

Programiranje za Internet

HTML

ak. god. 2023./24.

HTML

- HTML (*HyperText Markup Language*) je *markup* jezik koji se koristi za strukturiranje informacija za prikaz na strani web klijenta.
- *Markup* jezik je jezik koji kombinira sadržaj koji se objavljuje s dodatnim informacijama o tom sadržaju. Ti dodatni podaci mogu biti, npr. položaj sadržaja u pregledniku, organizacija paragrafa, naslovi i sl., i izražavaju se kao *markup*, izmiješan s osnovnim sadržajem dokumenta.
- Standardizacijom HTML-a se bavio W3 konzorcijum, a zadnja verzija HTML standarda koju je izdao W3C je [HTML 5.2](#). 2019. godine W3C je objavio da će razvoj HTML i DOM standarda nastaviti organizacija [WHATWG](#) (*Web Hypertext Application Technology Working Group*). WHATWG standardi se nazivaju "[Living Standard](#)".

Markup jezik

- Pojam *markup* je izveden iz tradicionalne izdavačke prakse "označavanja, obilježavanja" (*mark up*) rukopisa, kojima su se dodavale simboličke oznake za tiskanje na marginama rukopisa. To je kombiniranje sadržaja sa dodatnim oznakama tj. podacima o tom sadržaju koji se nazivaju *markup* i izmiješani su s osnovnim sadržajem dokumenta.
- Najpoznatiji *markup* jezik je HTML i koristi se za publiciranje informacija na webu.
- Markup jezici su i XML (*Extensible Markup Language*), XHTML (*eXtensible Hypertext Markup Language*), LaTeX,

Povijest HTML-a

- Tim Berners-Lee je postavljajući temelje weba uz HTTP i URI razvio i HTML.
- Širenje HTML-a dovelo je do postupka standardizacije. HTML 2.0, prva specifikacija HTML standarda, razvijen je pod vodstvom IETF (kraj 1995.).
- Razvojem HTML-a (od 1996. do 2019.) bavio se W3 konzorcijum.
- 2000. HTML je definiran i ISO standardom (ISO/IEC 15445:2000).

Verzija (W3C)	Godina
HTML 3.2	1997.
HTML 4.0	1998.
HTML 4.01	1999.
HTML 5	2014.
HTML 5.1	2016.
HTML 5.2	2017.

HTML standard

- Standard osigurava korišćenje HTML-a na različitim platformama i preglednicima.
- Svaka verzija HTML-a razvija se uz pokušaj postizanja što većeg konsenzusa među IT kompanijama.
- Postizanje konsenzusa osigurava interoperabilnost između različitih web preglednika i web servera, alata za razvoj web stranica, *developer* web sadržaja,
- HTML se stalno razvija zbog razvoja novih tehnologija (i softverskih i hardverskih).

HTML

- HTML dokument je tekstualna datoteka koja sadrži HTML kôd.
- Prazna mjesta i linije dozvoljeni su bilo gdje u HTML dokumentu i preglednici ih ignoriraju (preskaču) prilikom *parsiranja* HTML stranice.
- Može se promatrati kroz tri osnovna dijela:
 1. deklaracije tipa dokumenta,
 2. zaglavlja ograničenog elementom HEAD,
 3. tijela koje uključuje glavni sadržaj dokumenta, ograničenog BODY elementom ili FRAMESET elementom.

```
<!DOCTYPE HTML>
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Prva HTML stranica</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

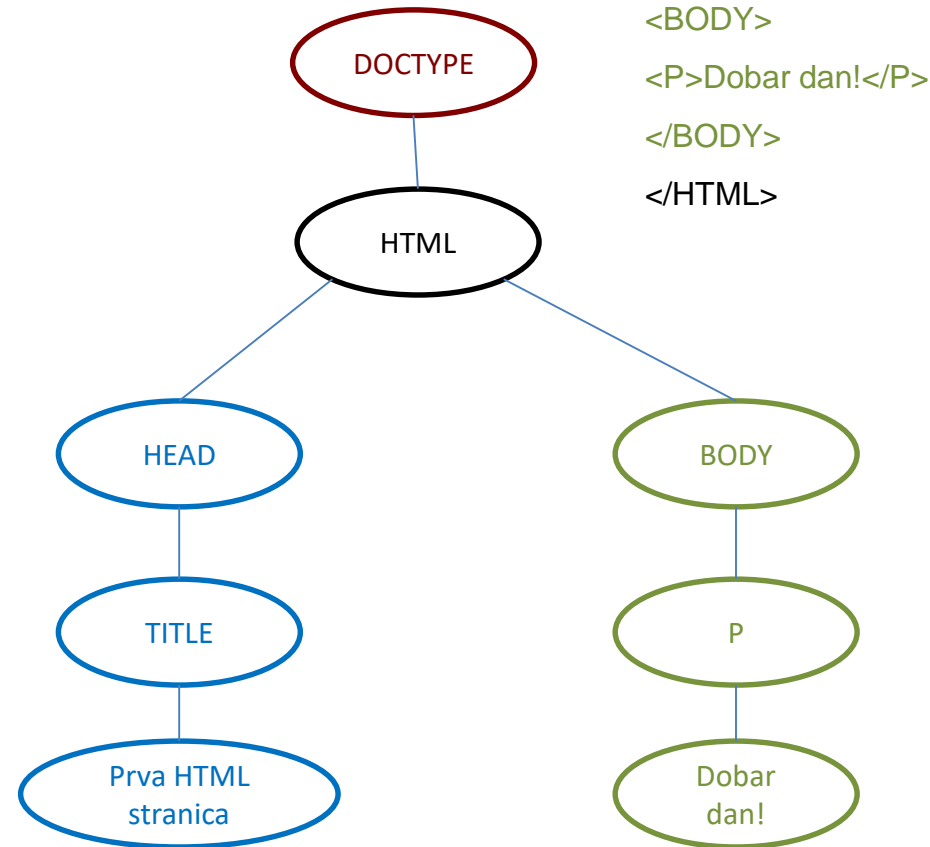
```
<P>Dobar dan!</P>
```

