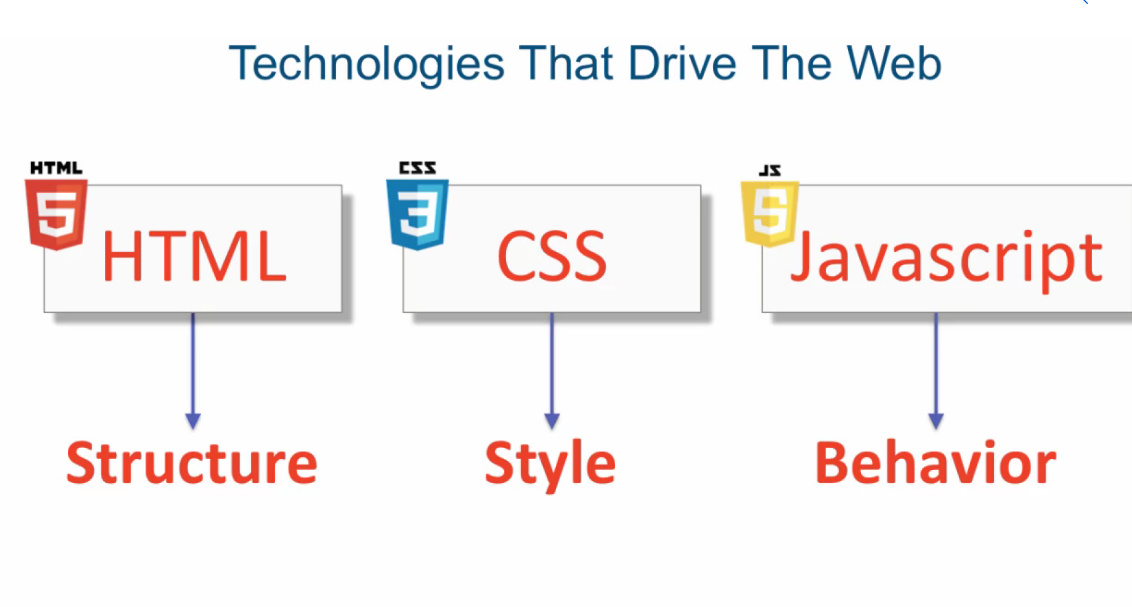
****

* **Language** – sintaxa proprie, semantica

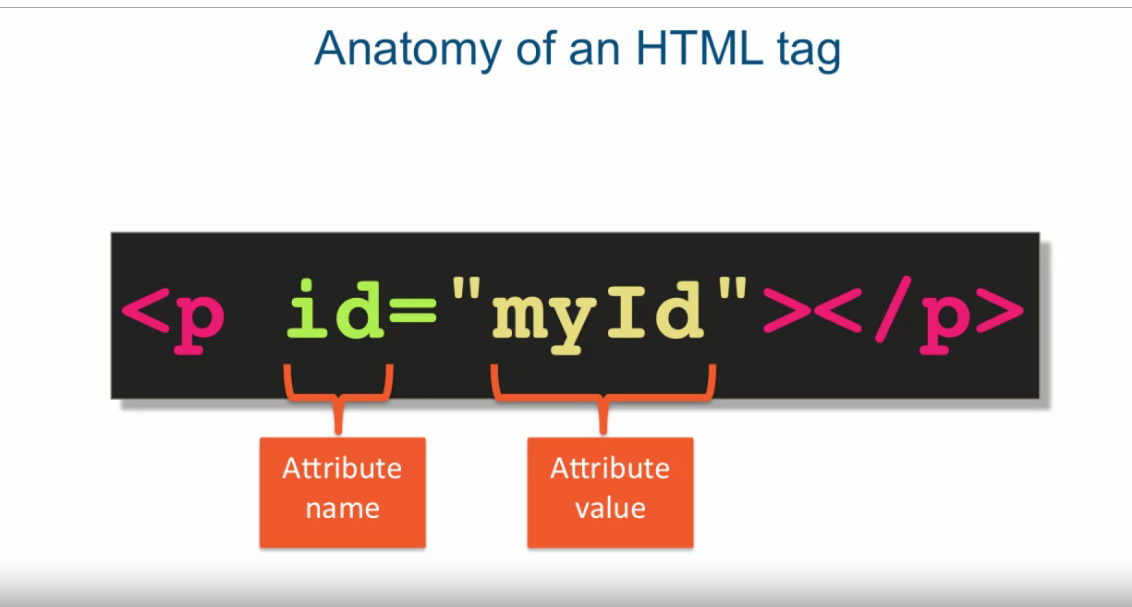
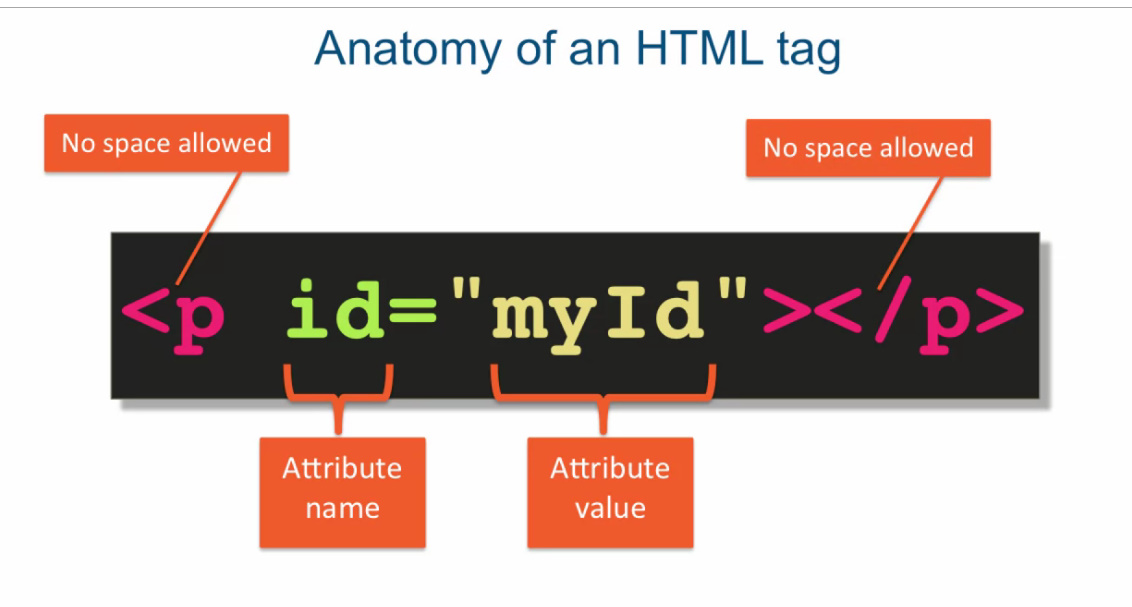
****

**HTML, CSS, JS**

* HTML – arata anume structura documentului, arata ce componente are documentul HTML. Nu ne spune cum arata componentele, dar numai care ele sunt.
* CSS – stilul, culoarea etc. Anume se ocupa de decorarea, stilarea
* JavaScript – ofera comportament, functionalitati



