

# Agenda kursa

Core JavaScript – php

**ITAcademy**

# Agenda kursa

- **Uvod u JavaScript**
  - Šta je JavaScript
  - Čemu služi JavaScript
  - Gde se pojavljuje JavaScript
  - Kako se implementira JavaScript
  - Osnovne funkcije JavaScript-a
- **DOM**
  - Navigacija po DOM dokumentu
  - Manipulacija DOM objektima
  - Stilizacija DOM objekata
- **Forme i kontrole**
  - Događaji
  - Validacija

# Uvod u JavaScript

Core JavaScript – php

**ITAcademy**

# Agenda

- Uvod u JavaScript
  - Šta je JavaScript
  - Čemu služi JavaScript
  - Alati za rukovanje
  - Gde se pojavljuje JavaScript
  - Kako se implementira JavaScript
  - Osnovne funkcije, svojstva i objekti u JavaScript-u

# Šta je JavaScript

- Programski jezik
- Derivat C-a
- Multifunkcionalan je
- Najčešće se koristi u izradi klijentskih delova web aplikacija
- Nije isto što i Java
- Dijalekti se razlikuju u odnosu na pretraživač
  - ECMA Script
    - JScript
    - JavaScript
    - ActionScript

# Šta je JavaScript?

- Osobine JavaScript jezika:
  - ❑ **Objektno zasnovan** – koristi hijerarhiju ugrađenih objekata sa definisanim metodama i osobinama (što daje jednostavnost jeziku), ali ne podržava sve koncepte objektno-orientisanih jezika (na pr. veoma je limitiran rad sa nasleđivanjem, važenjem i funkcionalnošću objekata)
  - ❑ **Platformski neutralan** – kod se izvršava u okviru web čitača, bez obzira na hardversko i softversko okruženje; veličina programa je obično mala
  - ❑ **Modularno programiranje** – za čuvanje i izvršavanje koda se koriste posebni dokumenti sa **.js** ekstenzijom
  - ❑ **Integrisanost sa HTML-om** – u okviru jedne stranice moguće je na proizvoljan način kombinovati JS i HTML kod; takođe iz JS moguće je generisati HTML kod

# HTML kao ograničavajući faktor

- HTML je platformski neutralan jezik koji omogućava efikasno prezentovanje željenog sadržaja na klijentskoj mašini.
- Glavni nedostatak HTML-a je u tome što ne podržava neposrednu komunikaciju sa korisnikom u smislu mogućnosti unosa podataka od strane korisnika i dinamičku obradu unetih podataka.
- HTML predstavlja **ograničavajući faktor** u aplikacijama koje zahtevaju dinamičku obradu unetih podataka, pa se javila potreba za uvođenjem novih tehnologija za realizaciju ovakvih aplikacija.

# CGI-Common Gateway Interface

- Prvi pokušaji rešavanja problema realizacije dinamičkih aplikacija bili su usmereni ka uvođenju novih serverskih komponenti. Najpopularnija među njima je bila CGI-Common Gateway Interface.
- **CGI tehnologija** je omogućila Web programerima obradu podataka koje je korisnik uneo i, u skladu sa njima, preuzimanje potrebnih akcija.
- Dobijena funkcionalnost je u početku bila dovoljna da se CGI tehnologija smatra zadovoljavajućim rešenjem. Međutim, kasnije se uvidelo da veliki problem može da predstavlja česta klijent-server komunikacija.
- **Primer:** Ako korisnik unese pogrešan podatak, najpre mora da sačeka da se podatak prenese na server, zatim da server obradi podatak i utvrdi da je pogrešan, nakon toga da dobije obaveštenje od servera da ponovo treba da unese podatak.
- Da bi se izbegla česta komunikacija sa serverom, javila se potreba za novim tehnologijama koje bi dale mogućnost **obrade podataka na klijentskoj strani**. To je dovelo do pojave **JavaScript-a**.

# JavaScript na html strani

- **JavaScript** je jezik koji omogućava postizanje značajne funkcionalnosti na klijentskoj strani.
- **Primer:** Ako korisnik unese pogrešan podatak, JavaScript na klijentskoj strani odmah proverava podatak, ustanavljava grešku i o tome obaveštava korisnika, koji samo može biti iznenaden brzinom odziva.

The screenshot shows a web form with several input fields and validation messages. The fields include:

- Country of Residence: United Kingdom (dropdown menu)
- Date of Birth: Day, Month, Year (dropdown menus)
- Title: Mr (dropdown menu)
- Name: Peter Jackson (text input)
- E-mail Address: my@mail.com (text input)
- E-mail Address: your@mail.com (text input)
- Password: Enter Password (text input)
- Re-enter Password (text input)

Validation messages are displayed in red circles:

- E-mail Address: my@mail.com (highlighted with a red oval)
- E-mail addresses must match. (highlighted with a red oval)

# Istorija JavaScript-a

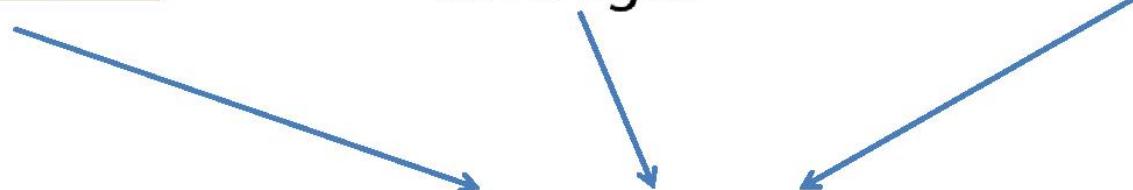
- Decembar 1995.godine – *Netscape* i *Sun* su predstavili jezik *JavaScript* 1.0, originalno nazvan *LiveScript*. Kod napisan u ovom jeziku je mogao da se izvršava u okviru web browser-a *Netscape Navigator* 2.
- Microsoft je na osnovu javne dokumentacije *Netscape-a* implementirao svoju, skoro identičnu, verziju jezika koju je nazvao *JScript*. Ova verzija je mogla da se izvršava u okviru *Internet Explorer* 3 web čitača.
- Udruženje ECMA (*European Computer Manufacturers Association*) je usvojilo svoj standard za ovaj jezik i nazvalo ga *ECMAScript* 1.0, kasnije *ISO/IEC 16262*. Tako je *JavaScript* postao *Netscape-ova* implementacija ovog standarda, a *JScript Microsoft-ova*.

# JavaScript danas

- JavaScript je tokom svog postojanja pretrpeo značajan evolutivni pomak. Iako i dalje ima istu namenu (uvodenje klijentske dinamike), sada su njegove mogućnosti daleko veće, a upotreba daleko učestalija.
- U ovom trenutku, zapravo je nemoguće zamisliti web aplikaciju bez upotrebe JavaScript-a

# JavaScript danas

- Većina klijentskih tehnologija (flash/silverlight/javafx), transportuju se u kombinaciju HTML 5/JavaScript zbog navodne portabilnosti i interoperabilnosti



*JavaScript* **ITAcademy**

# JavaScript danas

- Osim klijentskih, JavaScript je takođe pronašao svoje mesto i u kontekstu serverskih tehnologija. Pre svega, paketom **Node.js**



<http://nodejs.org/>

*ITAcademy*

# JavaScript danas

JavaScript možemo sresti i u tehnologijama koje nemaju direktne veze sa web-om. Na primer, kao jedan od dva skript jezika Unity endžina za kreiranje grafičkih aplikacija i igara.



<http://unity3d.com/>

ITAcademy

# JavaScript danas

JavaScript takođe i sam poseduje osobine koje mu omogućavaju da u koordinaciji sa HTML 5.



emy

# JavaScript danas

- <http://www.chromeexperiments.com/detail/webgl-terrain/?f=webgl>



# Java i JavaScript

- Iako se često poistovjećuju, Java i JavaScript su dva potpuno različita koncepta. JavaScript je jednostavan skript jezik, dok je Java složen potpuno objektno-orientisan jezik.

