JQUERY

preporucljivo je da stavimo link ka google-ovom hostu Jquery-ja, zbog performansi, u krajnjem slucaju mozemo da hostujemo svoju verziju za slucaj da ne uspe da downloaduje od googla a to radimo sa:

<script src="//ajax.googleapis.com/ajax/libs/jquery/1.6.2/jquery.min.js"></script>

<script>window.jQuery || document.write('<script src="js/jquery-1.6.2.min.js"><\/script>')</script>

jquery se pise u <script> tagu

a moze i u posebnom .js file-u pa ga ucitamo sa <script src="custom.js"></script>

$ - znak, koristi se za pozivanje jquery funkcije

$(selektor) - u zagrade se stavlja selektor na koji se primenjuje jquery funkcija

$("p").hide() - kada selektujemo elemenat na njega primenjujemo akciju, u ovom slucaju sakrivamo sve paragrafe

ako vec smo selektovali neki elemenat u funkciji mozemo da stavimo: $(this)

selektor moze biti i klasa u klasi (id-iju): $("#pop .brada")

$("ul li") - selektuje li elemente unutar ul

$(".selektor, .selekcija") - selektuje sve elemente sa selektor i selekcija klasom

$("ul li:nth-child(2)") - selektuje drugi li element

$(".info:first") - selektuje prvi element sa klasom: info (moze i :last ili :even, :odd)

$("li:not(.info)") - selektuje <li> elemente koji nemaju klasu info

U sustini funkcionise isto kao css selektori

$("a[target]") - selektuje elemente sa odredjenim atributom

$("a[target='\_blank']") - selektuje elemente kojima je atribut \_blank

$("a[target!='\_blank']") - selektuje elemente koji nemaju atribut \_blank

$("a[class^='ma']") - selektuje elemente cija klasa pocinje sa: ma npr max, ma-2 itd

$("a[atribute$='value']") - selektuje elemente sa odredjenim atributom cija vrednost se zavrsava sa zadatom vrednoscu

$("a[atribute\*='value']") - selektuje elemente sa odredjenim atributom cija vrednost sadrzi zadatu vrednost

$("a[class][id]") - selektuje elemente koji imaju sve atribute iz filtera, u ovom slucaju: definisanu klasu i id

$("div:contains(milina)") - selektuje sve div-ove koji sadrze tekst milina ili ciji “potomci” sadrze

$("div:parent") - selektuje sve div-ove koji sadrze neki element

$("div:has(.nova)") - selektuje sve div-ove koji sadrze element sa klasom nova (moze biti i bilo koji drugi selektor, ne mora klasa)

$('\*:empty') - selektuje sve elemente koji su prazni tj. ne sadrze nista

Za formulare, imamo ceo set pseudoselektora:

$('form :input'), :text, :password, :reset, :button, :radio itd.

Takodje mozemo i:

$('form :text:enabled'), kao i: :disabled, :checked, :selected

$(document).ready(function(){

//code goes here

});

-osnovna funkcija jquery-ja koja ceka na ucitavanje DOM-a da bi jquery mogao da se primeni, moze i ovako:

$(function(){

//code goes here

});

U nju stavljamo ostale f-je, npr:

$('a').click(function(){

alert('you clicked');

});

pored ) .click() imamo i druge event handlere: .dblclick() - dvoklik, .hover(), .mouseout()(mada je prakticnije kada koristimo .mouseLeave() kada imamo i child elemente - kada sklonimo mis sa elementa, .mouseOver(),.mouseEnter()itd.

mozemo elementu zadati nestane i ponovo se pojavi: $('.elemenat').hide(1000).show(500); - ovo ce ga sakriti kroz 1s a zatim prikazati kroz 500ms

pored ovih imamo i druge efekte: .slideUp(),.slideDown(), .fadeIn(), .fadeOut() a imamo i: .delay(500), napravice delay od 500ms

mozemo i da stavimo: .fadeToggle(500).fadeToggle(500) - sto ce ga fadeout-ovati a zatim fadein-ovati, sve kroz 1s

