

Osnove programiranja

1. Osnovni elementi programiranja
2. JavaScript
3. Skočni okviri
4. Događaji
5. Varijable
6. Uvjetne strukture
7. Petlje

Program

- - skup naredbi (instrukcija) koje su poredane po strogo određenom redoslijedu i koje nakon izvršavanja daju nekakav rezultat.
- Je li taj rezultat onakav kakav smo očekivali ovisi o tome koliko smo uspješno shvatili, razradili i napisali naš program.
- Vrlo je važno shvatiti da se svaki ozbiljniji program sastoji od tri gotovo potpuno odvojena dijela:
 1. **unos**a podataka za obradu
 2. **obrade** unesenih podataka
 3. **prikaza** rezultata.
- **Promjenom redoslijeda ovih dijelova programa promijenit ćete i njegov rezultat!**
- Jednostavnim rječnikom, to znači da **ne možete** na početku programa "popisati potrebne formule", pa onda "uvrstiti brojeve" kao što to radite npr. u matematici.

Naredba

- Definirali smo program kao skup naredbi. To znači da je naredba osnovni element svakog programa.
- Nizanjem naredbi u određenom redoslijedu dobivamo program. Svaka izostavljena naredba ili naredba postavljena na neko drugo mjesto u programu promijenit će rezultat programa.
-

Programski jezik

- dovoljno je reći da je programski jezik umjetno kreiran jezik pomoću kojeg programer piše program (programira).
- Kao što je osobi koja razumije jedino engleski jezik potrebno nešto objasniti na tom jeziku, tako i s računalom treba komunicirati na jeziku koji ono razumije.
- To nije baš jednostavno, jer znamo da računalo razumije samo jedinice i nule. Srećom postoje ljudi koji su taj problem uspješno riješili razvijanjem tzv. programa prevoditelja.

JavaScript

- Veliki broj primjera, dokumentacije i svega drugog vezanog za JavaScript možete naći na stranicama:
- <http://www.w3schools.com/>
- <http://www.gamelan.com>
- <http://www.tizag.com/javascriptT/>
- <http://homepage.ntlworld.com/kayseycarvey/>

JavaScript

- Objektно orijentiran skriptni jezik specijalno osmišljen za korištenje pomoću web preglednika da bi učinili web stranice što dinamičnijim.
- Umetnuti JavaScript kod u web stranice omogućuje promjenu izgleda web stranice u potpunosti, od promjene boje teksta, promjene boja do padajućih izbornika (i mnogo više).
- JavaScript je klijentski jezik, što znači da se sva akcija događa na klijentovoj (osoba koja pregledava stranicu) strani.
- To znači da za izvršavanje koda nije potrebna nikakva serverska tehnologija, odnosno kod JavaScripta se interpretira direktno od preglednika.
- Ta interakcija preglednika i JavaScripta je razlog zašto se JavaScript koristi prilikom detekcija browsera, validacije formi i sl.
- Ne opterećuje dodatno server pa je time i korisnički doživljaj neometen čekanjem. Sve se odvija u trenu. JavaScript reagira na korisničke događaje kao što je pokret miša, tipkovnice i sl. Što daje još jednu posebnu prednost JavaScripta nad običnim HTML-om.

Kompatibilnost Web preglednika:

- Netscape 2+, Internet Explorer 3+, Opera 3+ i većina drugih modernih web preglednika podržava JavaScript. Svako novo izdanje najpopularnijih preglednika ima podršku za nove generacije JavaScript koda. Problemi se mogu stvoriti jer JavaScript nije standardiziran kao HTML.
- Različiti web preglednici mogu različito interpretirati JavaScript naredbe. Npr. Internet Explorer podržava metode i događaje koji nisu u skladu s web standardima. Samim tim skripte koje sadrže takve nestandardne naredbe se ne mogu izvršiti u web preglednicima koji poštuju standarde.
- **Testiranje** JavaScript aplikacija u što **većem broju web preglednika** je najčešći i najjednostavniji put do izrade aplikacije koja sigurno radi.