## Uporedne karakteristike

| JavaScript   | Java  |
|--|---|
| Objektno baziran. Koristi postojeće objekte, bez klasa i nasleđivanja.     | Objektno-orientisan. Podržava klase i nasleđivanje.                             |
| Ne kompajlira se, već se samo interpretira kod klijenta.                   | Kompajlira se na serveru pre nego što se izvrši kod klijenta.                   |
| Kod je integriran sa HTML-om.  | Aplet je izdvojen od HTML-a.  |
| Dinamička povezanost. Objektne reference se povezuju u <i>run-time-u</i> . | Statička povezanost. Objektne reference moraju postojati u vreme kompajliranja. |
| Tip varijable se ne deklariše.   | Tip varijable mora biti deklarisan.   |
| Zaštićen. Ne može da zapisuje na disk.                                     | Zaštićen. Ne može da zapisuje na disk.  |

# Alati za rukovanje JavaScript-om

- Bilo koji tekst editor
- Bilo koji alat za rukovanje html-om  
(dreamweaver, front page...)
- Konzola u pretrazivacu (Debug bar,  
FireBug...)
- JetBrains WebStorm



# Gde se pojavljuje JavaScript

- U HTML dokumentima
- U vidljivom ili nevidljivom delu strane (head / body)
- U zasebnom dokumentu kao biblioteka
- U zasebnom dokumentu u mreži

# Kako se implementira JavaScript u HTML

- Direktnim unosom koda na stranu
- Dodavanjem reference na eksterni fajl

# Osnove JavaScript jezika

- JavaScript (JS) kod se u HTML stranicu može uključiti na dva načina:
  - direktnim pisanjem koda u okviru HTML stranice**
- Tag za definisanje JS koda je:
  - `<SCRIPT [atribut]> JS kod </SCRIPT>`
- gde je atribut obično **type**, a njegova vrednost **text/javascript**.
- pozivanjem posebnog dokumenta sa JS kodom (.js) u okviru HTML stranice**
- Tag za pozivanje dokumenta je:
  - `<SCRIPT atributi> </SCRIPT>`
- gde se, osim atributa **type**(kao u prethodnom slučaju), navodi i atribut **src** koji definiše putanju i naziv .js dokumenta.

# Implementacija JavaScripta u HTML

## Direktni unos koda u stranu

```
<!DOCTYPE html>
<html>
    <head>
        <script type="text/javascript">
            document.write("ovo je javascript kod");
        </script>
    </head>
    <body>
        Telo strane
    </body>
```

# Implementacija JavaScripta u HTML

Implementacija eksternog javascript dokumenta u stranu

```
<!DOCTYPE html>
<html>
    <head>
        <script src="mojaSkripta.js" type="text/javascript" />
    </head>
    <body>
        Telo strane
    </body>
```

# Implementacija JavaScripta u HTML

Implementacija eksternog udaljenog javascript dokumenta u stranu

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
    <script type="text/javascript"
src="http://ecn.dev.virtualearth.net/mapcontrol/mapcontrol.ashx?v=6.2"></script>
    <script type="text/javascript">
      var map = null;
      function GetMap()
      {
        map = new VEMap('myMap');
        map.LoadMap();
      }
    </script>
  </head>
<body onload="GetMap();">
  <div id='myMap' style="position:relative; width:400px; height:400px;"></div>
</body>
```

# Često korišćene JavaScript komande (funkcije/metode)

<http://overapi.com/javascript/>

# Ispisivanje teksta

Za prikazivanje HTML teksta pomoću JS koda koristi se sledeći metod:

`document.write(Tekst)`

Argument metoda (**Tekst**) predstavlja string koji može biti proizvoljan HTML kod.

Primer

```
<SCRIPT type="text/javascript">
    document.write("<b>Definisanje objekta</b><br><i>Kreiranje</i>")
</SCRIPT>
```

Izgled na stranici

```
Definisanje objekta
Kreiranje
```

Napomena: na kraju naredbe u JS nije obavezno korišćenje simbola ;, osim ukoliko je više naredbi u jednom redu

# Ispisivanje teksta

Pored naredbe (metode) document.write, koja emituje tekst na strani, često koristimo naredbu

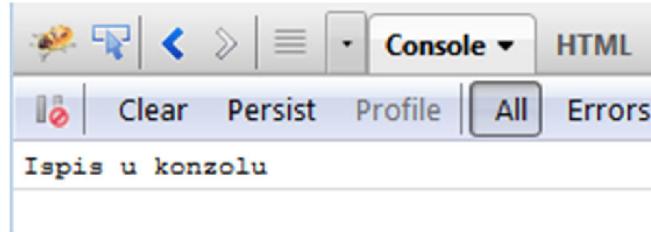
**console.debug(Tekst)** Odnosno **console.log(Tekst)**

Argument metoda (**Tekst**) predstavlja string koji može biti proizvoljan HTML kod.

Primer

```
<SCRIPT type="text/javascript">
    console.debug ("Ispis u konzolu");
</SCRIPT>
```

Izgled u konzoli



# Komentari

- **Komentari** u JS mogu se definisati na dva načina:

// - komentar u jednom redu

Primer: // komentar u jednom redu

/\* \*/ - komentar u više redova

Primer: /\*  
          komentar u više redova  
         \*/

# Promenljive

- **Imena promenljivih - varijabli ili identifikatori** u JavaScript-u zavise od veličine slova (*case sensitive*).
- Pravila za imenovanje promenljivih su:
  - Identifikator mora počinjati **slovom**, znakom \$, ili donjom crtom (\_).
  - U okviru imena se, osim slova, mogu koristiti i brojevi, ali ne na početku imena.
  - Ne mogu se ostavljati prazna mesta u okviru imena.
  - Rezervisane reči se ne mogu koristiti za identifikatore (na pr. *for*, *if*, *function*,...).

Sintaksa za kreiranje (deklaraciju) promenljive i njenu inicijalizaciju je:

**var imePromenljive = vrednost**

# Tipovi podataka (1)

- JavaScript ne podržava deklarisanje tipova podataka. JavaScript interpreter automatski prepoznaće koji tip podataka odgovara kojoj promenljivoj i dodeljuje joj taj tip dinamički. U JS-u je dozvoljeno da promenljiva ima različite tipove podataka u različito vreme izvršavanja programa.

Primer:

```
a = 5;  
b = 8;  
b = "broj" + a;
```

Tokom izvršavanja ovog koda, promenljiva **a** predstavlja broj 5, promenljiva **b** broj 8, a u poslednjoj naredbi potrebno je sabrati vrednost tipa String i broj, što je nemoguće uraditi bez konverzije obe vrednosti u isti tip. Kako tip String u JS ima prednost, broj 5 se konvertuje u String "5", a zatim se izvršava spajanje dva stringa, tako da se na kraju u promenljivoj **b** nalazi String "broj5".

# Tipovi podataka (2)

- Tipovi podataka koji su podržani u JS-u su: **celi brojevi**, **realni brojevi**, **stringovi** i **logičke** vrednosti.
- 1) **Celi brojevi** se mogu predstavljati u pozicionim brojnim sistemima sa osnovom 8 (moraju počinjati cifrom 0), 10 (ne smeju počinjati cifrom 0) i 16 (moraju počinjati sa 0x ili 0X).
  - 2) **Realni brojevi** se mogu predstavljati pomoću decimalne tačke (3.14), ili u eksponencijalnoj prezentaciji (314e+2).
  - 3) **Stringovi** predstavljaju proizvoljan niz karaktera između navodnika (" ") ili apostrofa (' '). U njima se mogu koristiti specijalni karakteri:
    - \b – jedno mesto levo (*backspace*)
    - \f – jedan red na dole (*form feed*)
    - \n – početak novog reda (*new line*)
    - \r – return (*carriage return*)
    - \t – tabulator (*tab*)
  - 4) **Logički tip** podataka dozvoljava vrednosti *true* (tačno) i *false* (netačno). Prilikom konverzije *true* se prevodi u 1, a *false* u 0.

# Operatori

- **Operatori** predstavljaju specijalne karaktere koji definišu operaciju koju treba izvršiti nad operandima. Operandi mogu biti promenljive, izrazi i konstante.
- Operatori se mogu klasifikovati u sledeće grupe:
  - Aritmetički operatori
  - Operatori na nivou bita
  - Logički operatori
  - Operatori poređenja

# Aritmetički operatori (1)

- **Aritmetički operatori** se koriste za izvođenje matematičkih operacija.

## Pregled aritmetičkih operatora

| Operator | Opis       | Operator | Opis              |
|----------|------------|----------|-------------------|
| +        | sabiranje  | + =      | sabiranje dodela  |
| -        | oduzimanje | - =      | oduzimanje dodela |
| *        | množenje   | * =      | množenje dodela   |
| /        | deljenje   | / =      | deljenje dodela   |
| %        | moduo      | % =      | moduo dodela      |
| ++       | inkrement  | --       | dekrement         |

# Aritmetički operatori (2)

- Rad sa operandom tipa **String**:

- **kod sabiranja** – operand koji nije tipa String se konvertuje u String i radi se konkatenacija (sabiranje, tj. spajanje stringova)
- **kod ostalih operacija** – String se konvertuje u broj, a zatim se obavlja operacija; ako konverzija ne uspe, dobija se vrednost NaN (Not a Number/Numeric)

Primer koda