```
</BODY>
```

```
</HTML>
```

HTML

- HTML tekstualna datoteka se nakon dohvaćanja u pregledniku najprije parsira.
- Parsiranje ili sintaktička analiza je postupak ocjene da li je kolekcija tokena/stringova napisanih u određenom jeziku u skladu s gramatičkim pravilima jezika.
- Tokeni su nizovi znakova smislenog značenja dobiveni leksičkom analizom teksta i organiziraju se u gramatički ispravnu strukturu u formi stabla.

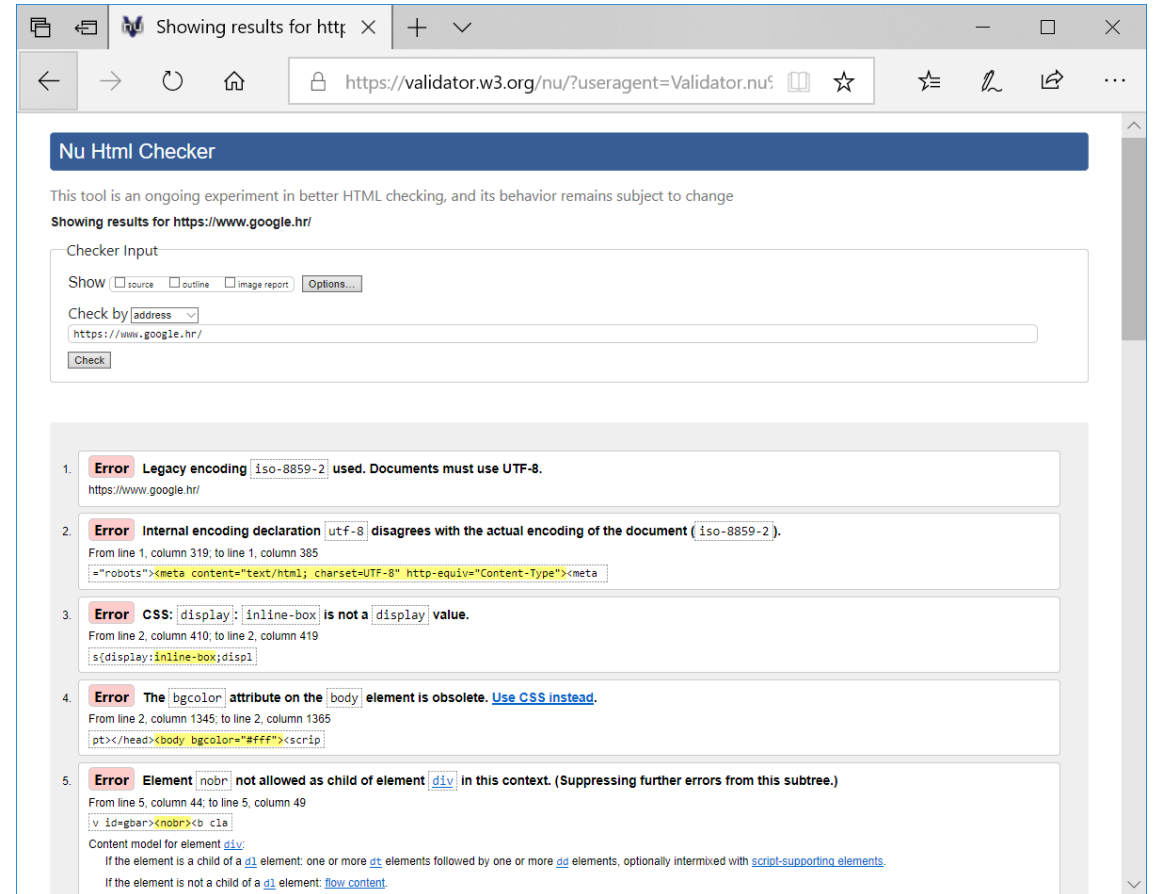


Deklaracija tipa dokumenta

- [Deklaracija tipa dokumenta](#) na početku HTML stranice kaže pregledniku kojim jezikom je dokument napisan. Oznaka tipa dokumenta html (<!DOCTYPE html>) se koristi za HTML 5.
- Ukoliko nije navedena, a trebala bi biti, softver koji prikazuje web stranice, a zove se mehanizam prikaza je obično realiziran vrlo robusno i može kompenzirati odstupanja od HTML standarda te će preglednik sam "zaključiti" o kojem se jeziku radi.
- Prije verzije 5, starije verzije HTML-a, koje su bile definirane [SGML \(Standard Generalized Markup Language\)](#) jezikom, za definiciju tipa dokumenta koristili su dokument napisan u jednom od XML shema jezika (XML Schema (W3C), RELAX NG) (npr. <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "<http://www.w3.org/TR/html4/strict.dtd>">) koji je sadržavao pravila HTML jezika (dopuštene elemente, attribute, moguće vrijednosti sadržaja, ...).

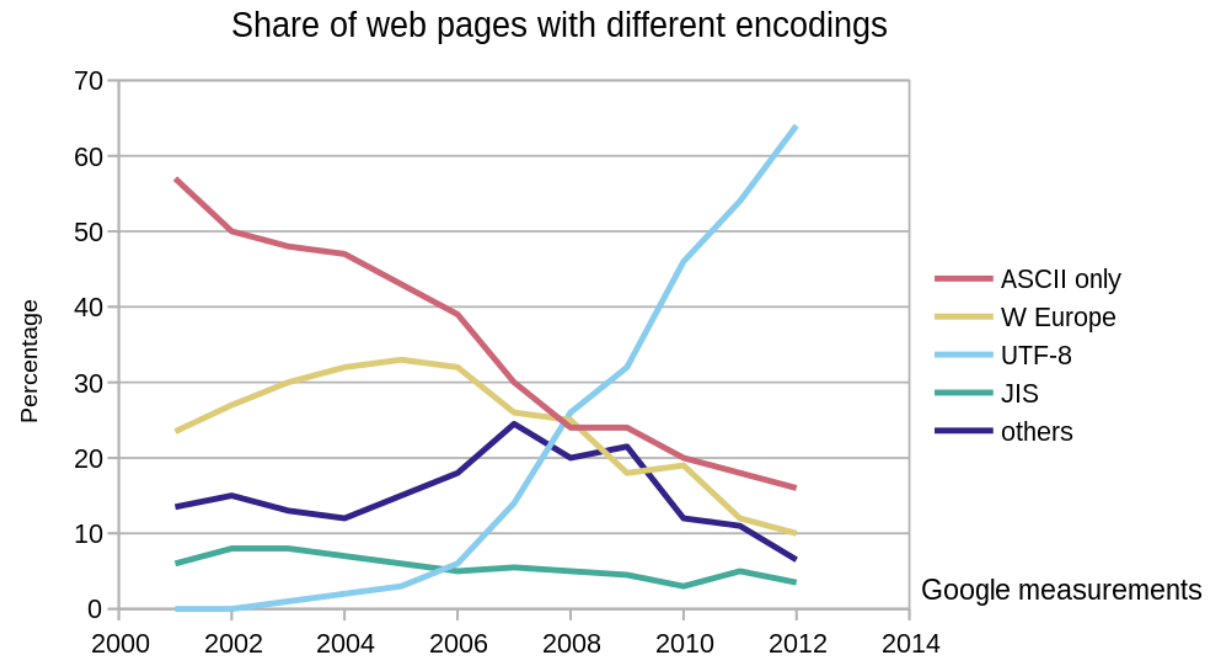
HTML

- Mehanizmi prikaza (softver koji čita i prikazuje web stranice na ekranu) dosta toleriraju greške i nepoštivanje HTML specifikacije te ih uspijevaju kompenzirati i uredno prikazati i ne tako uredan HTML kôd.
- Provjera validacije (ispravnosti) HTML dokumenta može se raditi *online* w3c alatom koji se nalazi na adresi <https://validator.w3.org/>
- HTML dokument je validan ako poštuje strukturu i ostala pravila definirana specifikacijom.



Karakteristi

- HTML je tekstualni dokument. Za tekstualne datoteke je potrebno definirati kodiranje kojim će se tekstualni sadržaj prevesti u binarni zapis i obrnuto da bi se tekstualni dokument mogao ispravno pročitati.
- Kodiranje (*encoding*) je postupak pretvaranja jednog formata zapisa u drugi.
- Svi podaci su u binarnom formatu, ali kada kažemo da su podaci tekstualni, to znači da se binarni zapis može prikazati kao tekstualni tj. kao niz karaktera korištenjem određenog skupa karaktera (*character set*) kojim se niz bitova interpretira kao određeno slovo.



<https://en.wikipedia.org/wiki/File:Utf8webgrowth.svg>

HTML karakteri

- U početku weba se koristio ASCII skup karakter.
- Od HTML 2.0 do HTML 4.01 predefinirani skup karaktera je [ISO-8859-1](#) (ISO-8859-2 za hrvatske karaktere), ali HTML 4 ima podršku i za UTF-8, koji se navede u meta tagu.
- Oznake skupova karaktera definirane su [MIME standardom](#).
- HTML 5 predefinirano koristi [UTF-8](#) karaktere.
- UTF-8 je riješio većinu problema s podrškom za različite karaktere.
- HTML 5 ima podršku i za ISO-8859.



```
<meta http-equiv="Content-Type"
content="text/html;charset=ISO-8859-1">
```

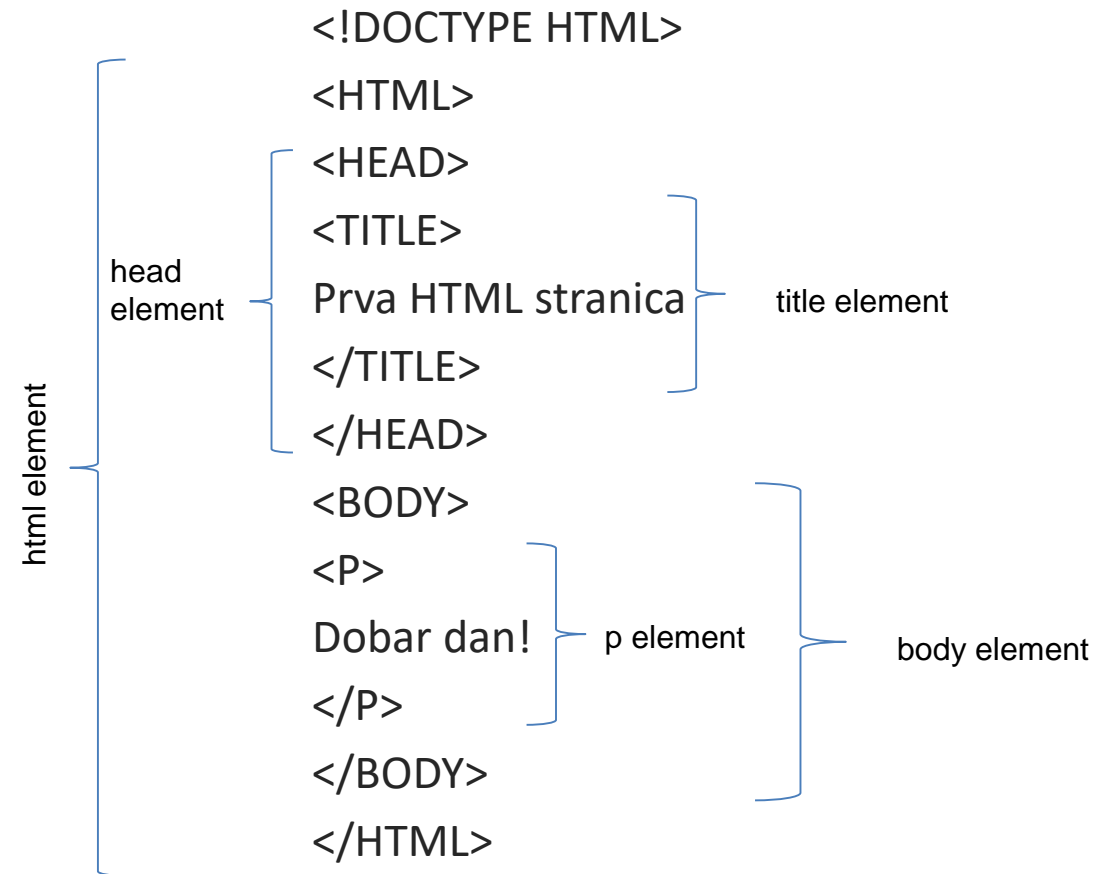
```
<meta http-equiv="Content-Type"
content="text/html;charset=UTF-8">
```

