HTML

Kada se daju imena fajlovima za web stranu: ako imamo space u url ce biti prikazano: %20, umesto toga mozemo koristiti\_ ili -, ne bi trebalo koristiti specijalne karaktere, praviti nazive sto krace moguce i pisati malim slovima /zbog nekih unix servera/

HTML5 osnovna struktura:

<!DOCTYPE html>

<html lang='en'>

<head>

<meta charset="UTF-8" />

<title>

HTML Document

</title>

<link rel="stylesheet" type="text/css" HREF="fajl.css" />

</head>

<body>

</body>

</html>

u <head> ide javaskript css itd. a i naslov: <title> Naslov </title>

a u <body> sadrzaj veb strane

neki kodovi ne moraju da se zavrse kao npr <hr> (iscrtava horizontalnu liniju preko strane), takvi kodovi se mogu obeleziti ovako: <hr/>

<meta> - stoji u zaglavlju i koristi se da pruzi browser-u odredjene informacije o html stranici, obicno u okviru njega stoji atribut: <meta charset="UTF-8" />, a moze i recimo: <meta http-equiv="refresh" content="3; http://www.sajt.html" /> ova komanda prebacuje stranicu na www.sajt.html za 3 sekunde

<meta name=”keywords” content=”portfolio,graphic design, itd” ovakav tag je osmisljen da bi search engine kategorisao stranu po kljucnim recima, ali to je dosta zloupotrebljavano pa sada ignorisu

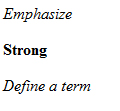
<div> sluzi da odvoji celine koda (divide), na koje recimo primenjujemo css

<br> line brake, nesto kao Enter, nema end tag<p> paragraf </p>

<h1> Heading(naslov), najveci </h1> - koristi se za glavno naslove, ono bitno sto zelimo da vidi google <h6> Heading(naslov), najmanji </h6> - heading je blok elemenat ali moze recimo u okviru sebe imati paragraf ciji ce tekst tada biti u velicini heading-a

<sup> - superscript - X2  <sub> - subscript - X2

<small> - mala slova - This is small

<q> navod </q> - navodnici

<blockquote> txt </blockquote> - izdvojen txt u paragrafu

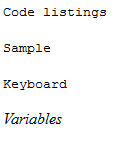
<s> - strikeout - ~~strikeout~~ , <del> - delete - ~~delete~~

<b> Bold - **Bold**, <i> Italic - *Italic*

<u> - underline - underline

<em> Emphasize </em> - kao italic

<strong> Strong </strong> - kao bold

<dfn> Define a term </dfn>

<code> Code listings </code>

<samp> Sample </samp>

<kbd> Keyboard </kbd>

<var> Variables </var>

<mark> - Hajlajtuje deo teksta - mark - ovakav je po defaultu, doduse moze se menjati sa css-om.

&shy; - soft hypen - kada imamo neku bas dugu rec sa ovim elementom sugerisemo browseru da na ovom mestu moze da prekine rec i nastavi je u novom redu

<wbr/> - word breaking - radi isto sto i &shy; samo to radi bez crtice

<pre> - preformated text - tekst u okviru ovog taga bice prikazan onako kako je otkucan

Postoje blok elementi kao div i inline elementi kao span razlika je u tome sto blok elementi se prikazuju odvojeni u blokovima, a inline u oviru linija teksta

<span> - obelezava deo teksta u recimo paragrafu koji mozemo da boldiramo, kosimo, bojimo itd uz pomoc css-a: <span style=”css” > text </span>. On je kao div za text, jer div ne moze da se koristi unutar txta

html tagovi mogu imati atribute koji blize odredjuju tagove

Atributi se uvek pisu u pocetnom tagu i to u formatu name="value" - navodnici nisu nuzni, broseri ce protumacitit i bez navodnika, ali ako se stavljaju moraju biti isti na pocetku i kraju “value” ili ‘value’ i ako ih ima vise, pise se 1,

Linkovi: <a href="http://www.adresa.com">Naziv linka</a> Naziv linka je ono sto pise na stranici

Atribut: target=”\_blank” - otvara link u novom tabu:

Link mozemo povezati sa delom strane (back to top recimo) i tada u linku kao adresu definisemo id tog dela: href=”#top” a tom elementu ka kom vodi link damo id: id=”top”

Kada povezujemo link ili css sa dokumentom koji je u istom folderu adresu definisemo samo imenom fajla, a ako se nalazi u folderu iznad trenutnog: ../fajl.html(css) ../ - znaci up, a ako se nalazi u nekom folderu koji je u folderu u kom nam je stranica: folder/fajl.html(css)

/images/slika.jpg - ovako definisemo adresu fajla koja krece od root foldera, ako nam se recimo strana nalazi u nekom folderu, krace je za kucanje od: ../images/slika.jpg

Neki browseri prave problem u slucajevima kada je link slika i da bi se to izbeglo trebalo bi da nema praznog prostora izmedju pocetnih kao i zavrsnih tagova linka i slike

Html dozvoljava postavljanje custom root foldera (folder na kompu ili serveru gde se nalazi html fajl) komandom u zaglavlju: <base href=”http://www.adresa.com/” /> /ovo verovatno ti nikad nece trebati/

U okviru linka (anchor taga) mozemo strpati vise elemenata - tako da recimo imamo heading i par paragrafa umesto naziva linka

Slika: <img src="w3schools.jpg" width="104" height="142" > ne moraju se unositi dimenzije slike, to ima znacaja pri sporoj internet konekciji-ako se strana bas sporo ucitava a nisu date dimenzije slike u jednom trenutku ce se organizacija strane promeniti, kada menjamo visinu ili sirinu slike u html-u ona se razvlaci ili skuplja, osim ako definisemo samo jednu vrednost, druga ce je proporcionalno pratiti.

Sa sajta placeold.it mozemo da ubacujemo placeholdere u sajt, tako sto cemo zadati: src=”http://placehold.it/300x200” sa zeljenim dimenzijama

Slika moze imati atribute: hspace="Xpx" i vspace="Ypx" - definise velicinu praznog prostora oko slike

alt=”tekst koji se prikazuje ako se slika ne ucita” title=”natpis koji se vidi kad smo kursorom iznad slike”

slika moze imati imagemap - oblike unutar slike koji sluze kao link - potrazi ako ti treba

Mozemo zadati da slika bude pozadina strane: <body background="pozadina1.jpg">

Ako elementu koda nije definisana pozicija browser ga postavlja levo i gore.

A ako nije definisana boja elemenat ce automatski biti crn

**Liste**: Numerisane: <ol> - ima atribut Start=”5” - lista ce poceti od br. 5 /koristimo kad imamo listu u listi/ - ovaj atribut se moze primenjivati i na liste sa rimskim br.