```
a = 1+1;  
b = a*3;  
c = b / 4;  
d = b - a;  
e = -d;  
f = "sto" + 2  
document.write("a,b,c,d,e, su ",a," ",b," ",c," ",d," ",e," ",f);
```

Izgled ekrana nakon izvršenja koda

a,b,c,d,e, su 2 6 1.5 4 -4 sto2

# Operatori na nivou bita (1)

- **Operatori na nivou bita** se koriste za izvođenje operacija nad celim brojevima razmatrajući svaki bit pojedinačno.

## Pregled operatora na nivou bita

| Operator                      | Izraz      | Opis   |
|-------------------------------|------------|--|
| Logičko I (AND)               | $a \& b$   | Rezultat je 1 samo ako su oba bita 1.  |
| Logičko ILI (OR)              | $a   b$    | Rezultat je 0 samo ako su oba bita 0.  |
| Logičko ekskluzivno ILI (XOR) | $a ^ b$    | Rezultat je 1 samo ako su bitovi različiti.                                    |
| Logičko NE (NOT)              | $\sim a$   | Komplementira bitove operanda a.   |
| Pomeranje uлево               | $a \ll b$  | Pomera bite operanda a za b mesta uлево.<br>Prazna mesta popunjava nulama.     |
| Pomeranje удесно са знаком    | $a \gg b$  | Pomera bite operanda a за b mesta удесно.<br>Prazna mesta popunjava MSB bitom. |
| Pomeranje удесно са нулима    | $a \ggg b$ | Pomera bite operanda a за b mesta удесно.<br>Prazna mesta popunjava nulama.    |

# Operatori na nivou bita (2)

## Primer 1

15 & 9; rezultat je 9

(logičko AND)

$$1111 \text{ \& } 1001 = 1001$$

## Primer 2

15 | 9; rezultat je 15

(logičko OR)

$$1111 \text{ | } 1001 = 1111$$

## Primer 3

15 ^ 9; rezultat je 6

(logičko XOR)

$$1111 \text{ \^{} } 1001 = 01110$$

# Logički operatori (1)

- **Logički operatori** se koriste za izvođenje operacija nad operandima logičkog tipa koji mogu imati samo vrednosti **true** (tačno) ili **false** (netačno). Rezultat operacije je takođe logičkog tipa.

## Pregled logičkih operatora

| Operator | Izraz          | Opis   |
|----------|----------------|--|
| I (&&)   | expr1 && expr2 | Rezultat je <i>true</i> samo ako su oba operanda <i>true</i> . U ostalim slučajevima rezultat je <i>false</i> .  |
| ILI (  ) | expr1    expr2 | Rezultat je <i>false</i> samo ako su oba operanda <i>false</i> . U ostalim slučajevima rezultat je <i>true</i> . |
| NE (!)   | !expr          | Rezultat je komplement operanda. Ako je operand <i>true</i> , rezultat je <i>false</i> i obrnuto.                |

# Logički operatori (2)

## Primer koda

```
a = true;  
b = false;  
c = a || b;  
d = a && b;  
f = (!a && b) || (a && !b);  
g = !a;  
  
document.write("a = " + a + "<BR>");  
document.write("b = " + b + "<BR>");  
document.write("c = " + c + "<BR>");  
document.write("d = " + d + "<BR>");  
document.write("f = " + f + "<BR>");  
document.write("g = " + g);
```

## Izgled ekrana nakon izvršenja koda

```
a = true  
b = false  
c = true  
d = false  
f = true  
g = false
```

# Operatori poređenja (1)

- **Operatori poređenja** se koriste za poređenje dve vrednosti i kao rezultat vraćaju vrednost logičkog tipa (*true* ili *false*).

## Pregled operatora poređenja

| Operator                                | Izraz     | Opis   |
|---|-----------|--|
| Jednakost (==)                          | $x == y$  | Rezultat je <i>true</i> ako su $x$ i $y$ jednaki.                          |
| Nejednakost (!=)                        | $x != y$  | Rezultat je <i>true</i> ako su $x$ i $y$ različiti.                        |
| Veće (>)                                | $x > y$   | Rezultat je <i>true</i> ako je $x$ veće od $y$ .                           |
| Veće ili jednako (>=)                   | $x >= y$  | Rezultat je <i>true</i> ako je $x$ veće ili jednako $y$ .                  |
| Manje (<)                               | $x < y$   | Rezultat je <i>true</i> ako je $x$ manje od $y$ .                          |
| Manje ili jednako (<=)                  | $x <= y$  | Rezultat je <i>true</i> ako je $x$ manje ili jednako $y$ .                 |
| Jednako bez konverzije tipova<br>(==)   | $x === y$ | Rezultat je <i>true</i> ako su $x$ i $y$ jednaki bez konverzije podataka   |
| Različito bez konverzije tipova<br>(!=) | $x !== y$ | Rezultat je <i>true</i> ako su $x$ i $y$ različiti bez konverzije podataka |

# Operatori poređenja (2)

- Operatore `==` i `!=` koriste svi dozvoljeni tipovi podataka (celi brojevi, racionalni brojevi, karakteri, stringovi, logički tip). Ukoliko su operandi različitog tipa, obavlja se konverzija podataka pre poređenja.
- Ostale operatore koriste samo numerički tipovi.

Primer koda

```
a = 4;  
b = 1;  
c = a < b;  
d = a == b;  
document.write("c = " + c + "<BR>");  
document.write("d = " + d);
```

Izgled ekrana nakon  
izvršenja koda

```
c = false  
d = false
```

# Funkcije

- U JavaScript-u se primenom rezervisane reči **function** deklariše funkcija sa zadatim parametrima koji mogu biti numeričkog tipa, stringovi ili objekti. Opšti oblik funkcije je:

```
function  
    ime([parametar1],[parametar2]...,[parametarN])  
{  
    telo funkcije  
}
```

# Promenljive

- U JavaScript-u koristi se rezervisana reč **var** za deklarisanje promenljive uz opcionu mogućnost njene inicijalizacije. Opšti oblik deklaracije promenljive je:

```
var imePromenljive[=vrednost]...[,imePromenljive1[=vrednost1] ]
```

# Ugrađene JavaScript funkcije / metode

Core JavaScript – php

**ITAcademy**

# **Ugrađene JavaScript funkcije**

U narednim slajdovima biće predstavljene korisne i često korišećene JavaScript funkcije

# Alert

*`alert('Hello World');`*

*ili*

*`window.alert('Hello World');`*

- Prikazuje message box
- Ne vraća nikakvu vrednost

# Confirm

*confirm("da / ne? ")*

- Uzima u obzir akciju korisnika
- vraća rezultat (true ili false)

*var x = confirm("da / ne?");*

*alert(x);*

## Primer:

- Kreirati JavaScript program koji pita korisnika da li je trenutno dan. Ukoliko korisnik odgovori sa da, program prikazuje poruku, dobar dan. U suprotnom, prikazuje poruku: dobro veče.

## Rešenje

```
alert(confirm("Da li je  
dan?")?"Dobar dan":"Dobro  
vece");
```

# Prompt

*prompt("Unesi vrednost")*

- Vraća unetu vrednost kao rezultat

```
var x = prompt("Unesi vrednost");  
alert("Uneli ste " + x);
```

## Primer:

- Kreirati aplikaciju koja traži od korisnika da unese operand a i operand b, i nad njima izvršava operaciju sabiranja

# Rešenje

- Rešenje 1:

```
var a = prompt("a?");  
var b = prompt("b?");  
var a1 = parseInt(a);  
var b1 = parseInt(b);  
var rez = a1 + b1;  
alert(rez);
```

- Rešenje 2:

```
alert(parseInt(prompt("a?")) +  
      parseInt(prompt("b?")));
```

# Date objekat (osnovna upotreba)

Kreiranje datuma:

```
var dt = new Date();
```

Prikaz datuma:

```
alert(dt);
```

```
alert(dt.getDate());
```

# Date objekat

Kreiranje sopstvenog datuma u inicijalizaciji:

```
var dt = new Date(2010,4,5);
```

ili

```
var dt = new Date("May 05, 2010 12:30:00");
```

Postavljanje serverskog vremena pomoću php-a:

```
echo "<script> var server_time = new Date(".  
(time()*1000) .") </script>";
```

# Date objekat

Eksplisitno podešavanje datuma:

```
var dt = new Date();  
dt.setDate(5);  
dt.setYear(2010);  
dt.setMonth(4);  
alert(dt);
```

# Timeout i Interval

## **setTimeout**

- Izvršava određenu funkciju jednom u određenom vremenskom roku

```
setTimeout("alert('pozdrav')",2000);
```

## **setInterval**

- Izvršava funkciju u određenom vremenskom intervalu

```
setInterval("alert('pozdrav')",2000);
```

## document.location / window.location

- Preuzima ili postavlja vrednost url stringa (i vrši redirekciju)

```
window.location="http://www.google.com";
```

...