```
<meta charset="UTF-8">
```

```
<meta charset="ISO-8859-1">
```

HTML elementi

- Osnovni građevni blokovi HTML-a su HTML [elementi](#).
- Elementi se označavaju oznakama koje se zovu tagovi.
- Obično element ima početnu i krajnju oznaku (*start, opening tag - end, closing tag*). Npr. `<html></html>`, `<form></form>`,...
- Tagovi označavaju početak i kraj HTML elemenata.
- Element je ustvari sve ono što se nalazi između početnog i krajnjeg taga uključujući i same tagove.

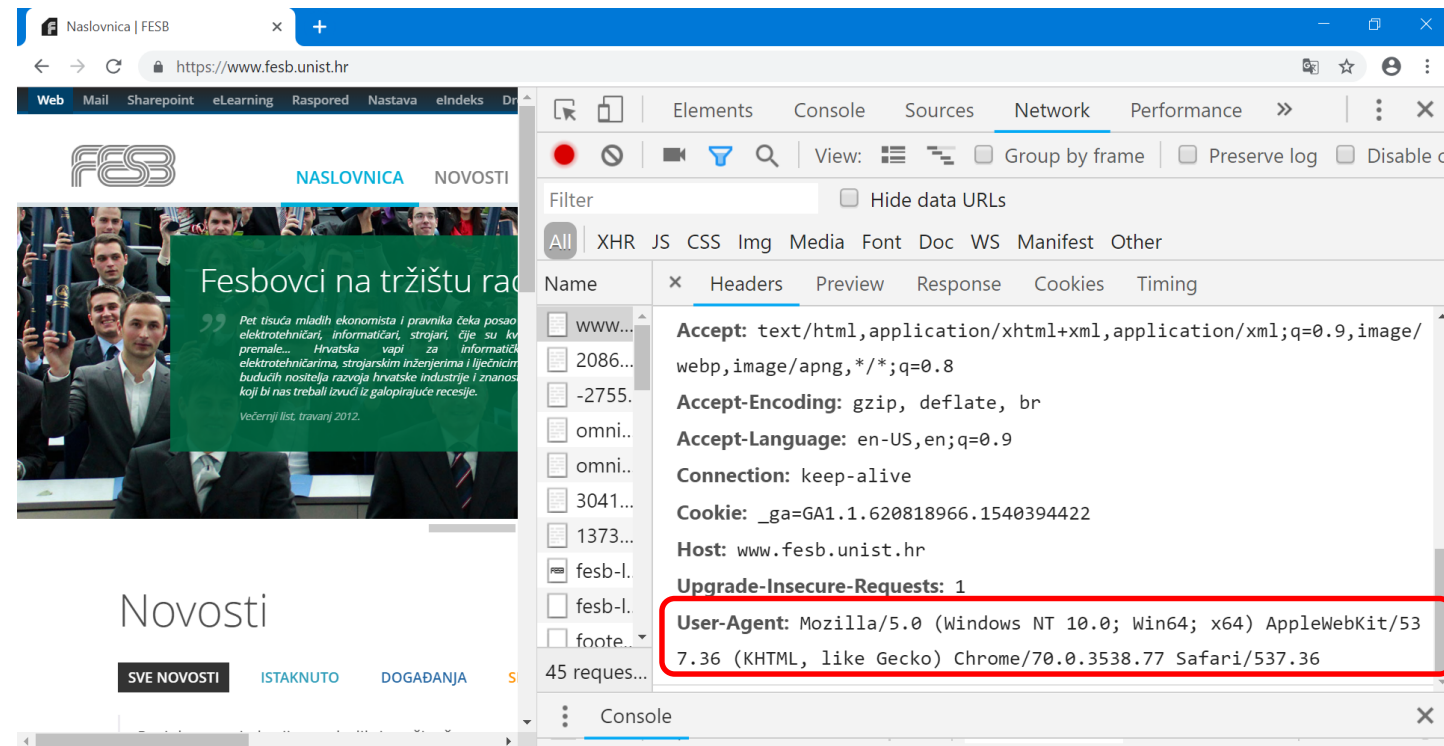


HTML elementi

- Prazni elementi koji između početnog i krajnjeg taga nemaju sadržaja ne trebaju krajnju oznaku (npr.
).
- Moguće je također koristiti i sintaksu samo-zatvarajućeg (*self-closing*) taga za prazne elemente (
) koja je definirana u XHTML-u.
- [XHTML \(*Extensible HyperText Markup Language*\)](#) je standard W3 konzorcijuma (2001.) kojim je specifikacija HTML 4.01 kombinirana sa sintaksom [XML-a \(*eXtensible Markup Language*\)](#). XHTML je postavio stroža sintaktička pravila no HTML u svrhu smanjenja opterećenje parsera i korisničkih agenata.
- [Korisnički agent](#) je softver koji djeluje u ime i za račun korisnika, u slučaju weba to je preglednik.

HTML

- HTTP protokol definira zaglavlje User-Agent koje preglednik šalje web serveru u sklopu HTTP zahtjeva kako bi se identificirao.
- Server može na osnovu te informacije prilagođavati sadržaj koji će isporučiti web klijentu obzirom na mogućnosti preglednika identificiranog stringom iz zaglavlja User-Agent.
- Google je predložio novi standard skupa zaglavlja [HTTP Client Hints](#) koja bi trebala zamijeniti User-Agent zaglavlje.



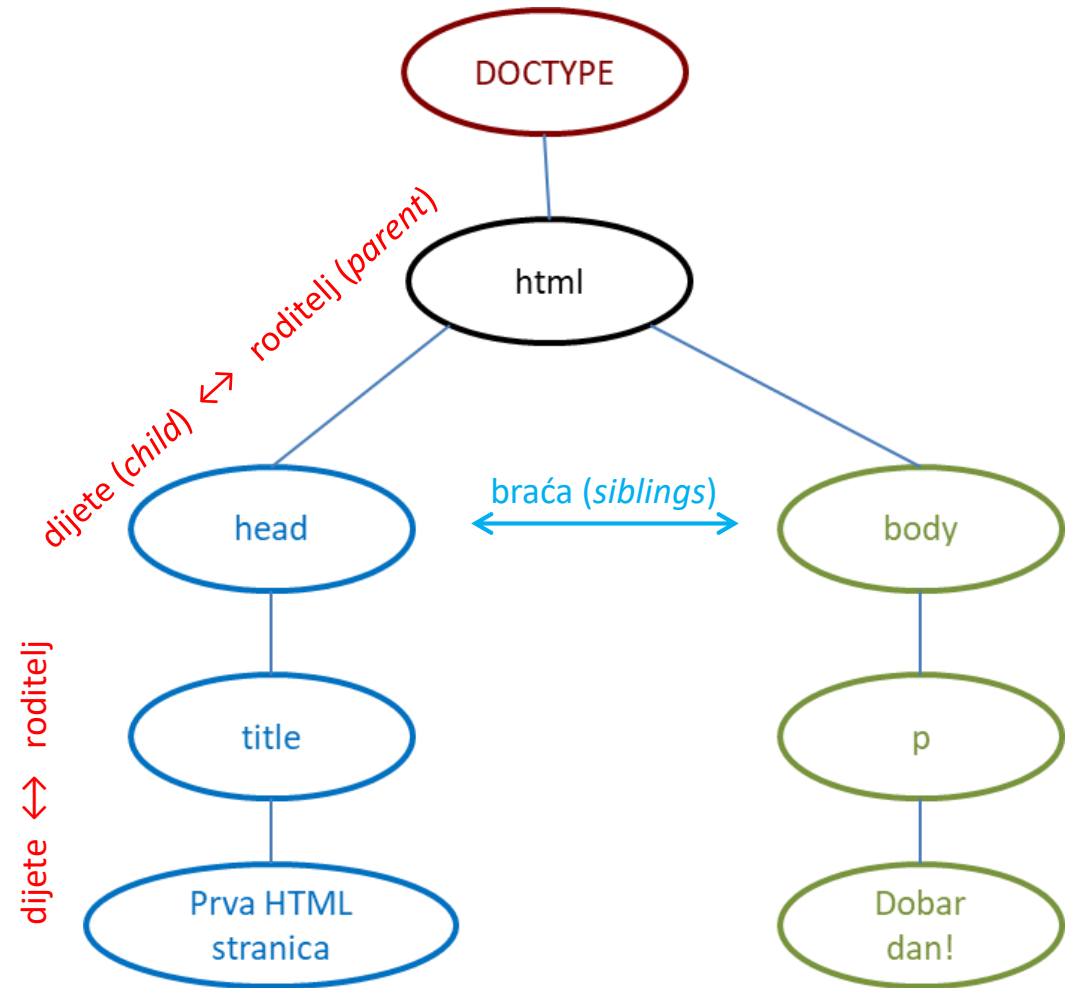
HTML

- HTML jezik nije osjetljiv na velika i mala slova (*case-insensitive*) pa su oznake <html> i <HTML> identične.
- Razlog je olakšavanje učenja i pisanja HTML-a, ali što je jezik fleksibilniji, parsiranje je zahtjevnije, a i preglednost dokumenta može biti manja.
- Preporučljivo je držati se jednog stila unutar jednog HTML dokumenta, a dosta autora preporuča XHTML stil. Imena elemenata i atributa u XML-u su *case-sensitive* pa XHTML definira da se koriste samo mala slova za imena elemenata i atributa.