Prednosti JavaScripta

- **JavaScript je jednostavan** do te mjere da ga neki autori opisuju kao puko proširenje HTML-a.
- **Moćan i fleksibilan.** Možda je baš zbog te jednostavnosti JavaScript moćan i fleksibilan alat koji svoju primjenu ima već u jednostavnim HTML stranicama. No to je tek početak primjene. JavaScript danas svoju vrijednost dokazuje u kombinaciji s moćnim aplikacijama sa serverske strane ([AJAX](#)) ili u interakciji s brojnim aplikacijama s korisničke strane kao što je primjerice [Flash](#).
- **Popularnost** je jedno od najjačih oružja JavaScripta. Svi suvremeni Internet preglednici ga bez problema razumiju, a velika zajednica korisnika omogućava brz razvoj Internet aplikacija i pomoć ukoliko vam zatreba.
- **Dostupnost je neupitna.** Ukoliko naiđete na stranicu koja ima dopadljivu JavaScript aplikaciju, jedino što trebate napraviti da biste vidjeli kôd aplikacije, jest pogledati source view (**View > Source**). Taj JavaScript kôd je dostupan svima, što je u potpunosti različito od programskih jezika poput PHP-a ili Jave.
- **JavaScript je brz** jer se kao i većina skriptnih jezika čita izravno red po red (i interpretira, a ne kompilira). Uz to, s obzirom da za svoje izvođenje JavaScript ne zahtijeva vezu sa serverom, provođenje JavaScript naredbi započinje odmah.

Mane JavaScripta

- JavaScript vam neće uspostaviti vezu sa serverom, neće vam promijeniti file na vašem računalu i neće se baš dobro snalaziti s grafikom.
- No, to su osobine koje uvjetuju sigurnost, brzinu i jednostavnost JavaScripta. Stoga ih ne treba gledati samo kao mane. Pogotovo što se JavaScript vrlo dobro nadopunjuje s alatima koji su specijalizirani za navedene poslove.

Praktična primjena JS

- Padajući izbornici i druga navigacijska rješenja
- Rolover slike
- Web galerije
- Redirekcije
- Sigurnost
- Search engines
- I sl.

Povijest JavaScripta

- Javascript je rođen 1995. sa ciljem da se programerima koji ne koriste Javu i web dizajnerima pruži jednostavan i moćan skriptni jezik. Prvi internet preglednik koji je podržavao JavaScript bio je Netscape Navigator 2.0
- Prvotni naziv koji je tvorac JavaScripta - [Netscape Communications Corporation](#) - izabrao je bio LiveScript, no uslijed velike popularnosti Jave naziv je promijenjen u JavaScript.

Razlike Jave i JavaScripta

- **Java je objektno orijentirani programski jezik**, izveden iz C i C++ jezika, namijenjen razvoju aplikacija za sisteme na računalnim mrežama.
- Aplikacija napisana u Javi prenosiva je između raznih strojnih platformi, što Javu i čini specifičnom.
- U Javu su ugrađeni brojni sigurnosni mehanizmi:
računalna memorija nije izravno dostupna niti jednom Java programu;
svi bajt kodovi preneseni mrežom provjeravaju se u cilju pronalaženja neželjenih izmjena na putu od servera do klijenta;
onemogućeno je da program-uljez kloniranjem klasa unese destruktivni kod u program itd.
- Pojavom Jave došlo je do znatnog unaprijeđenja komunikacije u Internet svijetu. Osim što su se otvorile mogućnosti interaktivne komunikacije, i web stranice poprimile su dinamičniji oblik.

- **Javascript je kompaktan i objektno baziran skriptni jezik** za razvoj klijent-server Internet aplikacija.
- Programski kod upisuje se direktno na HTML stranicu i omogućava nam izradu server-baziranih aplikacija.
- Javascript podsjeća na Javu i podržava mnoge Javine izraze, sintaksu i konstrukciju kontrole toka. Ima mali broj tipova podataka: numeričke, boolean i string vrijednosti. Podržava i funkcije, ali bez posebnih zahtjeva za deklariranjem.