```
alert(window.location)
```

..

```
window.location=prompt("unesi adresu");
```

# document.location / window.location

- Preuzimanje url parametara

```
var urlStr = window.location.href;
```

```
var parametri = urlStr.split("?")[1];
```

...

```
var parametri = window.location.search
```

# window.open()

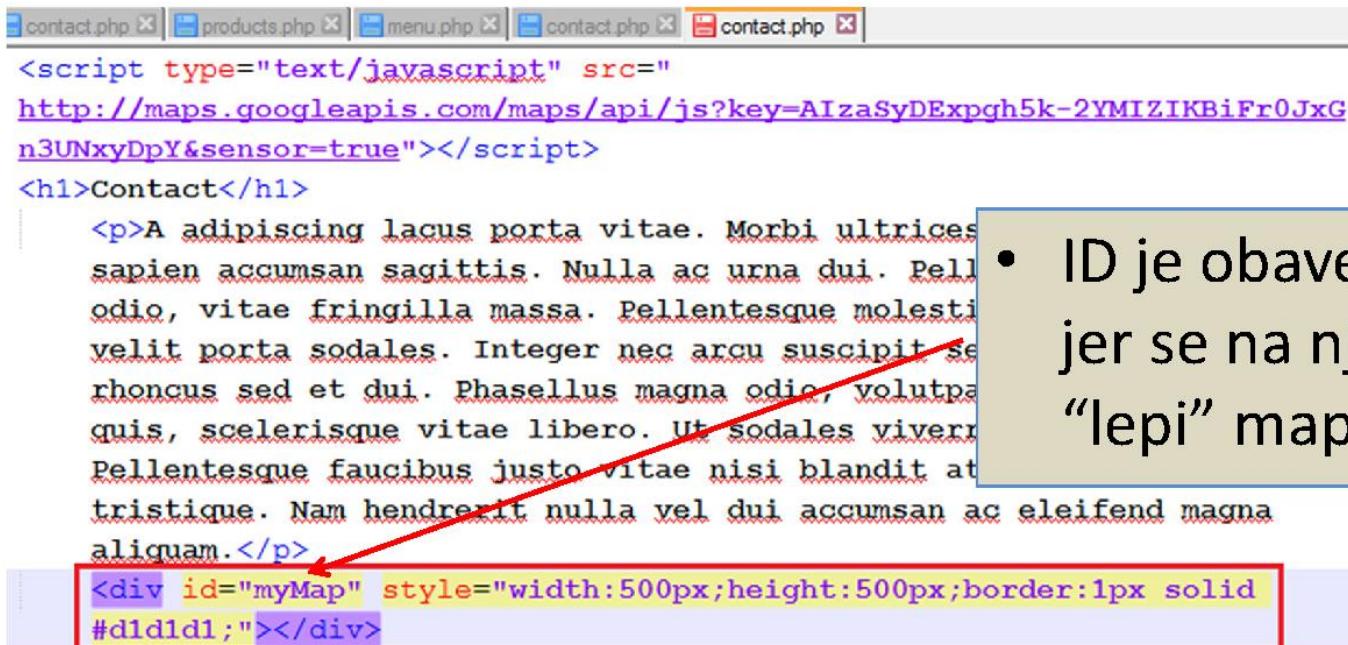
**window.open (url, naslov,argumenti);**

- Status (0/1)
- Toolbar (0/1)
- Location (0/1)
- menubar
- directories
- Resizable (0/1)
- Scrollbars
- height
- width

***window.open ("http://localhost","moj prozor","Status=0");***

# Postavljanje mape na stranu

- Kreirati kontejner koji će sadržati mapu
- Veličina i pozicija kontejnera mogu biti proizvoljne ali on mora imati atribut id



The screenshot shows a browser window with multiple tabs open, all titled "contact.php". The main content area displays a contact form with a heading "Contact" and a paragraph of placeholder text. Below the paragraph is a `<div>` element with the `id="myMap"` attribute, which is highlighted with a red box. A red arrow points from the text "ID je obavezan jer se na njega "lepi" mapa" to this highlighted code.

```
<script type="text/javascript" src="http://maps.googleapis.com/maps/api/js?key=AIzaSyDExpgh5k-2YMIZIKBiFr0JxGn3UNxyDpY&sensor=true"></script>
<h1>Contact</h1>
<p>A adipiscing lacus porta vitae. Morbi ultrices sapien accumsan sagittis. Nulla ac urna dui. Pellentesque odio, vitae fringilla massa. Pellentesque molestie velit porta sodales. Integer nec arcu suscipit semper rhoncus sed et dui. Phasellus magna odio, volutpat quis, scelerisque vitae libero. Ut sodales viverra. Pellentesque faucibus justo vitae nisi blandit at tristique. Nam hendrerit nulla vel dui accumsan ac eleifend magna aliquam.</p>
<div id="myMap" style="width:500px;height:500px;border:1px solid #d1d1d1;"></div>
```

- ID je obavezan jer se na njega "lepi" mapa

# Jednostavni primeri upotrebe gotovog JavaScript-a

- Implementacija Google mapa
- Implementacija JQuery biblioteke
- Implementacija HighSlide-a

# Implementacija Google mapa

```
<!DOCTYPE html>
<html>
  <head>
    <script type="text/javascript"
      src="http://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&sensor=true">
    </script>
  </head>
  <body >
    <div id="map" style="width:400px; height:400px;"></div>
  </body>
</html>
<script>
var myOptions = {
  center: new google.maps.LatLng(-34.397, 150.644),
  zoom: 8,
  mapTypeid: google.maps.MapTypeId.ROADMAP
};
var maptag = document.getElementById("map");
var map = new google.maps.Map(maptag, myOptions);
</script>
```

# Implementacija JQuery biblioteke

[www.jquery.com](http://www.jquery.com)

The screenshot shows the official jQuery website. At the top, there's a navigation bar with links for 'jQuery', 'Plugins', 'UI', 'Meetups', 'Forum', 'Blog', 'About', and 'Donate'. Below the navigation is the jQuery logo with the tagline 'write less, do more.' To the right of the logo is a horizontal menu with 'Download', 'Documentation', 'Tutorials', 'Bug Tracker', and 'Discussion'.

**jQuery is a new kind of JavaScript Library.**

jQuery is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development. **jQuery is designed to change the way that you write JavaScript.**

✓ [Lightweight Footprint](#)   ✓ [CSS3 Compliant](#)   ✓ [Cross-browser](#)

**GRAB THE LATEST VERSION!**

CHOOSE YOUR COMPRESSION LEVEL:

PRODUCTION (31KB, Minified and Gzipped)  
 DEVELOPMENT (229KB, Uncompressed Code)

**Download( *jQuery* );**

*Current Release: v1.7.1*

# Implementacija JQuery biblioteke