```
<!DOCTYPE html>
<html>
<head>
<title>Prva HTML stranica</title>
</head>
<body>
<p>Dobar dan!</p>
</body>
</html>
```

HTML

- HTML elementi mogu imati sadržaj.
- Sadržaj je sve ono što se stavlja između početnog i krajnjeg taga. To uključuje i druge HTML elemente, ali i drugi sadržaj poput teksta i sl.
- Struktura HTML dokumenta je HIJERARHIJSKA i sastoji se od elemenata koji su međusobno u hijerarhijskim odnosima koji su definirani položajem elementa u dokumentu. Ugniježđeni element je dijete od elementa roditelja unutar čijih tagova je naveden.



HTML atributi

- HTML elementi mogu imati [atribute](#). Atributi su dodatne informacije o svojstvima elementa.
- Atributi se navode uvijek u početnom tagu elementa. Obično su definirani u obliku parova ime_atributa = "vrijednost_atributa", ali neki atributi nemaju vrijednosti (npr. `<p hidden>Skriveni paragraf</p>`).
- Vrijednosti atributa su stringovi. Ukoliko nije drugačije specificirano, vrijednosti mogu biti bilo kakvi stringovi, uključujući i prazni string.
- HTML je *case-insensitive* i za attribute pa možete pisati `` i ``, ali je opet, u skladu s XHTML sintaksom, preporuka korištenja malih slova za imena atributa.
- Također je preporuka korištenje navodnika, uobičajeno dvostrukih, ali može i jednostrukih, za navođenje vrijednosti atributa (`<a href="https://www.google.hr"` , ``).

HTML atributi

- Različiti HTML elementi koriste različiti attribute.
- Specifikacija HTML elemenata sadrži i attribute podržane na pojedinom elementu.
- Listu atributa te elemenata na kojima su podržani možete dohvatiti na https://www.w3schools.com/tags/ref_attributes.asp.
- Globalni atributi su podržani na svim HTML elementima.
- Skup atributa (*onclick*, *onmousemove*, ...) koji omogućava povezivanje s JavaScript funkcijama za obradu događaja (*event handler content attribute*) su također globalni atributi. Ime tih atributa odgovara imenu rukovatelju događaja koje je definirano DOM (*Document Object Model*) *event* standardom. DOM definira sučelja između HTML elemenata i programskog koda.
- Atribut predstavlja referencu na JavaScript funkciju koja će se izvršiti kada se desi promatrani događaj.

HTML atributi

- Atributi za povezivanje s funkcijama za obradu događaja uvijek imaju isto ime kao i rukovatelji događaja čije ime uvijek počinje s prefiksom "on" nakon kojeg slijedi ime samog događaja.
- Npr. za događaj pritiska tipke miša tj. *mousedown*, rukovatelj događaja se zove *onmousedown*, a tako se zove i atribut HTML elemenata za povezivanje s funkcijom za obradu događaja klika miša.

```
<!doctype html>
<html>
<script type="text/javascript">
function spustenmis()
{
    alert("Pritisnili ste botun miša na p");
}
</script>
<body>
<p onmousedown="spustenmis()">Klikni
me!</p>
</body></html>
```

HTML

- Globalni atribut *class* sadrži imena klasa kojima element pripada odvojena razmakom.
- Obično se koristi za povezivanje s klasama CSS (*Cascading Style Sheets*) stilova, ali može se koristiti i za identificiranje skupa elemenata iste klase za korištenje u JavaScript kodu.

```
<!doctype html>
<html><body>
<p class="jedan dva">P element</p>
<div class="jedan">Div element</div>
<p class="dva">Drugi P element</div>

<script type="text/javascript">
var klasa_jedan = document.getElementsByClassName("jedan");
var klasa_dva = document.getElementsByClassName("dva");
var klasa_jedan_dva = document.getElementsByClassName("jedan
dva");
console.log(klasa_jedan);
console.log(klasa_dva);
console.log(klasa_jedan_dva);
</script>