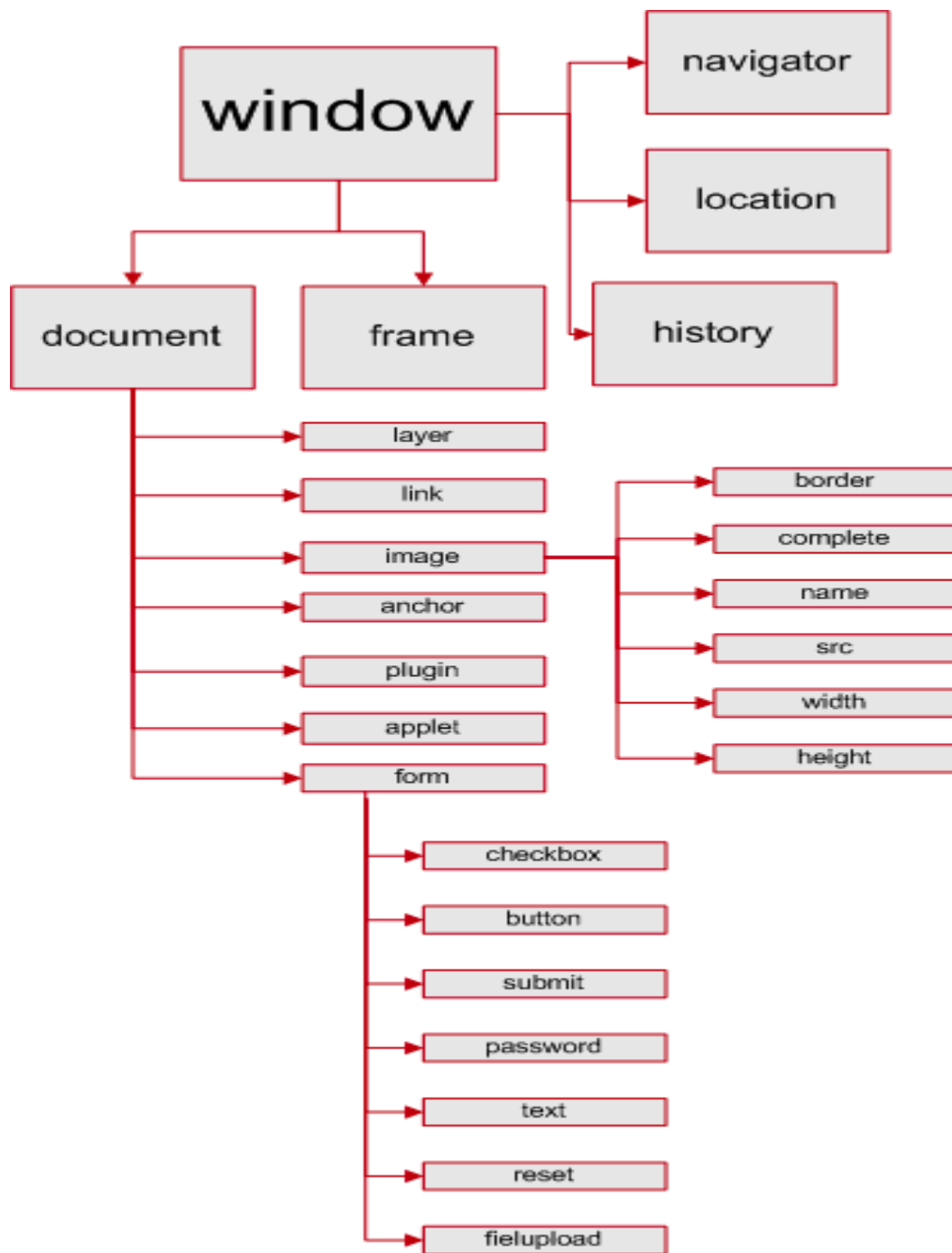
Java	JavaScript
Prevodi se u kompajleru prije izvršavanja	Interpretira ga WWW čitač – ne kompajlira se
OOP jezik – postoji podrška za objekte, klase, nasljeđivanje	Objekto baziran jezik – koriste se ugrađeni objekti
Apleti su odvojeni od HTML dokumenata (iz kojih se pozivaju)	Integriran u HTML dokument
Varijable se moraju deklarirati (jaka tipizacija)	Varijable se ne moraju deklarirati (slaba tipizacija)
Prevodi se u kompajleru prije izvršavanja	Interpretira ga WWW čitač – ne kompajlira se

DOM

Document **O**bject **M**odel je okvir u kojem se izvršava JavaScript. Rekli smo da je JavaScript objektno baziran. Razmislimo koji sve objekti čine jednu web stranicu?

Web stranica sama po sebi je objekt napravljen od drugih objekata kao što su forme, slike, tablice i sl. Forme su objekti napravljeni od još manjih objekata kao što su polja za unos teksta, gumbi...

- Svi navedeni objekti imaju neka svoja svojstva koji sa dodijeljenim vrijednostima definiraju boju objekta, dužinu i sl. JavaScript može čitati ta svojstva i modificirati ih ili može reagirati na događaje koji se izvršavaju nad objektima. Skripta može reagirati u ključnom trenutku ili se samostalno pokrenuti.
- DOM nam omogućuje pristup objektima stranice.