```
<!DOCTYPE html>
<html>
<head>
  <script type="text/javascript" src="jquery-1.7.1.min.js">
  </script>
</head>
<body >
  <div id="md" style="width:100px; height:100px; background-color: black;
    position: absolute;"></div>
</body>
</html>
<script>
  $("#md").animate({left:'300'},1000,'swing');
</script>
```

# Implementacija HighSlide-a

<http://highslide.com/>



# Highslide JS

JavaScript thumbnail and media viewer

Search the Highslide JS  
Google™ Custom Search

[Examples](#)  
[License](#)  
[Implementations](#)  
[Installation](#)  
[Compatibility](#)  
**[Download](#)**  
[Editor](#)  
[Configurator](#)  
[Barebones samples](#)  
[Changelog](#)  
  
[Support](#)  
[Tutorial](#)  
[Reference](#)  
[About the author](#)  
[Contact](#)

**What is Highslide JS?**

Highslide JS is an image, media and gallery viewer written in JavaScript. These are some of its advantages:

- Quick and elegant looking.
- No plugins like Flash or Java required.
- Popup blockers are no problem. The content opens within the active browser window.
- Single click. After opening the image or HTML popup, the user can scroll further down or leave the page without closing it.
- Lots of configuration options and scalability without compromising on simplicity. A component system lets you strip away unused features down to a filesize of 10kB.
- Outstanding, unconditionally compatible and safe due to browser redirects directly.
- Source code included.

**Tip!**  
Use the visual [Highslide Editor](#) to set up your Highslide installation without writing code.

**GET THE ZIP PACKAGE**

If you want to install Highslide manually or explore the examples offline you should get the zip package. Unzip it and run the index.htm file to see an overview of the examples.

Version:

SHOWCASE  
See how developers implemented Highslide on their own sites in [this thread](#).

TRANSLATIONS  
Get or share translated GUI strings in [this thread](#).

GET HIGHSLIDE  
[Download!](#)

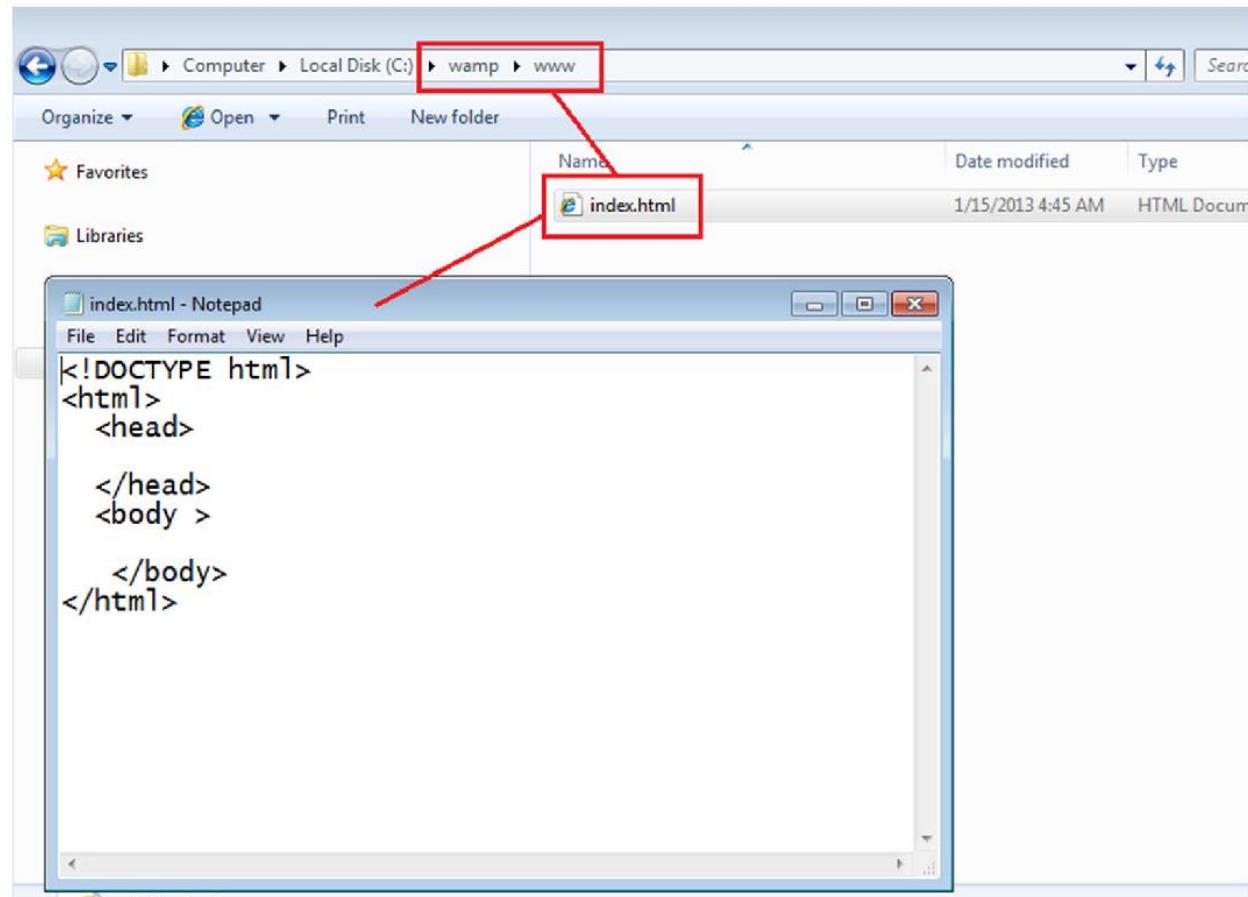
Highslide JS core

```
<!DOCTYPE html>
<html>
  <head>
    <script type="text/javascript" src="highslide/highslide.js"></script>
    <link rel="stylesheet" type="text/css" href="highslide/highslide.css" />
  </head>
  <body>
    <a href="images/full-image.jpg" class="highslide" onclick="return hs.expand(this)">
      