Nenumerisne: <ul> Stavke za obe liste se definisu <li> tagom i on je block level element

Lista moze sadrzati listu ili vise njih u sebi

Lista definicija: <dl> , a stavke liste su: pojam koji se definise: <dt> (sortira se automatski ulevo), definicije: <dd> pojavljuju se u novom redu uvucene /ovakav tip liste moze da se koristi recimo za intervju sa pitanjima i odgovorima ili naziv proizvoda + detalji o proizvodu u <dd> tagu/

<table> - tabela , atribut summary dodatno pojasnjava tabelu za screen readere a ne prikazuje se u browseru, <caption> - naziv tabele -stavlja se pre ostalih tagova u tabeli a na stranici se ispisuje iznad tabele, <tr> - red tabele, <td> - kolona tabele, <th> - header tabele, Konstrukcija tabele izgleda npr. Ovako:

<table>

<tr> <th> Naslov1</th> <th> Naslov2</th> </tr>

|  |  |
| --- | --- |
| Naslov1 | Naslov2 |
| kolona 1 | | kolona 2 | |
| kolona 2-1 | | kolona 2-2 | |

<tr> <td> kolona1 </td> <td> kolona2 </td> </tr>

<tr> <td> kolona2-1 </td> <td> kolona2-2 </td> </tr>

</table>

Html5 ima neke semanticke tagove za tabelu u koje se stavljaju tagovi koji se i inace koriste (<tr>, <th>, <td>): <thead> - u sebi sadrzi header tabele, <tbody> - u sebi ima sadrzaj tabele, <tfoot> - futer tabele, gde god ga stavimo prikazivace se na dnu tabele.

Da bi stilizovali kolone stavljamo tag <colgroup> /stavlja se na pocetak tabele, odmah ispod <caption> taga, ako postoji, a pre headinga tabele/ koji sadrzi <col/> tagove sa definisanom klasom. <col/> tagovi predstavljaju kolone i to prvi <col/> tag - 1. kolonu, drugi - 2. Kolonu itd. Ako hocemo da <col/> predstavlja 2 susedne kolone, dajemo mu atribut span=”2”

cellpadding=”Xpx” - unutrasnja margina izmedju sadrzaja celije i njene granice,

cellspacing=”Ypx” - razmak izmedju celija tabele

colspan i rowspan su atributi koji definisu broj kolona odnosno redova duz kojih se celija prostire, npr: colspan=”3” - celija zauzima 3 kolone

allign=”center” (“left”, “right”) - atribut za horizontalno poravnanje

vallign=”middle” (“top”, “bottom”) - atribut za vertikalno poravnanje (vaze za bilo koji element,ne samo txt)

(u html5 atributi za oblikovanje tabele su izbaceni i za to se sad kristi css)

<!-- komentar --> - browser ne cita komentar vec sluzi da nesto pojasni kad se pregleda izvorni kod (mislim da mora bas ovako sa praznim prostorom oko komentara zbog svih browser-a)

Jedino Internet Explorer cita odredjene uslovne komentare: <!--[if lt IE 9]> <script src="\_scripts/respond.min.js"></script> <![endif]--> - ovaj skript ce se izvrsiti ako ga ucita IE manji od verzije 9 / less then or equal - lte, greater than - gt, if not IE - !IE /

Neke karaktere koji se koriste u samom kodu ili druge specijalne karaktere u htmlu unosimo ovako npr: &alpha; - α &copy; - © &lt; (ili &#60; ili &#x3c)- < &commat; - @ itd.

&nbsp; - space

Javascript u headu mozemo ubaciti sa tagom <script>

<script type="text/javascript"> sadrzaj javascripta </script>

A kada ga uvozimo iz posebnog fajla: <script type="text/javascript" src="naziv fajla.js"></script> (iz nekog razloga bas ovako mora da se pise, da bi ga svi browseri pravilno tumacili)

preporuclivo je script tagove ubacivati pre </body> taga zbog performansi

async atribut: govori browseru da ucita script kasnije, i tako ubrza ucitavanje strane (ne moze se primeniti na inline skriptove, odnosno skript bez src atributa) : <script src="script.js" async></script>

koristimo za ucitavanje 3rd party skriptova poput twittera, shareholic i sl, tj. one koji nisu nuzni da se ucitaju odmah sa DOM-om

Novi HTML5 tagovi: namena im je da zamene <div> tagove semantickim tagovima

<nav> - navigacija

<section> - semanticka/logicka celina/deo stranice koja nije definisana nekim drugim tagom u principu sadrzi druge tagove

<article> - clanak, blog post, komentar ili drugi nezavisni elemenat, moze sadrzati druge tagove: section, header, nav itd

<aside> - sekundarni, povezani sadrzaj sa okolnim, sidebar, dijagrami itd.

<header> - zaglavlje stranice ili sekcije, moze da bude i zaglavlje article-a

<footer> - futer

<hgroup> - tag za grupisane naslove/heading-e - najveci ce biti naslov a sledeci podnaslov.

<figure> - moze biti slika, ilustracija, grafikon, dodatni sadrzaj uz text itd. Ako je slika u okviru ovog taga ide <img> tag

<figcaption> - tekst koji ide uz sliku, ilustraciju… kada je pre <img> taga: natpis je iznad slike, a kad je posle natpis je ispod

<details> i <summary> se koriste u kombinaciji, i funkcionisu tako da korisnik klikom na naslov koji se stavlja u <summary> tag moze da prosiri sadrzaj na kompletan sastav <details> taga, i ponovnim klikom da ga skupi, a tagovi se koriste ovako:

<details>

<summary> click to expand </summary>

<p> text </p>

</details>

Ako hocemo da sadrzaj bude prosiren po defaultu: details tagu damu atribut open: <details open>

<audio> - za umetanje zvuka, da bi audio zapis mogli da kontrolisemo ovom tagu dajemo atribut controls <audio controls> - tada ce na stranici biti prikazane default kontrole brosera-a ako hocemo da to stilizujemo - javascript

autoplay - zvuk otpocinje sa ucitavanjem stranice

loop - zvuk pocinje iz pocetka kad se zavrsi

u okviru <audio> taga stavljamo <source tag>:

<source src="media/song.m4a" type="audio/x-aac" />

<source src="media/song.mp3" type="audio/mpeg" />

<source src="media/song.ogg" type="audio/ogg" />

Posto razliciti Browseri podrzavaju razlicite formate, moramo da stavljamo vise audio fajlova u razlicitim formatima da bi svi mogli da ih ucitaju a format definisemo atributom type

<video> - tag za ubacivanje video zapisa, funkcionise isto kao i <audio>, stavljamo u <source> vise formata da bi razliciti browseri ucitali video, isto imamo controls, autoplay i loop

Razlika je u tome sto za video definisemo velicinu: <audio width="860" height="480">

preload - ucitavanje videa zajedno sa otvaranjem stranice

preload="none" - video se ucitava tek kad se klikne play

muted - video bez zvuka

poster - dok se ne pritisne play na prostoru videa stoji slika

bilo kom elementu mozemo da dodelimo odredjeni podatak sa atributom data-ime podatka=”vrednost podatka” a sa javascript-om pozivamo ime podatka i koristimo ga u njemu.