.fadeTo(‘fast’, 0.25) - fade-uje brzo (moze i u ms), do 0.25 opacity

mozemo i da menjamo sadrzaj html tagova, u ovom slucaju one koji imaju klasu foo:

$('.foo').html(‘novi sadrzaj’); , u novi sadrzaj mozemo da stavljamo i tagove

mozemo da “slusamo” odradjene evente pa da reagujemo na njih:

$('.foo').on('click', function() {

$('.panel').fadeToggle(800);

});

u ovom slucaju, kad kliknemo na .foo - panel ce nestati (fade-ovati) a kada opet kliknemo ponovo ce se pojaviti

.addClass() - dodaje odredjenu klasu selektovanom elementu, moze i vise njih, takodje imamo i: .removeClass(),

.appendTo(‘element2’) - dodaje selektovani elemenat, elementu2, ako hocemo da pri tom kloniramo:

.insertAfter("div.car\_well:last"); - ubacuje elemenat posle poslednjeg sa klasom: car\_well

.clone().appendTo(‘element2’); a ako hocemo da elemenat sadrzi i event handlere vezane za njega: .clone(true)

.children(li) - metod koji pronalazi child elemente, u zagradi mozemo da specifisemo tacno koje, u ovom slucaju li ali i ne moramo

.children(‘li:first’) - metod koji selektuje samo prvi li element potomak, a to isto mozemo i sa:

.children(‘li’).first()

.first() - selektuje prvi elemenat u vec selektovanom nizu

To isto mozemo i sa: .eq(0) - prvi elemenat, .eq(3) - cetvrti elemenat

.find()- trazi potomke selektovanog elementa,ne samo decu kao u prethodnom metodu

.next() - selektuje sledeci elemenat u nizu u odnosu na vec selektovani

.prev() - selektuje prethodni elemenat u nizu u odnosu na vec selektovani

.parent() - selektuje direktan parent elemenat selektovanog elementa

.parents(‘.container’) - selektuje “starije” elemente u odnosu selektovani, koji ima klasu .conatiner, i to ce ici sve do vrha DOM TREE ako ih ima vise, dok

.closest(‘.container’) ce selektovati blizi, isto penjuci se uz DOM TREE

.css() - dodaje css osobine: .css(‘background-color’, ‘#FFF’, a ako cemo vise vrednosti:

.css({‘background-color’: ‘#FFF’, ‘font-size’:’20px’});

.text(“novi tekst”) - unosimo tekst unutar selektovanog taga

.height(100) - menja visinu elementa u 100px

.width(50) - menja sirinu elementa u 50px

$(document).ready(function(){

$('div').mouseenter(function(){

$('div').fadeTo('fast', 1);

});

$('div').mouseleave(function(){

$('div').fadeTo('fast', 0.5);

});

});

JAVASCRIPT

Varijable mogu biti brojevi, text(string),

prompt() - popup sa poljem za unos

var name = prompt() - ono sto unesemo ce postati vrednost variable name

ako imamo prompt(‘what’s your name?’) onda ce to pitanje stajati u pop-up-u

alert(variabla) - popup na kom ce pisati vrednost variable

mozemo da ove stvari vezujemo jednu iza druge:

var name = prompt(); alert(name);

poredjenja vrednosti: >, <, <=, >=, == (jednako), != (nije jednako)

ako imamo ‘ u stringu: ‘can’t go’ - mozemo: “can’t go” ili da “escapujemo” apostrof:

‘can\’t go’

Variable je potrebno samo jednom definisati kao varijablu: var age = 15, posle mozemo samo da joj menjamo vrednost: age = 20

If: if (uslov) {ako je ispunjen uslov} else {ako nije}, u else mozemo da imamo dodatne if uslove

FUNKCIJE: function naziv(){ono sto radi funkcija}, sama po sebi f-ja se ne pokrece a mozemo da je pokrenemo pozivajuci je u kodu: naziv();

ARGUMENTI F-JE:

function go(ime, uvreda) {

alert(ime);

alert(uvreda);