</body> </html>
```

[Primjer](#)

HTML atributi

- Za razliku od *class* atributa koji se koristi za identificiranje skupine HTML elemenata, *id* atribut se koristi za identificiranje točno jednog elementa te svaki *id* atribut treba imati jedinstvenu vrijednost. (<p id="jedinstveni_P">)
- Vrijednost *id* atributa mora imati barem jedan karakter, ne smije sadržavati razmake, ali druga ograničenja na vrijednost *id* atributa nisu postavljena. Može sadržavati brojeve, interpunkcijske znakove i druge specijalne karaktere.
- Jedinstvena identifikacija pojedinačnih elemenata u HTML dokumentu može se koristiti u različite svrhe, a najčešće se koristi za postavljanje jedinstvenog stila na elementu ili dohvaćanje jedinstvenog elementa u JS kodu.

HTML atributi

- [Podatkovni atributi](#) sadrže specifične podatke koji se odnose na tu stranicu tj. aplikaciju. Korisnički agent u potpunosti ignorira ove attribute. Koriste se za uključivanje atributa same aplikacije, podataka aplikacije uz HTML elemente.
- Podatkovni atributi se sastoje od imena atributa i string vrijednosti koja može biti bilo što. Ime podatkovnih atributa započinje sa stringom "data-" nakon čega ide barem jedan karakter, bez znaka : (dvotočke) i bez velikih ASCII slova.
- Svaki HTML element može imati proizvoljno mnogo podatkovnih atributa koji se obično koriste u popratnom JS kodu (`<button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1">`).

HTML

- Određeni karakteri imaju specijalnu svrhu u HTML jeziku (npr. < >) pa se moraju u dokumentu zamijeniti s tzv. karakterima entiteta (*character entities*) ako se koriste kao "obični" karakteri.
- Karakteri entiteta se označavaju s:
 1. `&entity_name;`
 2. `&#entity_number;` } ILI
- Korisnički agent treba zamijeniti karaktere entiteta s "pravim" karakterima prilikom prikazivanja HTML dokumenta. Specijalni karakteri koji se obično ne nalaze na tipkovnicama (npr. znak za euro) se također navode karakterima entiteta.

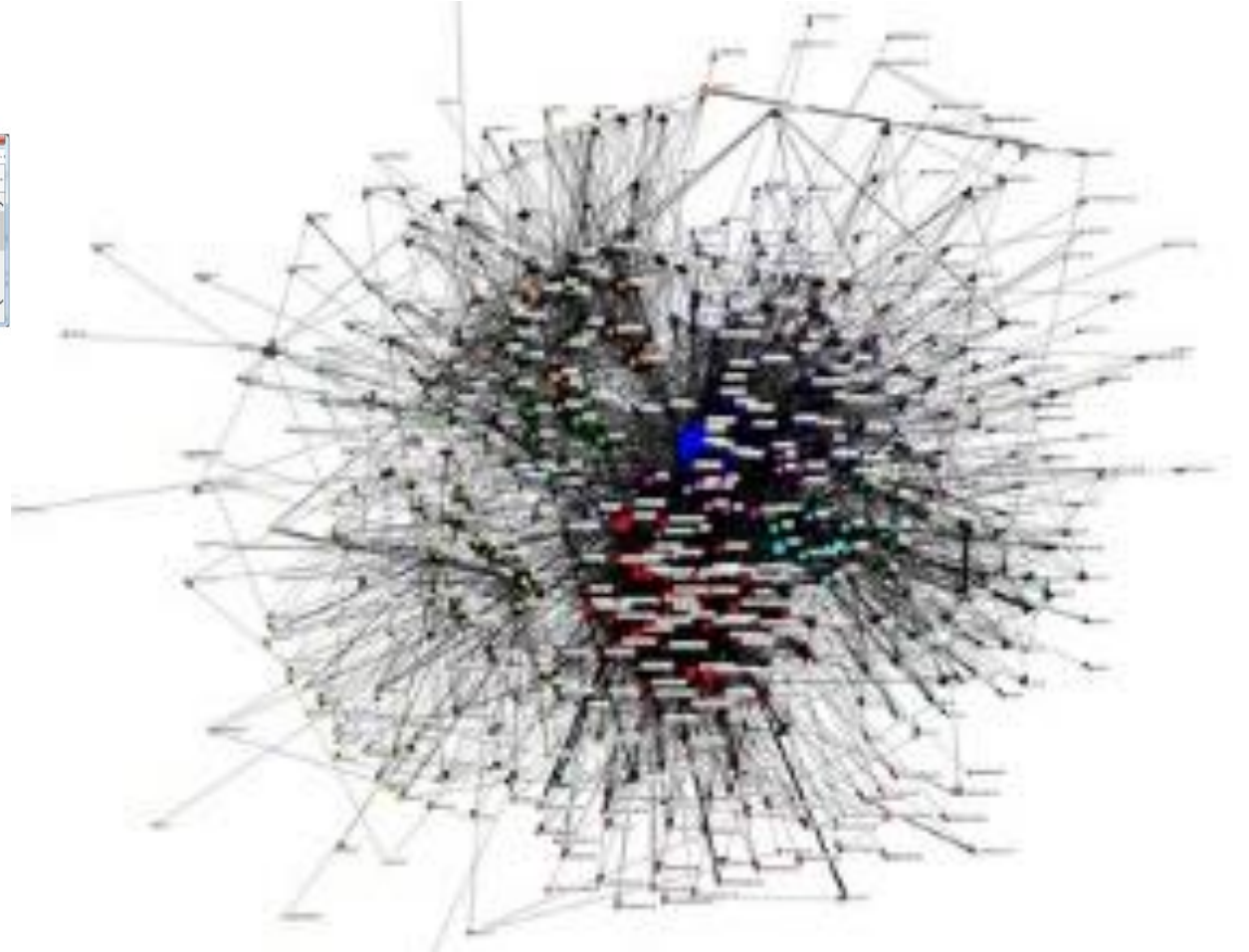
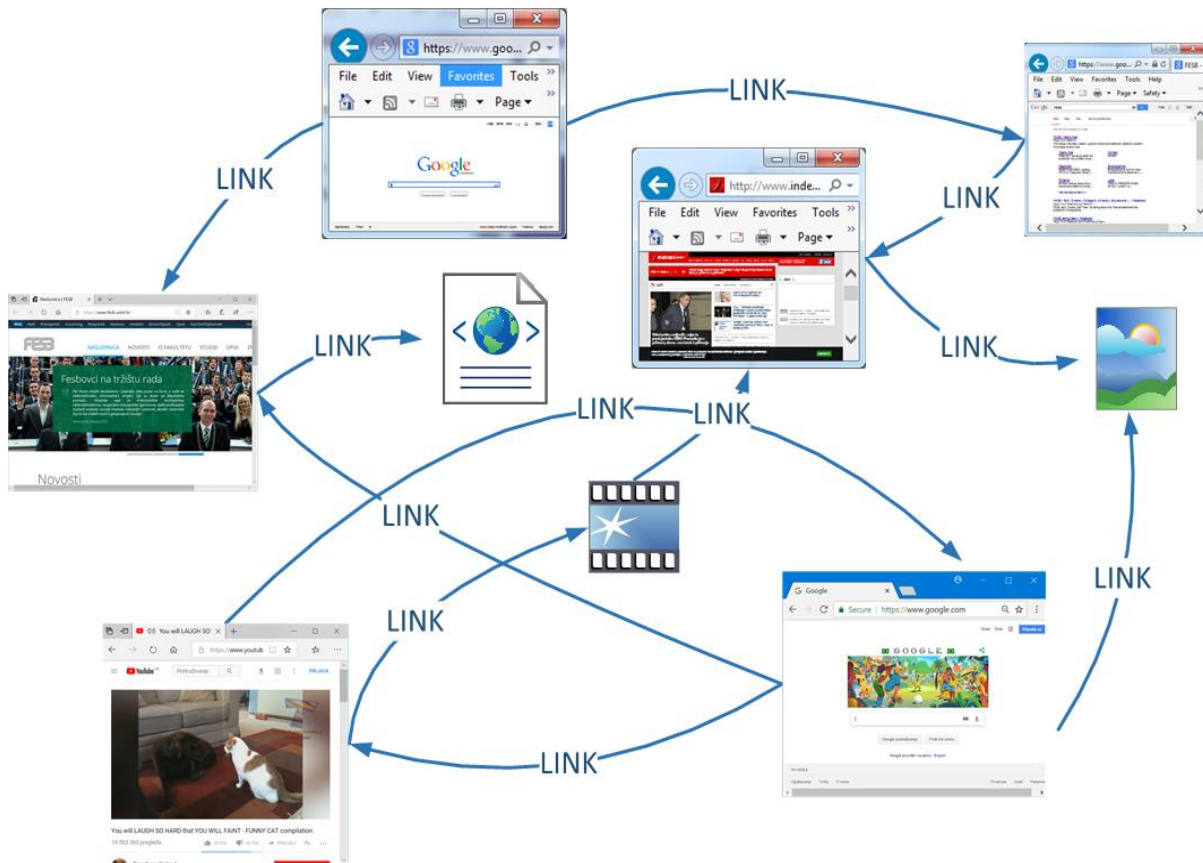
Primjeri karaktera entiteta

Karakter	<code>&entity_name;</code>	<code>&#entity_number;</code>
Razmak	<code>&nbsp;</code>	<code>&#160;</code>
<	<code>&lt;</code>	<code>&#60;</code>
>	<code>&gt;</code>	<code>&#62;</code>
&	<code>&amp;</code>	<code>&#38;</code>
"	<code>&quot;</code>	<code>&#34;</code>
€	<code>&euro;</code>	<code>&#8364;</code>
Γ	<code>&Gamma;</code>	<code>&#915;</code>
©	<code>&copy;</code>	<code>&#169;</code>

Linkovi

- Hipertekst je način organiziranja velikih količina informacija odnosno način rješavanja problema informacijskog preopterećenja (*information overload*).
- Premda je jednostavan, koncept linka je jedan od osnova uspjeha weba.
- Link uvijek ima dva kraja. Počinje od izvorišnog sidra i upućuje na određeno sidro što može biti bilo koji web resurs adresiran URI-jem (HTML stranica, slika, ...).
- Krajevi linka se nazivaju sidra (*anchor*).

Linkovi



Koncept linkova omogućava stvaranje mreže međusobno povezanih informacijskih resursa tj. weba!!!

Linkovi

- HTML elementi `link` i `a` omogućavaju povezivanje dokumenata odnosno kreiranje hiperteksta.
- Atribut `href` (i `link` i `a` elementa) sadrži vrijednost odredišta linka tj. URI.
`FESB`
- Ovo nisu jedini HTML elementi koji definiraju link. Link se definira npr. i preko `img` elementa (``) i preko `form` elementa (`<form method= "GET" action= "URI programa kojem se šalju podaci s forme">`).

Linkovi

- Link je određen odredištem i izvorištem. Odredište se opisuje URI vrijednošću, a izvorište je dokument u kojem je link naveden.
- URI može biti apsolutan URI (http://marjan.fesb.hr/~kiki/moja_stranica.htm), relativan URI (/~kiki/moja_stranica.htm), kao i link definiran unutar iste stranice (URI fragment <#nešto>).

<!-- link unutar istog HTML dokumenta -->

BIOGRAPHICAL NOTE

<!-- oznaka URI fragmenta -->

Professional skills:

lli

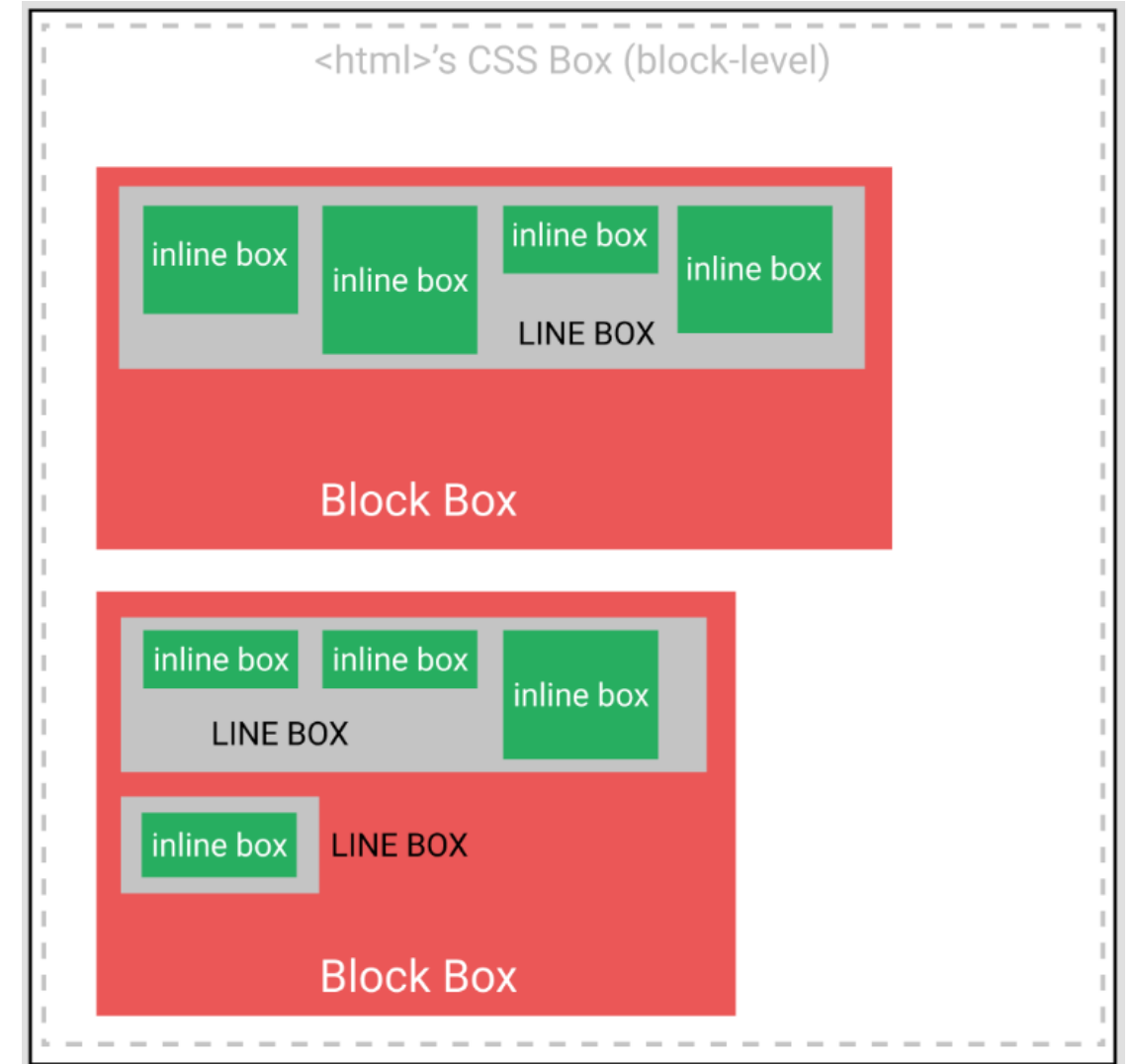
Professional skills:

HTML elementi

- HTML elementi se obzirom na vizualnu prezentaciju odnosno način iscrtavanja obično svrstavaju u dva osnovna tipa, elemente razine bloka (*block-level*) i linijske (umetnute) (*inline*) elemente.
- Elementi razine bloka ili blok elementi uvijek započinju prikazivanje u novoj liniji na prikazu i zauzimaju maksimalni raspoloživi prostor unutar elementa roditelja rastežući se od lijeve do desne strane raspoloživog prostora prikaza pri čemu se ništa ne može iscrtati sa strane blok elementa.