<meter> - graficki metar koji prikazuje progres nekog trenutno staticnog procesa recimo slobodnog procesa na serveru. Jedini neophodan atribut mu je value koji se izrazava: 0.6 je u stvari 60%, ako imamo ceo broj onda su potrebni atributi min i max koji odredjuju minimalnu i maksimalnu vrednost.

Kompletan tag sa svim atributima bi izgledao ovako: <meter value="50" min="0" max="100" low="10" high="80" optimum="47">47 of 100</meter> - ako je vrednost manja od low ili veca od high meter ce biti u zutoj boji, inace je zelen, za optimum trenutno je isto zelen bez neke razlike, 47 of 100 ce biti prikazano u browserima koji ne podrzavaju html5

<progress> - progres bar, namenjen je za prikazivanje progresa trenutno dinamickog procesa kao recimo download fajla, takodje ima atribute value, max i min -imace vrednost 0 ako nije definisan, i takodje izmedju pocetnog i zavrsnog taga mu upisujemo vrednost koja ce se videti u browseru ako ne podrzava html5. Mozemo umesto njega koristiti i tag meter i razlika ce biti cisto graficka /nece biti sistemski progress bar vec ce izgledati kao meter/

Microdata ili rich snippets /po googlu/ su podaci koje definisemo na strani da bi search engini bolje kategorisali stranu, definisali sta je na slikama… To je tehnologija koja je tek u nastanku. Mozemo sami definisati pojam ili koristiti neke od on-line recnika kao: http://schema.org/Thing - ?

Stavljamo atribute, u pocetni tag sekcije, div-a o predmetu: itemscope itemtype = "http://schema.org/Thing", gde imamo naziv predmeta - itemprop="name"", za sliku predmeta - itemprop="image"", opis - itemprop="description" ?

html5 outlines su sekcije stranice - prikazane i organizovane kao lista, postoje alati na internetu za njihovo prikazivanje a poenta svega je dobro semanticko i strukturno organizovanje web strane.

FRAMES: postoji prevazidjena varijanta framesa sa <frameset> tagom koji stoji umesto <body> :

<frameset cols="\*, 75%"> - ovaj frameset je podeljen na 2 dela po kolonama, desni od 75% i

<frame src="frames/f1.html" /> ostatak da smo stavili 2 \* onda bi taj ostatak bio jednako

<frameset rows="100, \*"> podeljen izmedju njih

<frame src="frames/f2.html" />

<frame src="frames/f3.html" />

</frameset>

</frameset>

Drugi frameset je podeljen po redovima, gornji od 100px i donji ostatak visine.

Frame moze sadrzati drugi frameset ili stranicu.

Da bi izbrisali ruzni okvir oko frejmova dajemo atribut frameborder=”0” framesetu

Novija varijanta je iframe: <iframe id="frame1" src="frames/f4.html"></iframe>

on takodje u sebe ucitava drugu stranicu i on se skroluje ako je sadrzaj veci od njegove velicine

iframe mozemo stilizovati css-om, sto ne bismo mogli sa klasicnim frame-om.

iframe mozemo simulirati sa css-om tako sto cemo delu strane koji bi bio iframe dati atribut: overflow: auto i to ce omoguciti da se sadrzaj ne “izliva” vec skroluje, prednost ovoga je da nam sadrzaj frame-a bude unutar stranice a ne na nekoj drugoj.

overflow: hidden koristimo da sadrzaj elementa stane u njegovu zadatu velicinu a ne izliva se van njega.

FORMULARI

<form> - tag koji sadrzi ceo formular

<input> - tag za input polje, checkbox ili radio button - predstavlja ono gde korisnik unosi neke podatke koji se posle salju serveru ili javascript-u, i to: naziv input-a, koji se pise kao atriut: name=”ime vrednosti” i vrednost koju unosi korisnik ili je mi definisemo: value=”neka vrednost”

Ako je mi definisemo ta vrednost ce biti poslata kad pritisnemo submit ako je korisnik ne promeni

readonly - atribut za vrednost u input polju koja ne moze da se menja /za slucaj da ne mogu svi da unose vrednost u odredjena polja, vec samo poneko, a to mozemo da regulisemo sa javascriptom, da ih ucinimo editabilnima/ <input type="text" name="text1" value=”foo” readonly/>

<label>Txt</label> - neki natpis u okviru formulara, ako mu dodamo atribut: for=”naziv id-ija” a odredjenom input polju definisemo ovaj id a ujedno i name=”naziv id-ija”, kada kliknemo kursorom na natpis input polje ce postati aktivno

Tip input -a odredjujemo atributom: type="text" - text input polje, “password" - polje za unosenje lozinke, “checkbox" - cekbox, /vrednost koju salje je on ili off/, “radio" - radio button /vise cekboxova moze biti cekirano u isto vreme, a radio buttoni su medjusobno iskljucivi, grupisu se sa istim name-om, a imaju razlicite value/, “submit" - submit button - salje unete podatke serveru.

Ono sto pise u submit button-u definisemo sa value="txt"

Html5 donosi neke nove tipove koji su delimicno podrzani po browserima: “date" - otvara date picker, “color" - otvara color picker, verovatno mora da ima i pocetnu value, “email" - polje za unos e-mail adrese ako uz sebe ima atribut: multiple mogu se unositi vise e-mail adresa odvojenih zarezom, “number" - broj chooser sa strelicama gore i dole klikom na koje povecavamo ili smanjujemo br. Ima atribute: min = “1” i max = “5” i verovatno mora da ima i value, “range" - slajder za odabir neke vrednosti. Ima atribute: min=“0”, max=“200” i step=”10” - sa kojim definisemo sekvencu za koju se slajder pomera /ne mora da se unosi, i tada joj je vrednost 1/, verovatno mora da ima i pocetni value, “search" - input polje za search, u sustini je text samo ako browser podrzava ovaj tip izgleda kao search box u skladu sa browserom i os, “url" - input polje za url

Default input-a je text atribut, on ce biti ako ne definisemo tip ili browser ne podrzava tu vrednost

url, email su kao i obicno polje text, samo se validuju: proverava se da li je uneta validna e-mail ili url adresa.

<textarea> - prostor za unos texta, najcesce sam vidjao ovo kao prostor za komentare, i ovaj tag ima atribut name. moze da ima i atribute rows=”x” cols=”y” gde su x i y broj redova i kolona koji odredjuju velicinu textarea

Dropdown lista se pise u okviru <select> taga u ovom formatu:

<select name="ime">

<option value="" disabled selected>naslov</option>

<option value="1">izbor 1</option>

<option value="2">izbor 2</option>

</select>

name atribut se daje listi, a vrednost koja se salje uz njega je atribut value <option> taga koji obmotava natpis koji ce biti prikazan korisniku, option tag ima atribute: selected - opcija koja je default selektovana /prikazana kad korisnik ucita stranu/ i disabled - opcija koja se ne moze odabrati.

Ako <select> tagu damo atribut size="3" onda ce lista biti skrolujuca, a atribut multiple - omogucava selektovanje vise stavki u listi drzanjem Ctrl na tastaturi. Ako imamo ovaj atribut po defaultu ce biti skrolujuca lista u velicini 4.