Vrste objekata

- Prema postanku JavaScript objekti se dijele na četiri vrste:
- **Objekte koje je definirao web preglednik.** Primjer za takav objekt je navigator objekt - koji opisuje osobine web preglednika. S obzirom da je taj objekt nastao još dok je Netscape Navigator bio dominantan web preglednik, tvorac JavaScripta - Netscape Communication ga je nazvao navigator.
- **Objekte ugrađene u JavaScript** kao što je Date ili Number.
- **Objekte koje su definirani putem HTML znakova.** To su, kako što se na strukturi DOMa može vidjeti, gotovo svi HTML elementi; počevši od dokumenta, formi, linkova, tekst polja, pa sve prozora ili ćelija tablice.
- **Objekte koje smo sami definirali.** Tvorba novog objekta se vrši putem 'new' operatora.

Implementacija JavaScripta

DOM Document

- JavaScript se piše na isti način kao i HTML — u nekom text-editoru. Mi ćemo koristiti Dreamweaver zbog izvrsnog kode hinting-a i bojanja koda pomoću kojeg je lako uočiti razlike.
- JS implementacija je vrlo slična pisanju CSS – a:
 1. Direktno unutar HTML dokumenta pomoću `<script>` taga (ako skriptu pišemo za jednu stranicu)
 2. Pisanje u eksternu datoteku sa nastavkom .js (ako će se skripta izvršavati na više stranica)

- Primjer 1.
- Prvi primjer ćemo napisati direktno unutar HTML-a
- **<script type="text/javascript">**
- **<!--**
- `document.write("Pozdrav svima!");`
- **//-->**
- **</script>**
-
- Navedeni kod ispisuje *Pozdrav svima!* na ekran.
- **script** tag označava da ćemo u sljedećim linijama pisati skriptu. **type** atribut govori pregledniku o kojoj vrsti skriptnog jezika je riječ (postoje i drugi, npr. VBScript).

Eksterne Skripte

- Da bi ubacili eksterni JS kod moramo kod pisati u posebnu datoteku sa ekstenzijom, nastavkom .js **BEZ script taga**. U tom slučaju primjer1 bi napisali samo kao
document.write("Pozdrav svima!")
spremimo datoteku kao eksterni.js i povežemo sa HTML dokumentom unutar <head> taga
<script type="text/javascript" src="eksterni.js"></script>
- Najbolje je uvijek pisati eksterne JS datoteke.
Prednost eksterne datoteke:
 - može se koristiti na više stranica
 - jednom učitana skripta sprema se u privremeni prostor na disku pa je time ubrzan proces pregledavanja stranica

Još malo o primjeru 1.

- `<script type="text/javascript">`
- `<!--`
- **`document.write("<i>Pozdrav svima!</i>");`**
- `//-->`
- `</script>`
- Počinjemo sa kontroliranjem **objekta document** koristeći **write() metodu** da bi ispisali neki tekst u dokument.
Da bi koristili objektove metode ili svojstva prvo **pišemo njegovo ime**, zatim **točku** i tada ime **metode/svojstva**.

- Primjer2:
- `<script type="text/javascript">`
- `<!--`
- `document.write("<h1>Glavni naslov</h1>");`
- `document.write("<p align='right'>Body</p>");`
- `//-->`
- `</script>`
- Primjetite da smo kod pisanja vrijednosti atributa koristili **jednostruke navodnike**. Ako se zabunom napišu dvostruki dogodit će se greška!

- Zadaci:

1. Ispisati na ekran: Pozdrav svima ja se zovem
Ime i Prezime
2. Ispisati na ekran gornju poruku u crvenoj boji

Napomena

** da bi ispisali u crvenoj boji potrebno je koristiti tag*

DOM Document

- Vidjeli smo da je `write()` jedna od metoda koje se mogu koristiti sa objektom **document**.

Osobina	Opis
<code>lastModified</code>	Datum zadnje izmjene dokumenta.
<code>linkColor</code>	Boja linka.
<code>referrer</code>	Adresa stranice s koje smo došli.
<code>title</code>	Naslov dokumenta.
<code>URL</code>	Vraća URL dokumenta.
<code>vlinkColor</code>	Boja posjećenih linkova.
<code>alinkColor</code>	Boja aktivnih linkova.
<code>bgColor</code>	Boja pozadine.
<code>body</code>	Početak i kraj tijela dokumenta.
<code>domain</code>	Domena na kojoj je smještena stranica.
<code>fgColor</code>	Boja teksta na stranici.