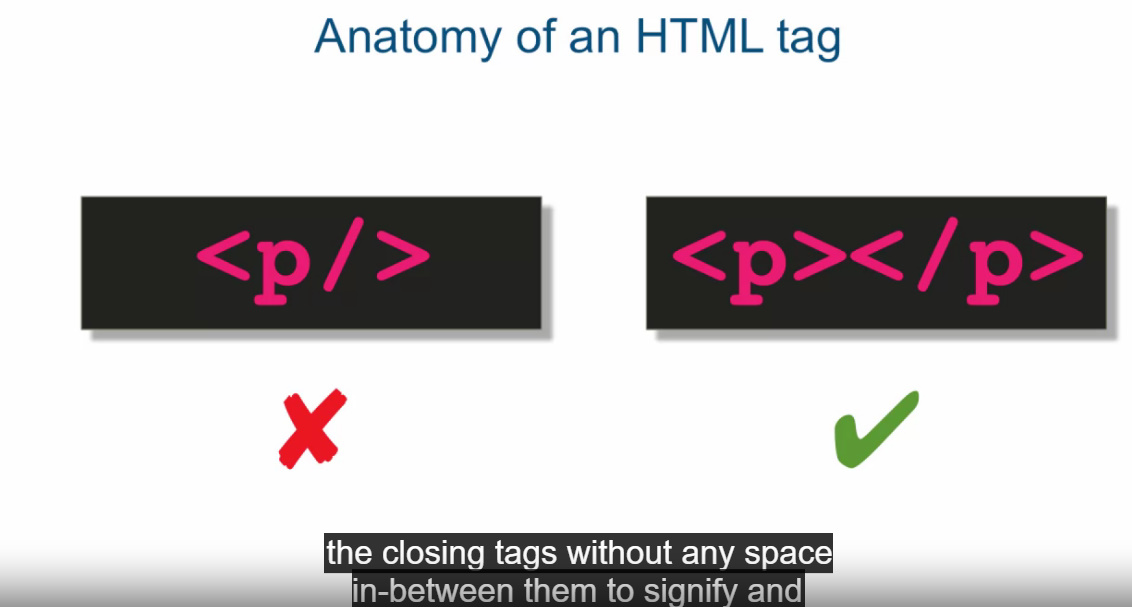
**Tag**

* Orice tag poate avea un id
* 
* Atributele sunt ca metadate
* Id creaza taguri unice, caci mai multe nu pot avea acelasi id
* 
* “” sau ‘’ nu sunt obligatorii, dar de dorit de folosit
* In “” putem folosi ‘ ‘ si intre ‘ ‘ putem “”.Cu care incepem, cu alea si terminam, gen



dar putem avea si **‘alert(“hi”)’**

* Unele elemente, care necesita tag de inceput si final, nu pot folosi doar tag de inchidere daca nu contin continut, de ex:



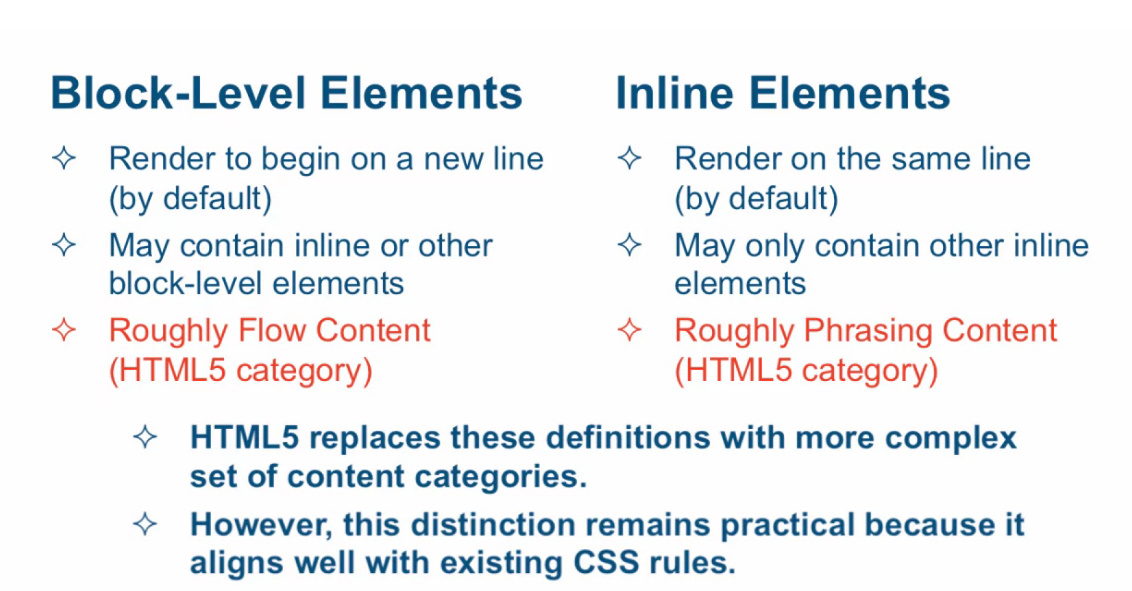
Daca folosim elemente ce nu au continut, cel mai bine e sa folosim un tag de deschidere si inchidere, dara fara nimic intre ele

**HTML DOC STRUCTURE**

* **<!doctype html>** - anume asta spune browserului sa trateze pagina noastra dupa standardele HTML. Candva nu exista HTML si fiecare site era implementant in felul lui. Fara acest tag, browserul va lua pagina ca una nestandarta cu HTML si ar putea aparea probleme
* **<head>** - descrie continutul paginii, metadata etc.
* **<title> este obligatoriu**
* Tagurile sunt executate rand pe rand de browser