Vise <option> tagova mozemo grupisati u okviru <optgroup> taga koji ima atribut: label=”naziv grupe”

Zbog izgleda pozeljno ih je umotavati u paragrafe, sa txt-om pored inputa da korisnik zna o cemu se radi: <p>Password:<input type="password" name="password1" required/> </p>

Radio button i checkbox imaju atribut: checked - polje koje ce default biti selektovano, a polja za unos txta /ukljucujuci i lozinku/ : autofocus - polje u kojem ce se kad otvorimo stranicu pojaviti kursor

required - atribut za polje koje mora da se popuni

placeholder=”text” - onaj sivi txt u poljima za input txta koji stoji dok ne krenemo da kucamo u njemu

pattern="\d{4}-\d{2}-\d{2}" title="YYYY-MM-DD" - pattern sluzi da definise format podatka koji se unosi ?, title - tooltip koji se prikazuje kada kursorom predjemo preko polja. /u ovom slucaju je upotrebljen kao pomoc korisniku da unese pravilne podatke ali moze da se koristi i u neke druge svrhe/

form tag ima atribut action="url/test.php" sa url-om skripta koji se poziva kada saljemo podatke serveru, odnosno url adresu na koju saljemo podatke, a rakodje i atribut: method="post"-u url-u ne mozemo da vidimo unete podatke ili “get” -salje unete podatke kroz url ?

Postoji i: type=”reset” - reset button i type=”button” - obican button, koristi se za pozivanje javascripta

Submit button mozemo zameniti slikom, umesto type=”submit” kucamo type=”image” src=”url slike”

Postoje i skrivena polja type=”hidden” ?

Atribut tabindex=”1” definise redosled po kojem ce se menjati aktivno polje kad pritisnemo Tab, 1 - prvo polje, 2 - drugo itd.

autocomplete=”off” - iskljucuje autocomplete /browser pamti sta smo vec kucali u odredjeno polje i kad krenemo ponovo to da kucamo u istom polju ponudi nam autocomplete/, moze se primeniti na ceo formular ili samo na odredjeno polje, isto vazi i za “on”

medjutim mozemo i mi definisati odredjene vrednosti autocomplet-a: tako sto cemo input polju dati atribut

list="id liste" i definisati listu u kodu:

<datalist id="id liste">

<option value="vrednost 1"></option>

<option value="vrednost 2"> </option>

</datalist>

Imamo i <output> tag za prikazivanje outputa neke racunske operacije ali je jos uvek nov i nedovoljno razvijen tag ima atribut for=”input name” input name je ime vrednosti koja se unosi u formularu na osnovu koje izracunava output

Delove formulara mozemo da grupisemo tako sto cemo ih staviti unutar taga <fieldset> i kao prvi tag u njemu stavljamo: <legend> Naziv Grupe </legend> u browseru ce polja u oviru <fieldset> taga biti prikazana uokvirena, sa nazivom grupe u gornjem levom uglu po defaultu

Formulari u okviru sebe mogu imati i druge elemente kao tabele liste, zavisno od toga sta nam treba

Atribut accesskey=”x” - moze da se definise za bilo koji elemenat a funkcionise tako da aktivira/fokusira taj element kombinacijom Alt + X za vecinu browsera, Shift + Alt + X - za mozilu precice na tastaturi

DATA URI - mozemo da ubacimo fajl unutar html-a i css-a koristeci base64 encoding, npr. da u CSS-u ubacimo SVG kao pozadinu, s time dobijamo manje http request-ova ali je taj fajl za 33% veci u bajtovima:

li {

background: url(data:image/svg;base64,<base64kod>) no-repeat;

}

mane su sto se slika ne kesira(ponovo se ucitava kad ponovo posetimo stranu), i sto zahteva dodatnu procesorsku snagu, sto predstavlja problem na telefonima. Generalno URI moze da se koristi za fontove i male vektorske (png) slike

CSS

OSNOVE

css se sastoji od pravila, koja cine selektor i blok deklaracija : p {line-height: 140%;}

Elementi nasledjuju svoje osobine od “roditelja” osim ako im css nije posebno definisan, mnogim css atributima mogzemo dati vrednost: inherit odnosno da naslede osobine.

Css koji je kasnije u kodu ima prvenstvo u odnosu na onaj pre njega ako definisu istu stvar

Moze da stoji u okviru pocetnih tagova nekog elementa u atributu style npr: <div style="background:green; width:50px">

komande css-a se razdvajaju sa ;

A moze i da stoji u head delu html koda u okviru <style> taga. A moze i da mu se doda atribut: type=”text/css”

Najprakticnije je definisati css u okviru posebnog fajla sa koga cemo ga ucitavati na stranicu sa komandom u headu /ispod naslova/: <link rel="stylesheet" type="text/css" href="css fajl.css">

to se radi da ako imamo vise stranica u okviru sajta definisemo css na jednom mestu a ne posebno na svakoj strani. Taj fajl sadrzi isti css kao sto bi bio u zaglavlju html strane.

Css selektori mogu biti tagovi ili vise njih odjednom i tad ih odvajamo zarezom: div, p, h1{css}

Moze biti body{css} ili svaki element: \*{css}

Mozemo selektovati i html elemente po atributu, npr: a[href=”google.com”] {color:green;}

Mogu biti tagovi koji su u odredjenim tagovima: p a {css} - css ce se odnositi na sve linkove koji su unutar paragrafa bez obzira da li su “sinovi” ili “unuci”. p > a {css} - ce se odnositi samo na linkove koji su direktni potomci. A p + a {css} ce se npr. odnositi na sve linkove posle paragrafa (naravno ako su braca,imaju isti roditeljski elemenat), dok p ~ a {css} radi isto kao: p + a s tim da link ne mora da bude odma iza paragrafa

Imamo i id selektore: <div id=”naziv id-a”> - u html-u, #naziv id-a{css} - u css-u

Medjutim id selektori se mogu koristiti samo za jedan objekat i zato postoje klase koje se mogu primenjivati na vise njih, u html-u ih definisemo: <div class=”naziv klase”>, a u css-u:

.naziv klase{css} - na pocetku stoji tacka - .

Ako na element primenjujemo vise klasa, u atributu ih odvajamo sa razmakom: <section class=”sidebar brown clear”>

Mozemo i da definisemo css za samo odredjene elemente koji pripadaju nekoj klasi: p.naziv klase{css}

Ako u ovoj adresi imamo vise klasa i stilova pisemo ih odvojene razmakom: #container .topbar .left .logo h1

U tu selekciju mozemo i da ubacimo drugi tag u okviru klase u okviru taga: p.sidebar h1{css}

Imamo i pseudo selektore, npr. za link: a:visited {css}, i njih mozemo slagati u okviru drugih elemenata i definisati vise zajedno : p.sidebar a:link, p.sidebar a:visited {css} ili: p.klasa:first-line{css}

:target nam omogucava da stilisemo delove na strani kad se pozovu linkom: <a href="#main-navigation">Menu</a> pa u css-u: #main-navigation:target {} mozemo da definisemo stilove za #main-navigation koji ce se primeniti onda kada kliknemo na link

:nth-child(x) - pseudo selektor za selektovanje odredjenog “child-a” (x), unutar odredjenog taga, npr: ul:nth-child(3)ce selektovati 3. elemenat liste.