- Linijski elementi ne počinju u novoj liniji već u onoj liniji gdje se trenutno mehanizam prikaza nalazi s iscrtavanjem i zauzimaju samo onoliko prostora koliko im je potrebno za prikaz. Sa strane linijskog elementa se iscrtava sadržaj ostalih elemenata braće ukoliko ima prostora.

HTML elementi

- Iscrtavanje HTML sadržaja po ekranu uvijek ide odozgo prema dole i s lijeva na desno i to redoslijedom kako su elementi navedeni u HTML dokumentu.
- CSS model kutije (*box model*) ovisno o tipu HTML elementa i njegovoj poziciji u HTML dokumentu, definira "kutiju" prikaza elementa. To je pravokutno područje na ekranu za iscrtavanje elementa.
- Linijski elementi bi trebali sadržavati samo podatke i druge linijske elemente, ali ne i blok elemente. Linijski elementi se u pravilu smještaju i iscrtavaju unutar blok elemenata.



HTML elementi

- Elementi *div* i *span* su generički kontejneri koji služe za obuhvaćanje i formatiranje drugih HTML elemenata. Za razliku od predefiniranih elemenata (*p*, *h1*, ...), dizajner sam definira čemu služi generički kontejner.
- Element *span* je linijski generički kontejner. Obično se koristi za formatiranje skupine riječi u tekstu.
- Element *div* je blok generički kontejner. Obično služi za slaganje više blok ili linijskih elemenata unutar jedne strukture koja dijeli zajedničko formatiranje.

`<div>ovo je div s SPAN umetnutim elementom koji se pozicionira unutar iste linije koju zauzima div elementa.</div>`

`<div>ovo je div s <p>P blok elementom</p> koji se pozicionira u novoj liniji u odnosu na div element.</div>`

[Primjer](#)

HTML elementi

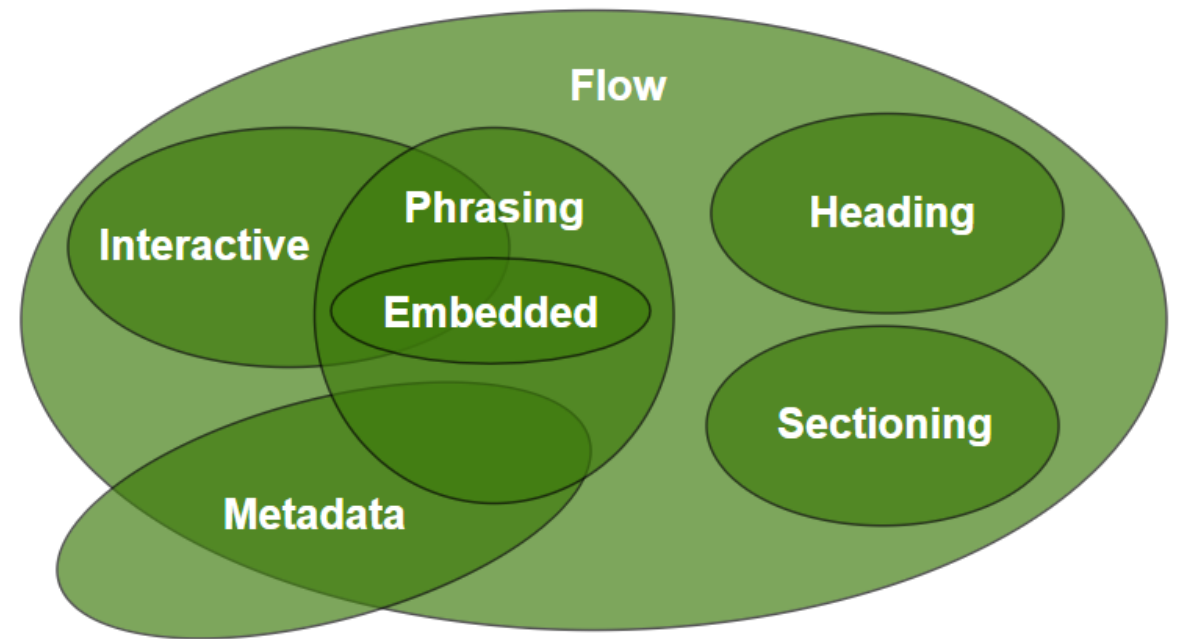
- Tip elementa obzirom na iscrtavanje se može promijeniti korištenjem CSS *display* svojstva koje možete postaviti na "inline" ili "block" i na taj način umetnuti element iscrtavati kao blok element i obrnuto.
- To neće utjecati na druga svojstva elementa, pa umetnuti element i dalje neće podržavati gniježđenje blok elementa iako se iscrtava kao blok element.
- Blok elementi se iscrtavaju jedan ispod drugog, razmak između sadržaja je određen svojstvima CSS kutije (marginama, rubom i nadopunom). Linijski elementi se iscrtavaju jedan do drugog također razmaknuti svojstvima CSS kutije unutar [kutije linije](#) u kojoj mogu biti vertikalno poravnati na različite načine (ili da im se poravna gornja margina, donja ili sadržaj unutar elementa).

HTML elementi

- Ova dva tipa HTML elemenata (*inline*, *block*) su se kategorizirala prema svojstvima vizualne prezentacije do HTML 4.01 specifikacije.
- Ova kategorizacija je i dalje zadržana zbog unazadne kompatibilnosti u mehanizmima prikaza, ali od verzije HTML5 definiran je cijeli skup tipova elemenata prema kategorijama sadržaja ([content categories](#)).
- HTML element može biti član jedne ili više kategorija (ili nijedne) dijeleći zajednička svojstva s drugim elementima koji su također članovi određene kategorije.

HTML elementi

- HTML 5 kategorije sadržaja su *Metadata*, *Flow* (tok), *Sectioning* (sekcije), *Heading* (zaglavlja), *Phrasing* (izrazi), *Embedded* (ugradbeni), *Interactive* (interaktivni) i međusobno se preklapaju (dijele neka zajednička svojstva).
- Većina elemenata koja se koristi u *body* dijelu dokumenta su u kategoriji *flow* sadržaja.



<https://html.spec.whatwg.org/multipage/dom.html#kinds-of-content>

HTML elementi

- Kategorije sadržaja definiraju mogući sadržaj elementa, odnosno sve ono što se može naći između otvorenog i zatvorenog taga, ali i druga svojstva elemenata i pravila koja se odnose na elemente u pojedinim kategorijama uključujući i vizualnu prezentaciju pojedinog elementa.
- Kategorija elemenata tipa "inline" otprilike odgovara HTML 5 *phrasing* kategoriji, dok "block-level" kategorija ne odgovara direktno ni jednoj HTML 5 kategoriji sadržaja, ali "block-level" i "inline" elementi zajedno odgovaraju kategoriji *flow content* u HTML 5 verziji. ([Primjer](#))
- Popis elemenata možete naći na <https://www.w3schools.com/tags/> .

HTML elementi

- Element `` je jedan od ugradbenih elemenata. To su elementi koji uključuju resurse iz drugih izvora. U ovoj kategoriji su još i elementi `<audio>`, `<canvas>`, `<embed>`, `<iframe>`, `<math>`, `<object>`, `<svg>`, `<video>`. Elementi ove kategorije se isrtavaju kao umetnuti elementi.
- Tag [<canvas>](#) definira kontejner, područje za crtanje u koje se može crtati, obično JS kodom.