**Block/Inline elements**

* Vechiul HTML avea doar aceste reguli:

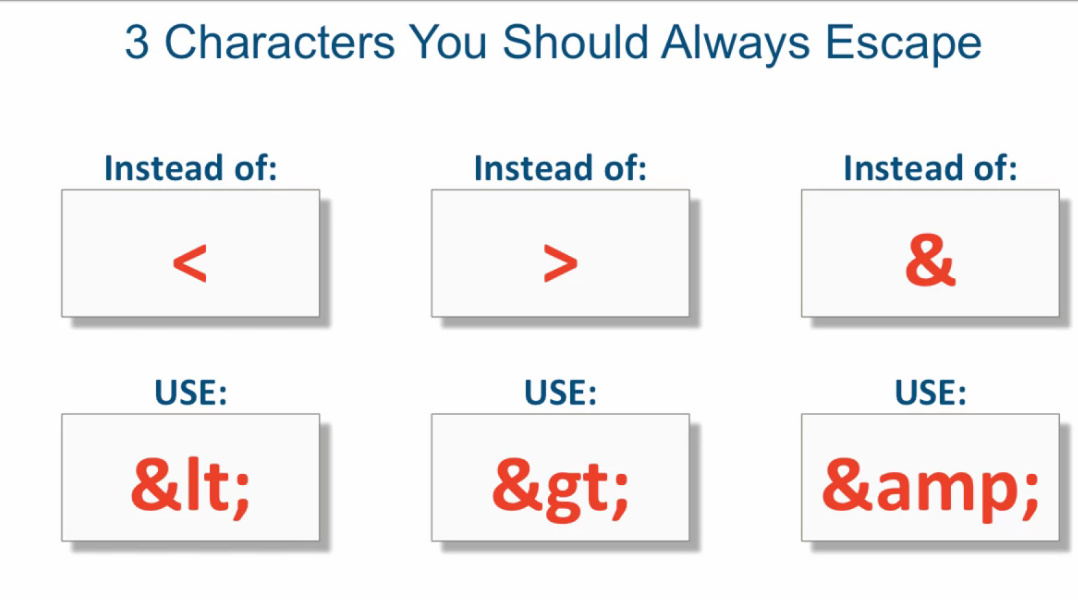


* HTML5 a decis sa le modifice, si sa le faca mai complexe, dar ele inca sunt folosite
* Block elements mereu au propria lor linie, mereu!
* **Nu punem block element in inline elements!!!**

**header, nav, aside, article,section**

* **Sunt block elements**
* **header –** cotine informatii de baza,ca logo, title
* **nav –** butoane de naviare, deobicei din header
* **section**
* **article** – in section
* **aside**

**Character entity reference**

****

**&nbsp**

* Multi folosesc &nbsp pentru a pune spatii libere, dar nu asta e rolul lui. Mai bine se foloseste <span> pentru asta
* rolul lui &nbsp este de a bloca punerea browserului a unui \n dupa elementele de dupa el,cand dimensiune paginii browserului e modificata de ex, asa incat ele sa nu se afle pe linii separate. Daca browserul va incerca sa puna \n dupa ele, atunci elementele de dupa &nbsp vor fi puse in noua linie iar \n pana la &nbsp

**<a>**

* <a href=”” title=””> - title va aparea in stanga jos can dam cu mouse pe link
* <a> este si un block element si inline. El nu incepe cu linie noua, si nici nu se termina cu linie noua, dar accepta in interiorul lui si block si inline elements, asa cum poate am vrea gen sa facem un paragraph ca link
* Inainte de HTML 5 era inline element, si asta putea crea probleme
* <a href=”#UnIdDeLaAltElement” > - deci referinta la id la alt element se face cu #, pentru a spune ca href nu are referinta la ceva din folder, dar la ceva din pagina chiar

**<img>**

* E important sa specificam width si height pentru imagine ,deoarece fara ele, cand pagina abea se va deschide, imaginea va fi incarcata putin mai tarziu, si elementele parca vor sari cand ea va fi incarcata
* In plus, daca imaginea nu poate fi incarcata, se pun un icon mic in loc de ea
* Cu width si height, browserul va crea deodata un spatiu cu dimensiunile specificate pentru imagine, ca se incarca ea sau nu.