Umesto broja mozemo da koristimo i :nth-child(odd)i tada ce gadjati svaki neparan elemenat

A isto: (even)parne

:first-of-type - pseudo selektor za selektovanje prvog taga odredjenog tipa, ostale iste tagove u kodu ce ignorisati

Postoje i negacijske pseudoklase: :not(.bucky) {color:red} – znaci svi elementi kojin nemaju klasu bucky ce biti obojeni crveno

komentari u css-u se pisu: /\* komentar \*/

!important - se stavlja u okviru css deklaracije, pre ; i govori browseru da to pravilo ima prednost u odnosu na neko drugu drugu deklaraciju koja se odnosi na istu stvar. Elemenat sa ovim atributim se ne menja ako mu kasnije u kodu zadamo druge osobine

Kada stavljamo prefikse to radimo sledecim redom:

-webkit-

-moz-

-o-

Bez prefiksa

VREDNOSTI

px - piksel

% - procenat u odnosu container, odnosno roditeljski element

pt - point /tipografska vrednost/

em - predstavlja velicinu fonta nasledjenu od “starijeg” elementa

rem - predstavlja velicinu fonta html-a (osnovnu, bazicnu velicinu)

za boju: naziv (black, red itd) ili: rgb(0, 0, 255), #0000ff

NAVIGACIJA

Linkovi: imaju pseudo selektore: a:link{css} ovde moze da stoji i a{css}, a:visited{css} - poseceni link, a:hover{css} - kada smo kursorom iznad linka, a:active{css} - kliknut link /trebalo bi pratiti redosled: :link, :visited, :hover, :active u definisanju css-a, da bi sve funkcionisalo kako treba/

Oni se takodje mogu zajedno definisati: a:link, a:visited{css} isto se odvajaju zarezom

a:hover { border-bottom: #boja solid Xpx; } - fora za hover efekat - crtice u boji ispod linka

mozemo staviti da boja dugmeta odgovara stranici na kojoj smo, za to je potrebno dati id selektor <body> tagu: <body id=”about”> i klasu svakom od linkova ili <li> elemenata: <a class=”about”>about</a> i <a class=”contact”>contact</a> a u css-u definisemo body#contact a.contact, body#about a.about{css1} - za jedan stil linka koji odgovara toj strani i: body#contact a.about, body#about a.contact{css2}- za drugi stil

Horizontalni meni pravimo sa listom a mozemo da <li> elementima damo: display:block i float:left - i tada ce se slagati jedan uz drugi ili display:inline i tada ce se isto ponasati

Mozemo da koristimo jednu sliku za sva rollover stanja, samo da je pomeramo: a:hover{background-position: 0 -20px;}

:hover pseudo-selektor moze da se primeni i na druge elemente osim linkova, recimo da boji redove tabele kako pomeramo kursor po njoj /radi na mozili i IE9, ne znam na kojima ne radi/

TEKST

text-decoration:none -obican tekst, najcesce se primenjuje se npr. da linkovi ne budu podvuceni crtom, mada se ovaj atribut moze primenjivati na bilo kakav txt

underline - podvucen txt, overline - linija je iznad teksta, line-through - precrtan txt, mozemo i kombinovati vrednosti: text-decoration: underline overline;

font-weight:bold - podebljan txt, vrednosti koje jos postoje su: normal, lighter, bolder ili u brojevima od 100 do 900 u koracima po 100 /100, 400, 800 itd./, /normal=400, bold=700/

font-style:italic - iskosen txt

text-align:center - centriran tekst ili right, left, justify - poravnat

color:boja - boja texta

font-size:Xpx - font velicine X piksela, moze i u nekoj tipografskoj meri kao: pt, em ...

font-family:naziv fonta - definisanje odredjenog fonta koji ce biti prikazan na stranici, a mozemo i definisati *fallback* fontove ili familije fontova koje ce browser prikazati ako ne moze prvi po redosledu: font-family: Arial, Helvetica, Sans-serif; /odvajamo ih zarezima/, postoji 5 glavnih genericnih grupa fontova koje bi trebalo stavljati na kraju ove sintakse: sans-serif, serif, fantasy, cursive i monospace zavisno od toga kakav font nam treba

text-indent:Xpx - uvucen tekst za X piksela na pocetku paragrafa

Ako cemo ceo paragraf da uvucemo, mozemo ovako: padding-left:Xpx

line-height: 1 - rastojanje izmedju base line slova u paragrafu gde 1 odgovara velicini slova(a moze i u % gde je 100% = 1)

Za vertikalno poravnjavanje txta unutar diva: line-height: Xpx gde je: Xpx onoliko kolika je visina diva

font-variant: small-caps - ispisuje tekst velikim slovima koja imaju velicinu malih

text-transform: uppercase - ispisuje txt velikim slovima, text-transform: lowercase - ispisuje txt malim slovima, a imamo i: capitalize - ispisuje txt sa velikim prvim slovom svake od reci.

paragraf ima pseudo selektor za prvu liniju: :first-line

BACKGROUND

Background obuhvata sadrzaj elementa, padding i border a ne obuhvata margine, izuzetak su IE6 i IE7 kod kojih background ne obuhvata border.

background-color:boja - pozadinska boja elementa ili strane

boja se moze definisati na vise nacina, recimo plava: blue ili rgb(0, 0, 255) /ne sme da ima razmak izmedju rgb i ( / ili #0000ff a boja moze da bude i transparenta: rgba(10, 180, 255, 0.56) - 0.56 je 56% transparentna, 1 je 100%, tj nimalo transparentna

background-color:transparent - pozadina moze biti i providna ?

background-image: url (“image.jpg”) - postavljanje slike kao pozadine /po defaultu slika ce se ponavljati dok ne popuni element ako je manja/

a moze izgleda i: background: url (“image.jpg”)

mozemo zadati da pozadina bude i slika i boja pa ono sto ne pokrije slika ce biti u zadatoj boji

background-repeat: repeat; - default podesavanje slika ce biti ponavljana i po x i po y osi

background-repeat: no-repeat; - slika se nece ponavljati, bice samo jedna

background-repeat: repeat-x; - slika se ponavlja samo po x osi / horizontalno

background-repeat: repeat-y; - slika se ponavlja samo po y osi / vertikalno

background-repeat: inherit; - ponavljanje ce biti isto kao kod roditelja elementa

background-repeat: space;- slika se ponavlja sa jednakim razmacima dok ne popuni element

background-repeat: round;- slika je smanjena tako da poslednja ponovljena cela stane u element

ove 2 komande su css3 i trenutno ne rade na firefoxu

background-position:Xpx Ypx - pozicija slike izrazena u rastojanju od gornjeg levog ugla, ove vrednosti mogu imati negativnu vrednost, sliku mozemo pomerati kako god zelimo