    </a>
  </body>
</html>
```



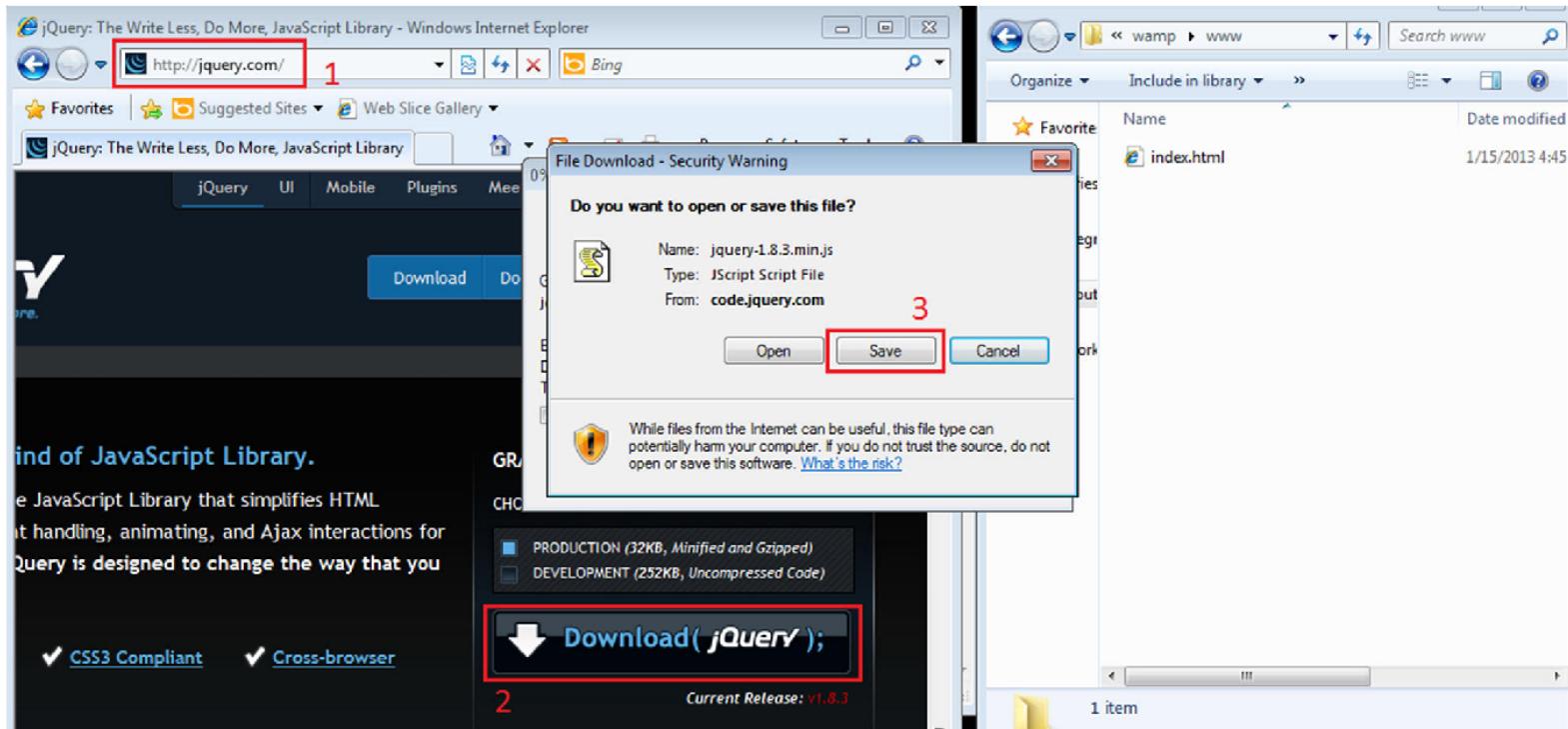
# Primer – implementacije Nivo Slider slideshow-a

- Pripremiti dokument koji će sadržati slide show:



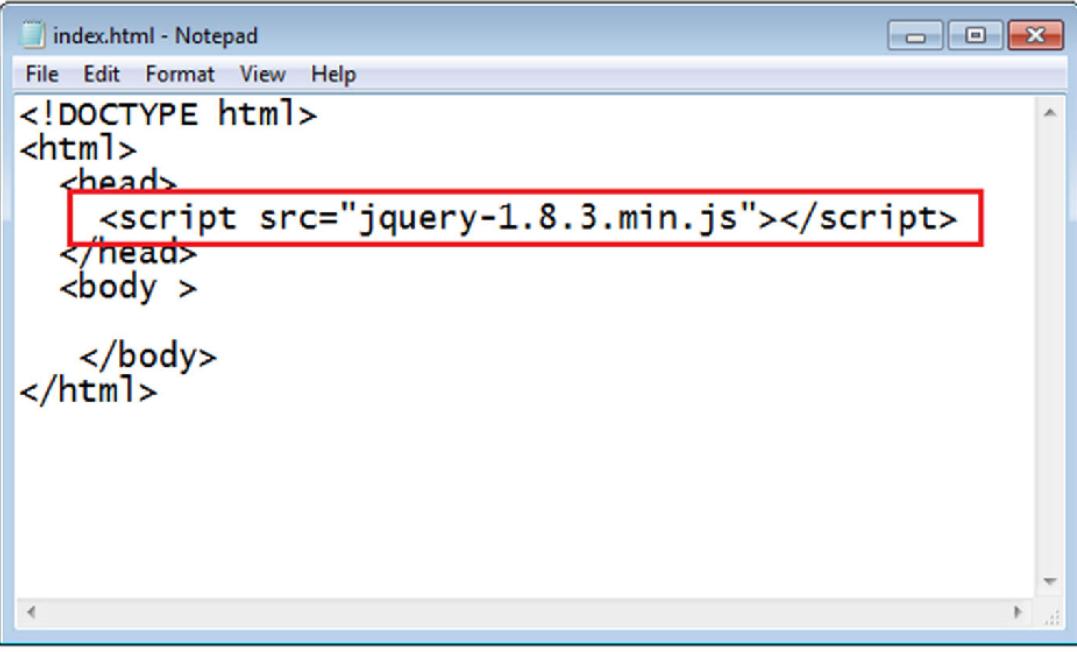
# Primer – implementacije Nivo Slider slideshow-a

- Preuzeti i implementirati jQuery biblioteku



## Primer – implementacije Nivo Slider slideshow-a

- Unutar dokumenta izvršiti implementaciju jQuery biblioteke



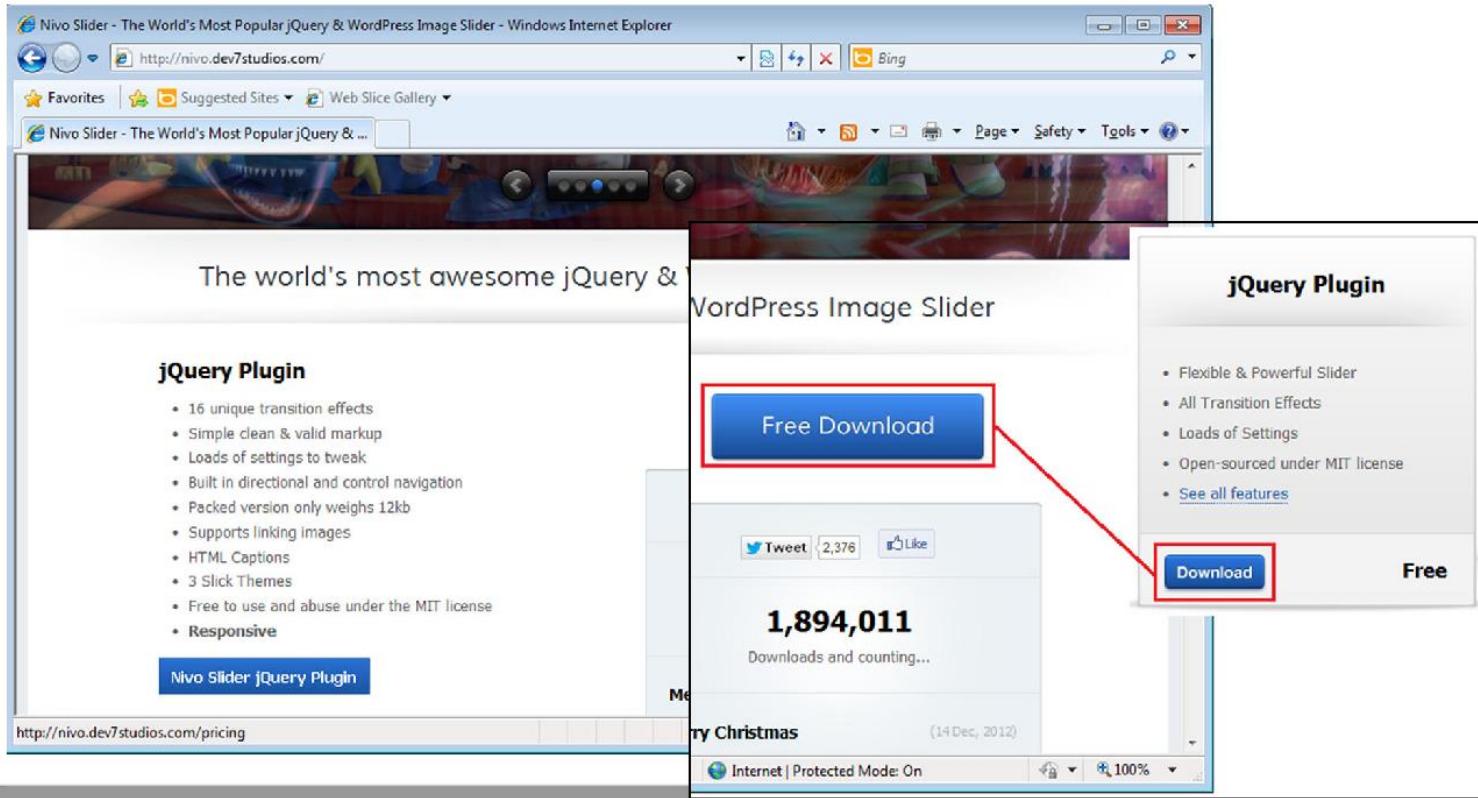
The image shows a screenshot of a Windows Notepad window titled "index.html - Notepad". The window contains the following HTML code:

```
<!DOCTYPE html>
<html>
  <head>
    <script src="jquery-1.8.3.min.js"></script>
  </head>
  <body >
    </body>
  </html>
```

The line containing the `<script>` tag is highlighted with a red rectangular box.

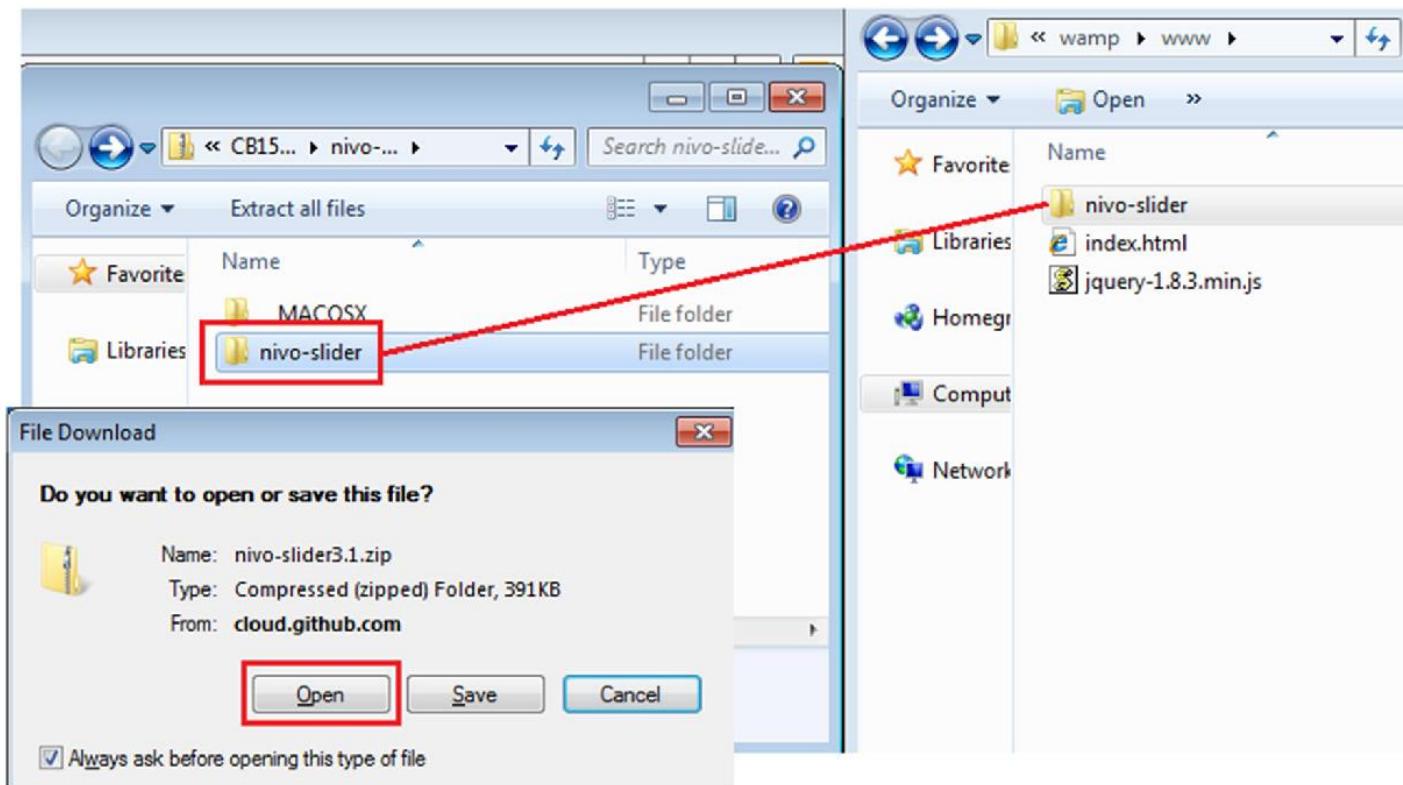
# Primer – implementacije Nivo Slider slideshow-a

- Pristupiti nivo slider sajtu na sledećoj adresi:  
<http://nivo.dev7studios.com/>
- Izvršiti download nivo slider jQuery plug in-a



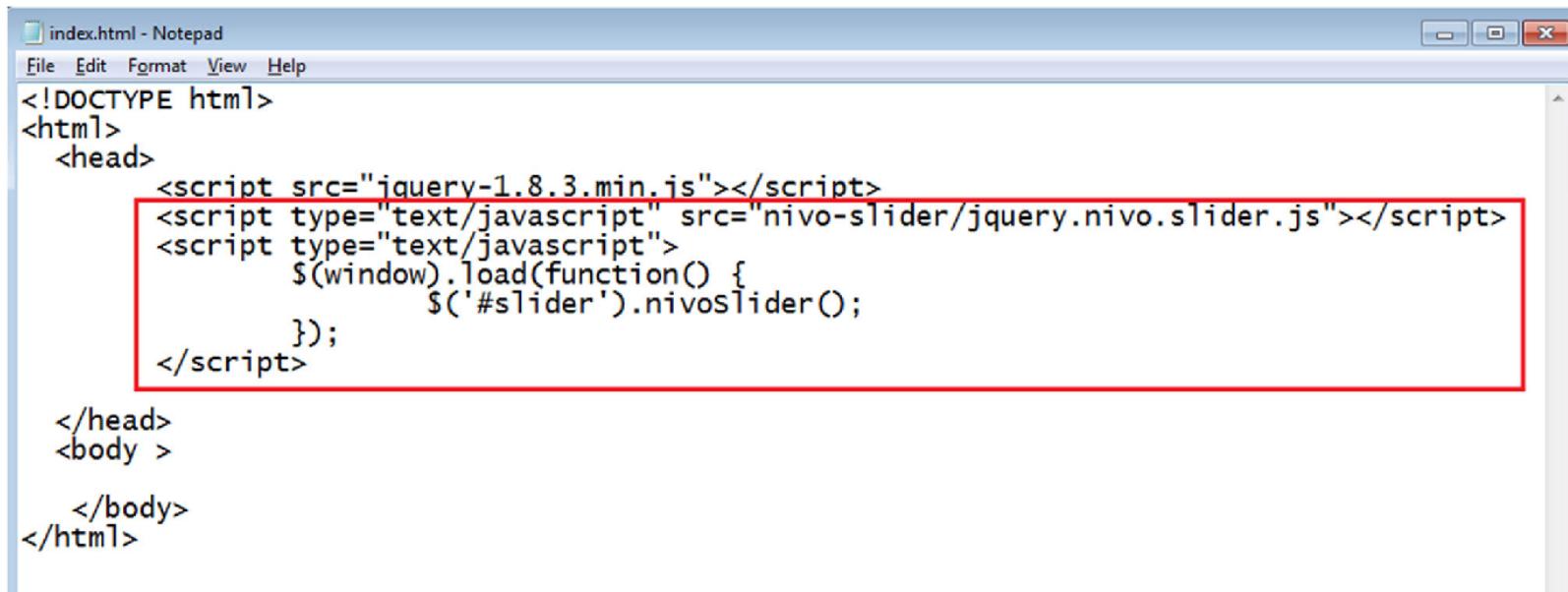
# Primer – implementacije Nivo Slider slideshow-a

- Sadržaj arhive kopirati u folder web aplikacije:



# Primer – implementacije Nivo Slider slideshow-a

- U kod html strane dodati nivo slider endžin i aktivirati nivoSlider metod nad ciljnim elementom:

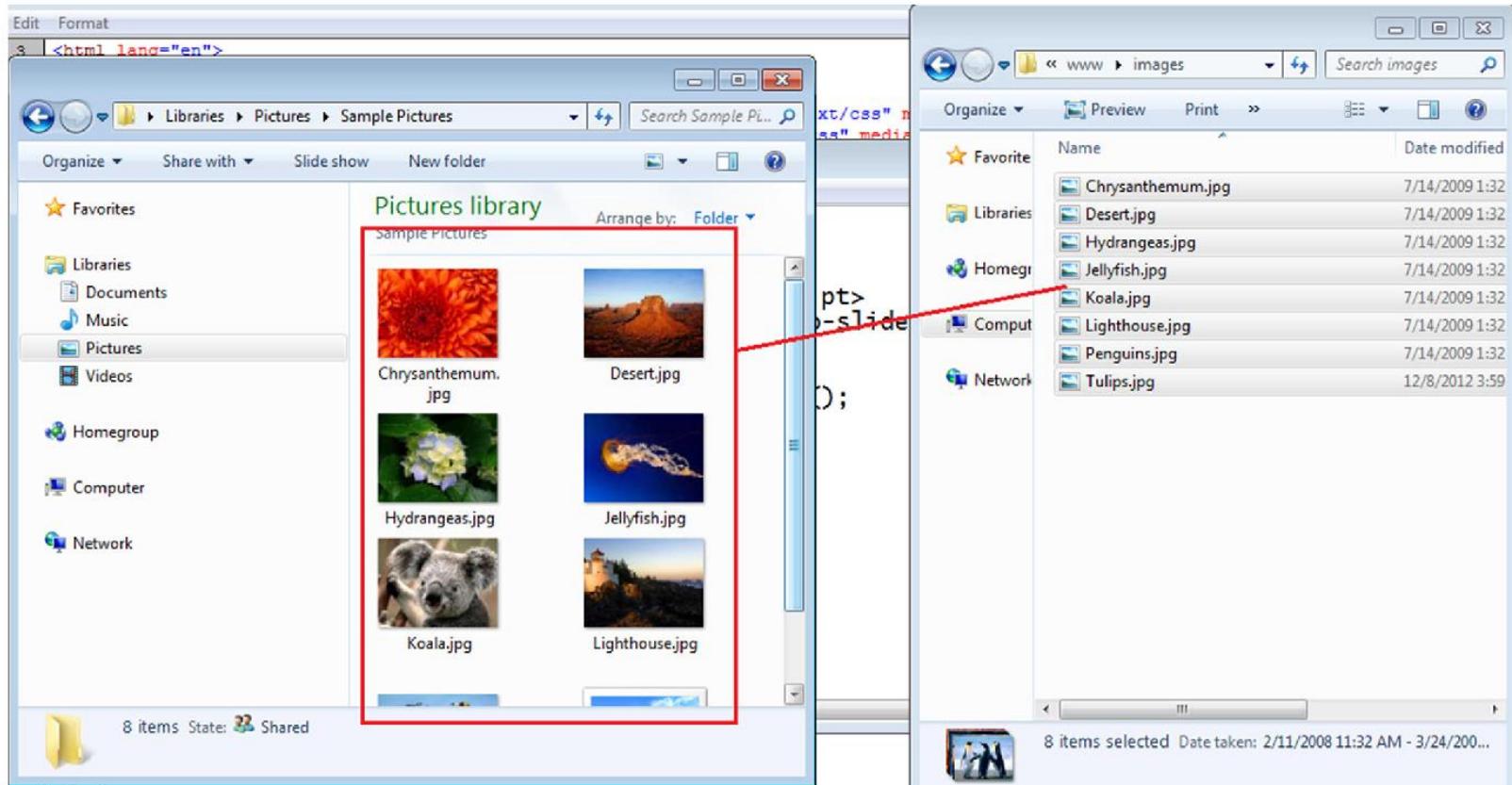