```
<canvas id="myCanvas"></canvas>

<script>

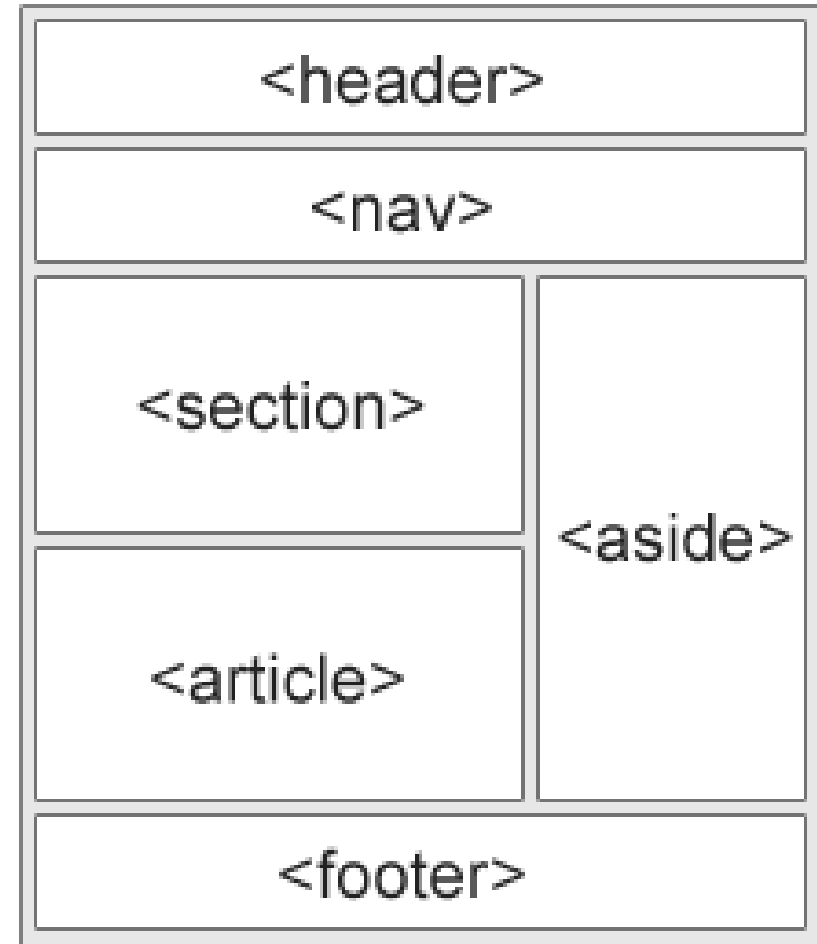
var canvas =
document.getElementById("myCanvas");
var ctx = canvas.getContext("2d");
ctx.fillStyle = "#FF0000";
ctx.fillRect(0, 0, 80, 80);