umesto u pikselima mozemo definisati i recima, za X osu: left, center, right. za y osu: top, center, bottom moze biti i u %: background-position: 100% 50% - slika ce biti postavljena uz desnu stranu na polovinu elementa. Ako n

background-attachment:fixed - slika stoji fiksirana na svojoj poziciji na ekranu kad skrolujemo stranicu, po defaultu je: scroll i tada se slika skroluje zajedno sa ostalim sadrzajem

skraceno mozemo sve ove vrednosti da napisemo ovako: background: <color> <image> <position> <attachment> <repeat> ne moramo da pisemo sve od ovih atributa, onaj koji preskocimo imace default vrednost i ne razdvajamo ih , ili ;

background-size: contain - smanjuje/povecava sliku dok cela ne stane u elemenat zadrzavajuci odnos sirina:visina background-size: cover - skalira sliku dok ne pokrije elemenat zadrzavajuci odnos sirina : visina, a mozemo velicinu pozadinske slike izraziti u px ili % sa ovim atributom

trik da txt bude u nekom custom fontu a search engine da ipak moze da ga procita: pozadini elementa sa txt-om u css-u damo da bude slika txta u nasem fontu, a tekst pomerimo skroz levo da se ne vidi: text-indent: -99999px;

sa css3 cemo moci da stavimo vise slika u pozadinu: background-image: url(gornja.jpg), url(srednja.jpg),

url(donja.jpg); bice poredjane jedna preko druge

tako mozemo da imamo potamnjenu sliku koristeci ovaj trik:

background-image: linear-gradient( rgba(0, 0, 0, 0.5), rgba(0, 0, 0, 0.5)),

url(02.jpg);

u sustini preko slike stavljamo gradient koji koristi sa obe strane istu boju, koja je transparentna

trik da imamo tekst sa blurovanom pozadinom preko neke slike, a da moze da se skroluje je:

elementu sa slikom damo:

background-image: url();

background-attachment: fixed;

position: relative;

a blok teksta:

position: absolute;

background: inherit;

overflow: hidden;

zajedno sa pseudoelementima: before i after, gde jedan bluruje a drugi zatamnjuje:

:before{

content: "";

background: inherit;

position: absolute;

filter: blur(20px);

}

:after{

content: "";

position: absolute;

background: rgba(0, 0, 0, 0.25);

overflow: hidden;

}

Potrebno je i korigovati im pozicije i velicinu da se lepo poklope

GRADIENTI

Se pisu kao background: linear-gradient(#fff,#000);sa vendor prefixima po potrebi, ili kao background-image:

Po defaultu ce gradient ici odozgo na dole, ako zelimo da dodamo pravac,pre boja unosimo: (left,#fff,#000) - sa leva na desno, (bottom,#fff,#000) – iz donjeg levog coska: (bottom left,#fff,#000) itd.

Naravno, nismo ograniceni sa samo 2 boje mozemo da ih trpamo do mile volje

Poziciju pune boje podesavamo: ,#fff 25%,#000 u % ili px

Radial: radial-gradient da nam gradient budi pravi krug a ne razvucena elipsa, u slucaju recimo da su razlicite sirina i visina: radial-gradient(circle,#fff,#000); da nam gradient ide do najblize ivice elementa: (contain circle, #fff, #000); ako hocemo da ide do dalje ivice:

(farthest-side, #000, #fff, #000); a da zadrzi oblik kruga: (circle farthest-side, #000, #fff, #000);

LAYOUT

Mozemo menjati prirodu elementa /da se ponasa drugacije nego obicno/ sa:

display:block; ili inline

display:none; mozemo koristiti za sakrivanje elemenata, stranica ce biti renderovana kao da element ne postoji za razliku od visibility:hidden;

float:left ili right - elemenat se pribija uz levu/desnu stranu / Ako hocemo recimo da nam tekst ide oko slike, a slika da bude postavljena desno: <img style="float:right" src="2.gif" /> /

float moze imati i vrednosti: inherit - nasledjuje float od roditeljskog elementa/ne radi u nekim verzijama IExplorer-a/ i none - default vrednost

uz float bi trebalo i definisati sirinu elementa da bi izbegli probleme koji mogu nastati

clear:left, right ili both - elemenat se postavlja ispod float elementa koji mu je sa leve/desne/bilo koje strane - primenjujemo ako imamo 2 elementa kojima je zadat float u istu stranu pa stoje jedan do drugog vertikalno /po defaultu/, ovo sluzi da bi bili jedan ispod drugog

trebalo bi ispod bilo kog elemenata sa float-om ubaciti prazan div koji ce imati neku klasu tipa .clear za koju cemo u css-u definisati samo: .clear{clear:both;} /i to bi trebalo da bude u okviru container-a u kojem su float elementi/ i to iz vise razloga

a jos jedan nacin za resavanje ovog problema je: da u css-u wrap-eru koji sadrzi elemente sa float-om damo: overflow:hidden;

position:absolute ili relative ; left:Xpx; top:Ypx - pozicija elementa

Imamo: static – nepozicioniran element /default vrednost/, relative – ponasa se slicno kao i static s tim da ga mozemo pomerati sa marginama ili sa: top, right, bottom i left i ako ga pomeramo sa marginama drugi elementi nece popuniti upraznjen prostor, fixed – fiksira elemenat u odnosu na prozor browsera, i ne skroluje se, isto mozemo koristiti: top, right, bottom, left i na kraju: absolute koji se ponasa slicno fixed s tom razlikom da se pozicionira ne prema prozoru browser-a vec prema svom prvom pretku koji je pozicioniran /ima bilo koji position osim static, ako ne postoji takav, onda u odnosu na body/

width: height: - visina, sirina - moze biti izrazena u pikselima ili % velicine ekrana, a moze biti i auto

border:Xpx - okvir X pixela solid - normalni dotted - tackasti

border mora da ima definisan tip linije (solid, dotted, dashed, double) inace se nece videti.

Pored njih imamo: inset - kao utisnut okvir, outset - kao izbacen, groove - spoj ove 2, ridge - isto, samo u obrnutoj kombinaciji, pored njih imamo i none i hidden.

Pise se u formatu: border: 1px solid #000000; i primenjuje se na tabele, slike, txt, container element itd.

Border-color: -podesavamo boju bordera

padding: Xpx - unutrasnja margina(kod bordera - rastojanje izmedju bordera i elementa),

margin: Ypx - spoljasnja margina

margin: 3px 6px 3px 5px; - skraceni zapis u redosledu: top right bottom left

mozemo posebno svaku od 4 margine da podesavamo npr: padding-top:4px; margin-left:10px

i za padding i za margine mozemo ovako definisati vrednosti: padding: Xpx Ypx; gde je x gornja i donja a y leva i desna strana ili cak: padding: Xpx Ypx Zpx; x gornja, y leva i desna strana i z donja

margin: 0 auto; - css trik za centriranje sadrzaja, ali bi trebalo definisati i: width: Xpx;

za responsive sajtove umesto width: Xpx;koristimo max-width: Xpx;

margin moze imati negativnu vrednost, padding ne } /mozemo negativnu vrednost donje margine koristiti za smanjivanje rastojanja izmedju headinga i paragrafa/

Elemenat moze imati maksimalnu i minimalnu visinu i sirinu npr: max-height:350px; min-width:300px

j

<hr> tagu mozemo definisati: border, height, width, color.