}

Ovde zadajemo 2 parametra pa kada ih kasnije prosledimo f-ji pri njenom pozivanju: go('picko', 'jedi govna'); ona se izvrsava

return - vraca rezultat f-je i prekida njeno izvrsavanje, takodje

NPM

npm init --y – kreira package.json file u folderu sa default vrednostima, ovde mozemo da unesemo dependencies tako da prekopirani fajl I komanda npm install ih instalira na recimo drugom kompjuteru

unutar package.json mozemo da izmenimo:

"scripts": {

"start": "webpack && start index.html"

},

Sto ce kada ukucamo npm start u terminalu pokrenuti webpack komandu i otvoriti index.html u browseru

-D je shortcut za --save-dev a i za -install tako da mozemo:

npm install paket --save-dev ili npm i paket -D /s tim da D mora velikim slovima

GULP

prvo definisemo potrebne module unutar gulpfile.js:

var gulp = require( ‘gulp’ ),

var uglify = require( ‘gulp-uglify’ ),

zatim definisemo taskove, npr.:

gulp.task (‘scripts’, function(){

//code goes here

});

tada mozemo iz terminala da pozivamo zeljenu komandu sa gulp scripts

i posle njega mozemo da definisemo watch task:

gulp.task (‘watch’, function(){

gulp.watch(‘app/js/\*\*/\*.js’, [‘scripts’]);

});

koji ce paziti na promene u zadatim fajlovima i po promeni izvrsavati task scripts

Default task koji se definise na samom kraju, obuhvata sve taskove i iz terminala se pokrece sa: gulp

gulp.task (‘default’, [‘scripts’, ‘watch’]);

se stara da pokrene sve zadate definisane u default tasku

kada instaliramo gulp u nekom projektu: npm install gulp, gulp kao i pluginovi instalira se u folder node\_modules

pre toga mozemo i da kreiramo package.json file sa: npm init ili da ka prekopiramo ili sami napisemo

kada instaliramo potrebne gulp plug-in-e ili sam gulp: npm install gulp --save-dev, npm install gulp-uglify --save-dev onda oni bivaju zapisani u package.json file-u, sa svojom verzijom

mozemo i da ih upisemo u package.json pa onda npm install u terminalu

BOWER

bower list --path - izlistava nam lokacije svih bower komponenti pa mozemo odatle da prekopiramo path u neki fajl

WEBPACK

npm install webpack –g – instalira webpack globalno na masini, tako da mozemo da koristimo njegove komande

npm i -D webpack – instalira webpack u folderu, dodajuci dependecy u package.json file

npm install webpack –save-dev – bi trebalo da radi istu stvar

webpack.config.js - file bi trebalo da stoji u root-u i u njemu su podesavanja webpack-a i u njemu:

module.exports ={

entry: './src',

output: {

path: 'build',

filename: 'bundle.js',

},

};

Ovde smo definisali da webpack proverava src folder za sve dependecies i kompajlira ih u bundle.js

U index.js stavljamo dependecies, npr:

require('./alert.js');

ovo ce pri kompajliranju povuci javascript iz alert.js file-a i kompajlirati ga u bundle.js

webpack - komanda za pokretanje webpacka, kompajlira fajlove koje smo mu zadali u webpack.config.js

LOADERS su pluginovi za webpack koji se koriste za citanje .scss .css .ts ili drugih fajlova, posto webpack manipulise sa javascriptom oni omogucavaju rad i sa non-javascript fajlovima

npm i -D babel-loader babel-core babel-preset-es2015 instaliramo loader za BABEL koji transkompajlira novu js sintaxu (ES2015 ili ES6) u staru. Treba jos da napravimo .babelrc u rootu u kom:

{

"presets": ["es2015"]

}

A onda u webpack.config.js dodamo:

module: {

loaders: [

{

test: /\.js/,

loader: 'babel',

include: \_\_dirname + '/src',

}

],

}

Ovde test sluzi da selektuje samo .js fajlove a include u kom folderu zelimo da to radi /da ne bi dirao druge foldere kao npr node\_modules