</script>
```

[Primjer](#)

HTML raspored prikaza

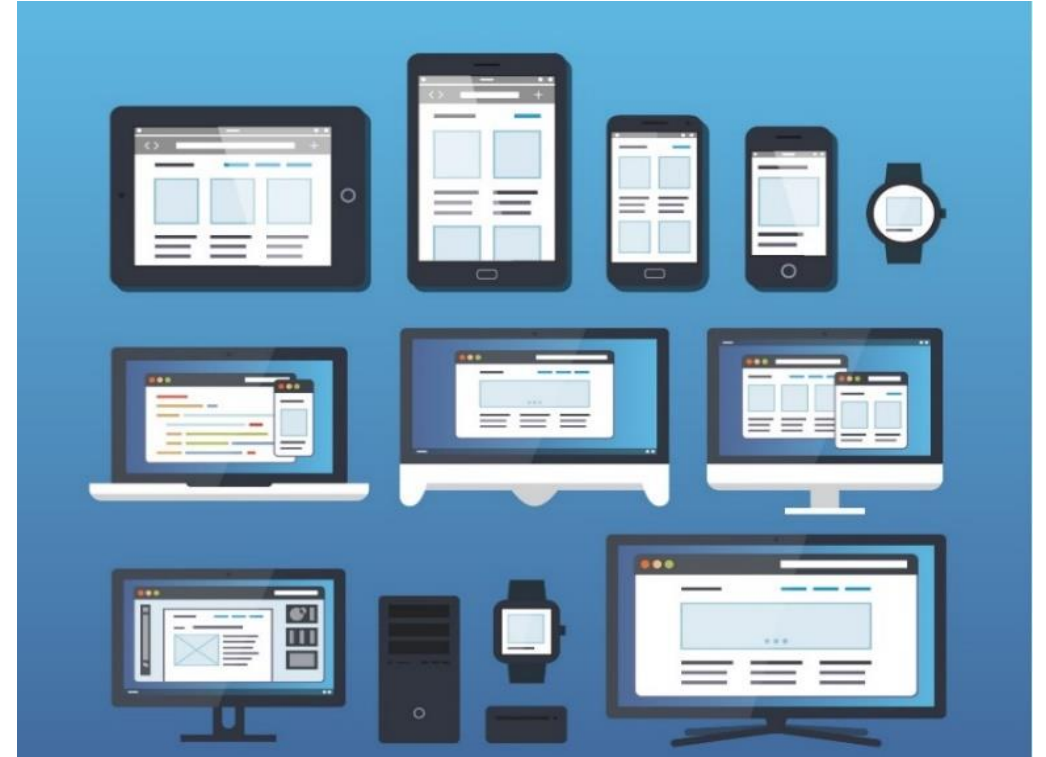
- Često je potrebno web sadržaj prikazati u više kolona, osobito pri prilagođavanju različitim dimenzijama uređaja. HTML 5 uvodi i nove elemente za definiranje različitih dijelove prikaza stranice:
 - `<header>` definira zaglavlje dokumenta
 - `<nav>` definira kontejner za navigacijske linkove
 - `<section>` definira sekciju dokumenta
 - `<article>` definira zasebni neovisni članak
 - `<aside>` definira sadržaj kao bočni izbornik
 - `<footer>` definira podnožje dokumenta
 - `<details>` definira dodatne detalje
 - `<summary>` definira zaglavlje `<details>` elementa



https://www.w3schools.com/html/html_layout.asp

HTML responzivni dizajn

- Danas je uobičajeno korištenje različitih uređaja za pristup web sadržaju.
- Ugrađenim prilagođavanjem izgleda uređaju na kojem se web sadržaj prikazuje postiže se optimalno prikazivanje sadržaja, što se naziva responzivnim dizajnom.
- Elementi responzivne stranice se određenim tehnikama prilagođavaju različitim uređajima koji se danas koriste za pristupanje web sadržaju.



Izvor: iconicbestiary / Freepik

HTML responzivni dizajn

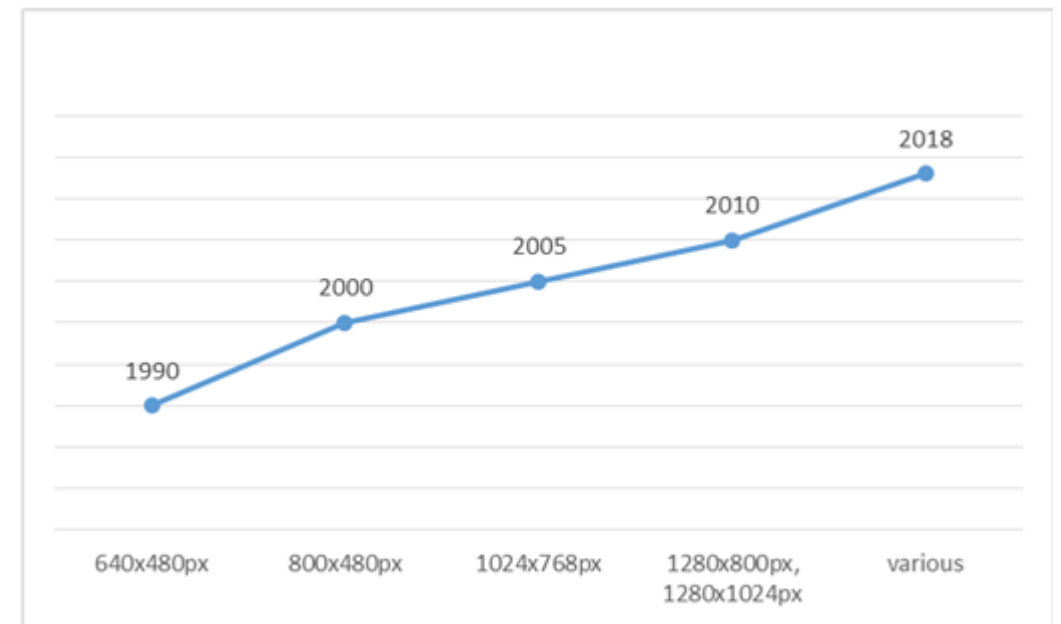
- Prilagođavanje izgleda web stranice uređaju na kojem se prikazuje postiže se na različite načine.
- Jedna od osnovnih tehnika je prilagođavanje rasporeda stranice. Raspored elemenata stranice se obično organizira u više kolona te se na taj način prikaz može vertikalno posložiti u kolonama na užim displejima.
- Također se definiraju i elementi koji nisu neophodni pa se mogu i preskočiti u svrhu prilagođavanja prikaza, jer je prikazivanje samo bitnih elemenata stranice dostatno da bi se korisnicima prenijele informacije u web stranici.
- Prilagođavaju se i dimenzije slika, teksta i sl., oblikuju se veći i jednostavniji navigacijski i drugi elementi, itd.

HTML responzivni dizajn


- Prikaz sadržaja HTML stranice se može organizirati u više kolona na više različitih načina:
 1. HTML tablice su zastarjeli način koji se više ne preporuča jer su to HTML elementi (`<table>` `<tr>` `<td>`) za organizaciju podataka, a ne prikaza sadržaja stranice
 2. Korištenjem CSS float svojstva
 3. Korištenjem CSS flexbox novog rasporeda (CSS3)
 4. Korištenjem CSS definiranog grida s kolonama predefiniranih širina (`display: grid`)
 5. Korištenjem gotovih CSS *frameworka* (Bootstrap, ...) koji koriste neki od navedenih pristupa

HTML responzivni dizajn

- Nekada su se za pristup web sadržaju koristili uređaji sličnih dimenzija i rezolucija prikaza.
- Pojavom mobilnih uređaja i tableta dimenzije uređaja i rezolucije mogu jako varirati.
- Kada je veličina stranice veća od veličine prikaza preglednici ubacuju *scrollbarove* za prikaz stranica većih od područja za pregled.
Vertikalno *scrolliranje* nije problem, to je uobičajeno i korisnici to očekuju, međutim horizontalno *scrolliranje* je problematično.



Evolucija prosječnih rezolucija ekrana

 PHONE	OPERATING SYSTEM	PHYS SIZE ^{IN}	PHYS SIZE ^{CM}	W ^{PIX}	H ^{PIX}	DEVICE WIDTH	PIX PER INCH	PIX DENSITY	ASPECT RATIO
Apple iPhone 7 Plus	iOS	5.5	14.0	1080	1920	1080	401	100% MDPI	9 : 16
Apple iPhone XR	iOS	6.1	15.5	828	1792	828	326	100% MDPI	19 . 5 : 9
Apple iPhone 8 Plus	iOS	5.5	13.9	1080	1920	1080	401	100% MDPI	16 : 9
Apple iPhone XS	iOS	5.8	14.7	1125	2436	1125	458	100% MDPI	19 . 5 : 9
Apple iPhone XS Max	iOS	6.5	16.5	1242	2688	1242	458	100% MDPI	19 . 5 : 9
Apple iPhone SE	iOS	4.0	10.0	640	1136	640	326	100% MDPI	16 : 9
Apple iPhone 6 Plus	iOS	5.5	14.0	1080	1920	1080	401	100% MDPI	9 : 16
Samsung Galaxy S	Android	4.0	10.0	480	800	320	233	150% HDPI	3 : 5
Samsung Galaxy S8+	Android	6.2	15.7	2960	1440	2960	529	100% MDPI	5 : 9
Apple iPhone 5 (5c, 5s)	iOS	4.0	10.0	640	1136	320	326	200% XHDPI	40 : 71
Google Pixel	Android	5.0	12.7	1080	1920	1080	441	100% MDPI	16 : 9
Apple iPhone 4 (4, 4S)	iOS	3.5	8.9	640	960	320	326	200% XHDPI	2 : 3

HTML responzivni dizajn

- Mobilni preglednici na uređajima manjih dimenzija iscrtavaju stranicu na širinu desktop ekrana (obično 980px) i onda pokušavaju prilagoditi prikaz skaliranjem sadržaja da stane unutar ekrana mobitela na način da se izbjegne horizontalno skroliranje. Npr. ako je ekran 490px = 980px/2, sve dimenzije će biti duplo manje.
- Tada ni veličina slova nije konzistentna od uređaja do uređaja. Dimenzija sadržaja vezana uz rezoluciju uređaja na manjem uređaju je jednostavno manja pa korisnici trebaju zumirati sadržaj.
- HTML 5 uvodi novi meta tag [viewport](#) kojim se web stranice automatski optimiziraju za prikaz na različitim uređajima. U zaglavlju stranice treba uključiti ovaj meta tag koji upućuje preglednik kako da upravlja dimenzijama i skaliranjem stranice. Npr. `<meta name="viewport" content="width=device-width, initial-scale=1">`

HTML responzivni dizajn

- Meta tag *viewport* je područje prozora preglednika u kojem je sadržaj stranice vidljiv. Ovaj element neće jednostavno skalirati cjelokupni sadržaj na manju dimenziju nego barem donekle prilagođava izgled stranica koje nisu optimizirane za manje ekrane.
- Sadržaj *width=device-width* osigurava da naš prikaz odgovara širini ekrana u pikselima neovisnim o uređaju pa se tako sadržaj prilagođava različitim dimenzijama ekrana.
- Sadržaj *initial-scale=1* uspostavlja relaciju 1:1 između CSS piksela i piksela neovisnih o uređaju odnosno da je početni zoom stranice neovisno o veličini ekrana uvijek 1 odnosno cijelo područje displeja. Neki preglednici ne prilagođavaju stranicu *landscape* modu, ali *initial-scale=1* nalaže pregledniku da uspostavi relaciju 1:1 između CSS piksela i piksela neovisnih o uređaju neovisno o orijentaciji uređaja, pa tako sadržaj stranice može iskoristiti cjelokupnu širinu ekrana *landscape* modu.

HTML responzivni dizajn

- Prvi prikaz stranice na mobilnom uređaju je bez meta taga viewport, a drugi prikaz je sa meta tagom `<meta name="viewport" content="width=device-width, initial-scale=1">`



HTML responzivni dizajn

- Pored meta elementa *viewport* i u pisanju samog HTML i CSS koda treba voditi računa o prilagodbi različitim dimenzijama ekrana.
- Definiranje sadržaja u više kolona je samo jedan dio toga.
- Korištenjem relativnih geometrijskih jedinica (visine, širine, položaja elementa) (width: 100%, a ne width: 300px) izbjegavamo mogućnost prikaza preširokog za uređaj na kojem ga prikazujemo zbog čega onda korisnik mora horizontalno pomicati sadržaj da ga vidi. Korisnici nisu baš naviknuti na horizontalno pomicanje (*scrolling*) da bi vidjeli sadržaj stranice.
- Također je preporučeno da jedna kolona ne sadrži više od 70 do 80 karaktera po liniji (10~tak riječi), pa bi svaki put kada veličina tekst bloka prijeđe 10 riječi trebalo staviti prelazak u novi red.
- Glavni koncept responzivnog dizajna je fluidnost i proporcionalnost prikaza u odnosu na raspoređivanje elemenata fiksnih širina.