Sa css-om mozemo i da menjamo kursor /The Css Anthology 101 Esential…pdf 103. strana/

Vise kolona mozemo praviti sa definisanom sirinom, levom ili desnom marginom i float:left, samo treba sracunati da se sve uklopi u okviru elementa u kome se nalaze

FLEXBOX

PODESAVANJE CONTAINERA

elementi sa display:flex su containeri, moze biti i display:inline-flex, u prvom slucaju imamo block-level conatiner a u drugom inline container.

Njihovi direktni potomci (deca) su flex elementi koji se slazu u container, po x i y osi, (po defaultu levo-→desno i gore→dole)slaganje po osama se moze menjati u css-u. Po defaultu x je glavna, a y poprecna osa, a ako elemente redjamo vertikalno, onda je y glavna osa.

smer redjanja elemenata mozemo podesavati sa:

flex-direction:row - (default vrednost) elemeti su poredjani horizontalno sleva na desno,

row-reverse - elemeti su poredjani horizontalno sdesna na levo - ovo menja flex start(desno) i flex end(levo),

column - redjanje po vertikali odozgo,

column-reverse - redjanje po vertikali odozgo, isto menjaju flex start i end.

flex-wrap:

nowrap - (default) - elementi popunjavaju prostor duz glavne ose iako prelaze container

wrap - kad popune prostor duz glavne ose elementi se nastavljaju u novom, redu (koloni)

wrap-reverse - redovi(kolone) idu u suprotnom pravcu

flex-flow:[flex-direction] [flex-wrap] - skracenica koja kombinuje flex-direction i flex-wrap, npr. flex-flow: column-reverse wrap;

justify-content - sluzi za rasporedjivanje praznog prostora po glavnoj osi imoze da ima vrednosti:

flex-start - (default) elementi su poredjani od pocetka containera

flex-end - elementi su poredjani od kraja containera

center - elementi su centrirani

space-between - elementi imaju isti medjusobni razmak, a elementi na pocetku i kraju su pomereni do ivice

space-around - podjednak razmak izmedju i oko elemenata, prazan prostor koji ovako rasporedujemo je onaj koji ostane ako elementi

align-items - se koristi za razvlacenje po poprecno osi i moze da ima vrednosti:

flex-start - elementi su poravnani po pocetku poprecne ose

flex-end - elementi su poravnani po kraju poprecne ose

center - elementi su centrirani

baseline - elementi su centrirani po njihovim baseline-ima

stretch - (default) elemeti su razvuceni da popune ceo container po poprecnoj osi ali postujuci min, max, width, height

align-content: sluzi za rasporedjivanje po poprecnoj osi (nema efekta kad imamo samo 1 red(kolonu))

flex-start - (default) linije su poredjane od pocetka flexbox container-a dokle ih ima

flex-end - linije su poredjanie od kraja

center - linije su centrirane, ali spojene

space-between - linije imaju isti medjusobni razmak s tim da su razvucene tako da prva bude na pocetku containera a poslednja na kraju

space-around - linije su ravnomerno rasoredjene sa jednakim razmakom izmedju njih (sto znaci da ce imati duplo veci medjusobni razmak izmedju elemenata nego od krajeva)

PODESAVANJE CHILD ELEMENATA

flex-grow: - definise koliko ce elemenat da raste, da bi popunio slobodan prostor, definise se brojevima koji predstavljaju proporciju (ako svi elementi imaju 1 svi ce rasti isto ako npr elementu zadamo 2 zauzece duplo vise slobodnog prostora od ostalih) , 0 znaci da ne raste.

flex-shrink: - definise koliko ce elemenat da se smanji, ako nema dovoljno prostora, definise se brojevima isto kao flex-grow, 0 znaci da se ne smanuje.

flex-basis: - definise minimalnu velicinu elementa, ako nije definisano default vrednost je 0px

ove 3 vrednosti se mogu pisati i skraceno: flex: 1 0 200px;

browser ce prvo odvojiti prostor za margine i padding pa onda raspodeliti px-ele po flex elementima i to prvo im dodeliti velicinu po basis a onda ostali slobodan prostor rapodeliti u skladu sa proporcijom

order: - (ceo)broj koji definise redosled elemenata (1, 2, 3, itd. kao i -3, -1, 0)

align-self - podesava align po poprecnoj osi posebno za svaki od flex elemenata, vrednosti su iste kao kod: align-items

LISTE

list-style-type: - definise stil liste, a moze biti: none - nista /koristimo kad pravimo meni od liste/, circle - okruglo dugme, disc - ispunjeno okruglo dugme, square - kvadratic, decimal - redni brojevi, lower-alpha - mala slova, upper-alpha - velika slova, upper-roman - veliki rimski br., lower-roman - mali rimski br., decimal - 1. 2. 3., decimal-leading-zero - 01. 02. 03., moze cak i slika: list-style-image: url (slike/slika.png) /ovo pravilo mozemo dati i li i celoj listi i u tom slucaju li elementi ce naslediti tu osobinu/

i moze se primenjivati i na unordered (<ul>) i na ordered (<ol>) listu.

Ako <ol> i <ul> listama stavimo padding:0, /kada recimo radimo css reset/ - onda ce brojevi i tackice naci van okvira ovih elemenata /ako zadamo border ili backgound-color, bice izvan okvira/ i u slucajevima kad su nam te liste na levoj strani ekrana - nece se videti,

to se popravlja komandom: list-style-position: inside;

TABELE

css za tabelu: border okvir oko tabele ili elemenata tabele, naravno koristi se kao i za ostale elemente /iz nekog razloga border ne radi za <th>/

border-collapse: collapse; - popravlja ruzni izgled tabele gde svaka celija ima okvir oko sebe sa razmakom izmedju tih okvira, ova komanda brise te razmake pa tabela izgleda normalno iscrtana tj. susedni elementi dele border. Suprotno tome default vrednost je odvojeni border-i: border-collapse: separate; definise se za border

za caption table{caption-side:bottom ili top} /naslov tabele/

mozemo recimo dati klase redovima i bojiti ih pomocu tih class selektora, definisati fontove u tabeli

TRANZICIE, TRANSFORMACIJE I ANIMACIJE

TRANZICIJE

Za tranzicije koristimo najcesce :hover pseudoklasu u kojoj definisemo kako ce elemenat izgledati kad preko njega predjemo misem

Tranzicija se pise u formatu: transition: <osobina> <trajanje> <vrsta prelaza> <delay>; kao npr: transition: background-color 2s linear 100ms; Jedina zahtevana vrednost u skracenici je trajanje.