```
index.html - Notepad
File Edit Format View Help
<!DOCTYPE html>
<html>
    <head>
        <script src="jquery-1.8.3.min.js"></script>
        <script type="text/javascript" src="nivo-slider/jquery.nivo.slider.js"></script>
        <script type="text/javascript">
            $(window).load(function() {
                $('#slider').nivoSlider();
            });
        </script>
    </head>
    <body>
    </body>
</html>
```



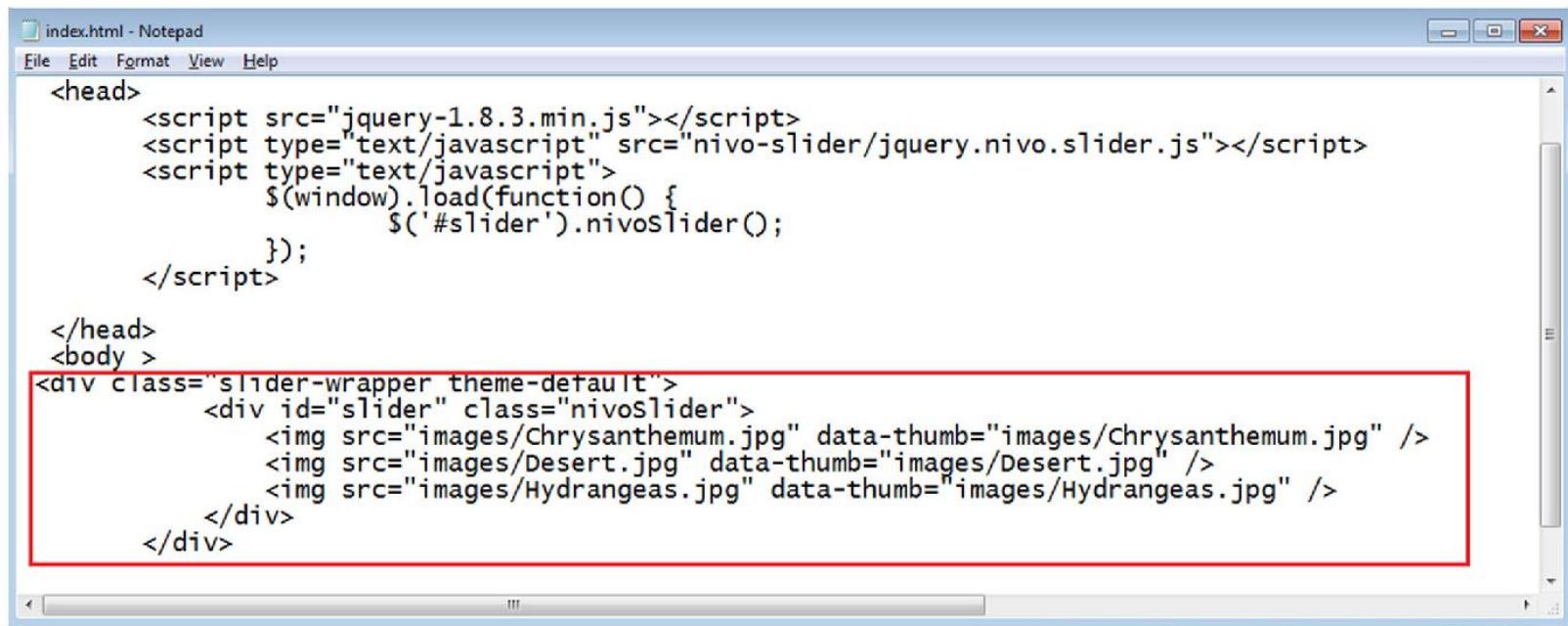
# Primer – implementacije Nivo Slider slideshow-a

- U folderu web aplikacije kreirati podfolder images i u njega smestiti slike po izboru :



## Primer – implementacije Nivo Slider slideshow-a

- Unutar fajla aplikacije, smestiti nivo slider wrapper tag i slike koje će biti korišćene u slajderu:



The screenshot shows a Windows Notepad window titled "index.html - Notepad". The code is as follows:

```
<head>
    <script src="jquery-1.8.3.min.js"></script>
    <script type="text/javascript" src="nivo-slider/jquery.nivo.slider.js"></script>
    <script type="text/javascript">
        $(window).load(function() {
            $('#slider').nivoslider();
        });
    </script>
</head>
<body>
<div class="slider-wrapper theme-default">
    <div id="slider" class="nivoslider">
        
        
        
    </div>
</div>
</body>
```

A red rectangular box highlights the entire content of the `<div class="slider-wrapper theme-default">` tag and its inner `<div id="slider" class="nivoslider">` tag, including the three `<img>` tags.

# Primer – implementacije Nivo Slider slideshow-a

- U glavnom dokumentu aplikacije, navesti putanje do stilova (tema) neophodnih za funkcionalnost nivo slider-a
- Dovoljna je samo jedna tema

The screenshot shows a Windows Notepad window titled "index.html - Notepad". The file contains an HTML document with the following code:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Nivo Slider Example</title>
    <link rel="stylesheet" href="nivo-slider/themes/default/default.css" type="text/css" media="screen" />
    <link rel="stylesheet" href="nivo-slider/themes/light/light.css" type="text/css" media="screen" />
    <link rel="stylesheet" href="nivo-slider/themes/dark/dark.css" type="text/css" media="screen" />
    <link rel="stylesheet" href="nivo-slider/themes/bar/bar.css" type="text/css" media="screen" />
    <link rel="stylesheet" href="nivo-slider/nivo-slider.css" type="text/css" media="screen" />
    <link rel="stylesheet" href="style.css" type="text/css" media="screen" />
</head>
<body>
    <div id="slider"></div>
    <script src="jquery-1.8.3.min.js"></script>
    <script type="text/javascript" src="nivo-slider/jquery.nivo.slider.js"></script>
    <script type="text/javascript">
        $(window).load(function() {
            $('#slider').nivoSlider();
        });
    </script>
</body>
</html>
```

A red rectangular selection highlights the first six `<link>` tags, which define the CSS themes for the Nivo Slider. A blue rectangular selection highlights the last three `<link>` tags, which include the main Nivo Slider CSS and the style.css file.

## Zadatak 1:

- Potrebno je zatražiti od korisnika da unese  
operande i aritmetičku operaciju. Nakon  
unosa, pojavljuje se message box, sa  
porukom “da li ste sigurni”. Ukoliko korisnik  
odgovori potvrđno, izvršava se odabrana  
operacija. Ukoliko odgovori odrično,  
program se prekida.

# Rešenje:

```
var a = parseInt(prompt("a?"));
var b = parseInt(prompt("b?"));
var op = prompt("operator?");
if(confirm("Da li ste sigurni?")) {
    switch(op){
        case "+": alert(a+b); break;
        case "-": alert(a-b); break;
        case "*": alert(a*b); break;
        case "/": alert(a/b); break;
    }
}
```

## Zadatak 2:

- Pitati korisnika da navede adresu na koju želi da bude redirektovan, a nakon toga ga redirektovati na tu adresu

## Rešenje:

```
window.location = prompt("adresa?");
```

## Zadatak 3:

- Korisnik unosi vreme (u minutima)
- Korisnik unosi poruku
- U određenom minuti, aktivira se alert box sa porukom koju je korisnik uneo

## Rešenje:

```
var alarmTime = parseInt(prompt("Minut?"));
var message = prompt("Poruka?");
setInterval("var dt = new Date();
if(dt.getMinutes()=='+alarmTime+' &&
dt.getSeconds()=='0')
alert(''+message+'')",1000);
```