Easing: linear - ista brzina cele animacije/tranzi–cije, ease-in - usporena na pocetku, ease-out - usporava na kraju, ease-in-out - usporava na kraju i pocetku, ease - skoro isti kao ease-in-out

Umesto reci mozemo da koristimo i custom definisanu krivu: cubic-bezier(0.680, -0.550, 0.265, 1.550); -postoje web alati za njeno kreiranje: http://cubic-bezier.com i http://matthewlein.com/ceaser/

Elementu mozemo definisati razlicite tranzicije za razlicite osobine:

.elemenat {

transition-property: border-radius, background-color;

transition-duration: 2s, 0.5s;

transition-timing-function: linear, ease;

transition-delay: 0s, 1s;

} ili:

.elemenat {

transition: border-radius 2s linear 0s, background-color 0.5s ease 1s;

}

TRANSFORMACIJE

transform:translate(x,y); - pomeranje po x i y osi, a mozemo i:

translatex(x); ili translatey(y);

transform:rotate(30deg); - rotiranje: 30°

transform:scale(1.2); - povecanje velicine za 20% /ako stavimo negativnu vrednost, npr. 1 element ce se smanjiti pa ce se povecati na pocetnu velicinu, samo ce biti naopak/

transform:scale(1.2,5); - neproporcionalno povecavanje sirine i visine

ako zelimo da primenimo vise transformacija odj ednom, onda ih pisemo jednu iza druge:

transform: scale(1.8) rotate(90deg);

transform: matrix(1, 0, 0, 1, 0, 0); - sa ovakvim vrednostima element se nece promeniti

prva 1 menja sirinu, druga visinu, ako ih promenimo u istu vrednost elemenat ce se proporcionalno smanjiti/povecati, 2. i 3. broj prave skew i to: 1. po bocnim stranama, 2. po horizontalnim, ako im damo istu vrednost samo jednom negativnu a drugom pozitivnu elemenat ce se rotirati. 3. i 4. vrednost pomeraju elemenat po x, odnosno y osi /njih 2 preedstavljaju px a ove ostale bi trebalo da budu negde u rasponu od -2 do 2/ matrix(1, 0.2, -0.2, 1.1, 50, 100)

transformacije se mogu primeniti na elemente pri ucitavanju stranice a i na :hover pseudoelement

ANIMACIJE

Animacija se definise kao svojstvo na element koji zelimo da primenimo h1{ animation-name: njeno\_ime;} /s tim da moramo da pazimo na prefikse, narocito -webkit-/

Pored toga mozemo da definisemo i:

animation-delay:3s; - odlaze izvrsavanje animacije, izrazava se u s ili ms

animation-direction:normal; - pravac animacije u ovom slucaju od prvog do poslednjeg frame-a tj. Od 0%-100%, reverse; - animacija u obrnutom pravcu: 100%-0%, alternate; - prvo 0%-100% a zatim od 100%-0%, alternate-reverse; - funkcionise isto samo je prvo 100%-0%. oba zahtevaju vise od 1 ponavljanje animacije

animation-duration:3s; - trajanje animacije

animation-play-state:running; - broj ponavljanja animacije, od 1 do beskonacno

animation-timing-function: ease-in; - podesava easing animacije

animation-fill-mode: forwards; - elemenat preuzima osobine od zadnjeg keyframe-a posle animacije, backwards; - elemenat uzima osobine od 1. Keyframea pre i posle animacije i both; - preuzima osobine od 1. Keyframe-a pre i poslednjeg posle animacije.

animation-iteration-count: infinite; - ponavljanje animacije moze biti od 1 do beskonacno.

Skraceno sve ove vrednosi mozemo zapisati: animation: [name] [duration] [timing-function] [delay] [iteration-count] [direction] [fill-mode] –naravno mozemo izostaviti neke vrednosti koje ce biti po defaultu ali se moora paziti na redosled

keyframes se definisu u zasebnom bloku css-a

@keyframes ime\_animacije{

from{}

to{}

}

– definisemo keyframes-e, treba nam pocetni i zavrsni

A mozemo i sa procentima:

0%{}

20%{}

100%{}

vrednosti i keyframesi trebalo bi da imaju prefiks

@-webkit-keyframes ime\_animacije{

0%{-webkit-transform: translateX(0px);}

100%{-webkit-transform: translateX(400px);}

}

Keyframes se moze animirati na vise elemenata kojima mozemo da zadamo istu animaciju sa razlicitim osobinama animacije.

from{-webkit-transform: translateX(0px);}

to{-webkit-transform: translateX(400px);}

– pomera elemenat u desno 400px, od pocetne pozicije, isto funkcionise i sa Y osom

from{opacity:0;} – pocinje animaciju sa transparentnim elementom, pored ovoga mozemo zadati i druge css vrednosti: background-color, font-size …

mozemo elementu zadati: animation-play-state: paused; a njegovom :hover – u : animation-play-state: running; - i tako ce se animacija odvijati samo kad smo misem iznad elementa, u suprotnom bice pauzirana onde gde je stala.

box-shadow: Xpx Ypx Zpx Qpx boja - postavlja senku oko elementa: - X - pozicija senke po X osi, Y - pozicija senke po Y osi, ako su obe 0 onda je senka poravnata sa elementom, ako su negativne vrednosti senka se pomera ka gore i levo, Z - velicina senke, Q - jacina senke /kao spread u Photoshopu/ Z ne moze a Q moze imati negativnu vrednost. boja - boja definisana na neki od nacina, ako stavimo: box-shadow: inset 0 0 24px - senka ce biti unutrasnja

border-radius: Xpx Ypx Zpx Qpx; - zaobljene ivice: X - gornja leva, Y - gornja desna, Z - donja desna, Q - donja leva. Ovo sve mozemo definisati i posebnim komandama: border-top-left-radius: Xpx; border-top-right-radius: Ypx; border-bottom-right-radius: Zpx; border-bottom-left-radius: Qpx;

text-shadow: Xpx Ypx Zpx boja - senka za tekst: - X - pozicija senke po X osi, Y - pozicija senke po Y osi, Z - velicina senke, boja - boja senke. Sa ovim mozemo da napravimo trik da txt izgleda utisnuto tako sto cemo senci dati neku svetlu boju.

cursor:pointer; kada misem predjemo preko elementa, kursor se menja u onu rukicu za linkove

SVG

koristimo inline svg

fill i stroke

viewbox

mozemo Ctrl + A u Illustratoru, Ctrl + C - Ctrl + V u code editor-u

u css-u mozemo da podesavamo svg:

fill:#fff;

stroke-width:1px;

RESPONSIVE

<meta name="viewport" content="width=device-width, initial-scale=1"> - da bi se na telefonovima ucitavala mobilna verzija a ne da bude u punoj sirini kao na desktopu

<picture>

<source srcset=”img/logo.png, img/logo\_retina.png 2x”>

<img class=”logo” srcset=”img/logo.png, img/logo\_retina.png 2x” alt=””>

</picture>

Ovaj kod bi trebao da funkcionise tako da u slucaju da se strana pregleda na retina ekranu, ucita vecu sliku.

Podrska media queries na starijim browserima?

Vellicina fontova na razlicitim uredjajima?