Metode:

Metoda	Opis
<code>clear()</code>	Briše sve elemente dokumenta.
<code>close()</code>	Zatvara dokument.
<code>open()</code>	Otvora novi dokument.
<code>write("abc")</code>	Unosi tekst "abc" u HTML dokument.
<code>writeln("abc")</code>	Unosi tekst "abc" u HTML dokument, s tim da na kraju naredbe prelazi u nov red.

- Primjer upotrebe metode i osobine dokumenta:
- **document.write("Ispiši URL stranice sa koje sam došao: " + document.referrer + "
");**
- document.write - ispisuje na ekran poruku
- + spaja znakovne nizove
- document.referrer – ispisuje adresu na kojoj smo prethodno bili

- **Zadatak:** Napravite skriptu koja će ispisati podatke o vašoj stranici. I to **tekst:** “Traženi podaci su:”, **naslov, URL, boju pozadine, boju teksta, boju linkova, kôd u kojem je stranica pisana i vrijeme** kada je stranica zadnji put izmijenjena.

Napomena:

*

Potrebno je prije izrade zadatka napraviti web stranicu u kojoj je: pozadinska boja žuta, boja teksta crvena, napraviti link na npr. [google.com](https://www.google.com), postaviti naslov stranice na Vježba.

- Koju metodu smo koristili uz **document** objekt?
- Koje podatke možemo provjeriti pomoću osobina objekta **document**?
- Što znači + u JavaScriptu?
- Zadatak: Napravite novu HTML stranicu kojoj je naslov vaše ime i prezime. Pomoću JS i osobine objekta **document** ispišite poruku na ekran: Naslov ove stranice je:

Skočni okviri

DOM Window

Objekt window je roditeljski objekt svih objekata. Prikazuje prozor našeg web preglednika i sve što se unutar njega nalazi. Stoga window objekt omogućava pristup svim objektima koji se nalaze unutar njega.

Osobina	Opis
<code>closed</code>	Vraća boolean vrijednost (da/ne): je li prozor otvoren ili ne.
<code>complete</code>	Vraća boolean vrijednost (da/ne): je li stranica završila učitavanje ili ne.
<code>history</code>	Vraća URL na kojem smo prije bili.
<code>location</code>	Vraća trenutni URL.
<code>name</code>	Postavlja ili vraća ime prozoru.
<code>parent</code>	Vraća roditeljski objekt trenutnog prozora.
<code>status</code>	Postavlja poruku u traci stanja web preglednika.

Metode	Opis
<code>alert()</code>	Pokazuje prozor upozorenja s porukom i OK gumbom.
<code>close()</code>	Zatvara trenutno otvoreni prozor.
<code>confirm()</code>	Pokazuje poruku upozorenja s OK i CANCEL gumbima.
<code>navigate()</code>	Odводи nas na zadani URL.
<code>open()</code>	Otvora nov prozor.
<code>print()</code>	Ispisuje otvoren dokument.
<code>prompt()</code>	Ispisuje prozor upita s porukom i trakom za unos odgovora.
<code>resizeBy()</code>	Mijenja veličinu prozora za x i y veličine zadane u zagradama.
<code>resizeTo()</code>	Mijenja veličinu prozora na dimenzije zadane u zagradama.

- Veliko ograničenje HTML-a je što je dizajner sa HTML-om ograničen samo na jedan prozor za prikaz dokumenata. Ako bi iz čistog HTML-a htio poslati potvrdu korisniku da je neki tekst ispravno uešen npr. u formi (ili slično) to bi zahtijevalo mnogo posla.
- Sa JavaScriptom je to prilično jednostavno izvesti. JavaScript ima mogućnost da stvori male dijaloge za ispis ili unos teksta koji su sadržajem potpuno neovisni od sadržaja HTML dokumenta
- Najjednostavniji način da se ispiše kratka poruka korisniku je korištenje skočnih okvira.

Alert()

- Primjer ispisa teksta pomoću alert metode
alert("Kliknite na OK za nastavak.");
- Primjetite da metoda alert() nema ime objekta ispred sebe. Razlog za to je što pripada objektu **window**, koji je osnovni objekt DOM-a i zato se uvijek podrazumijeva da je riječ o tom objektu ako se ispred metode ne navede naziv objekta.
- Znači, "puno" ime za alert() metodu bi bilo:
window.alert("Kliknite na OK za nastavak.");
- alert() metoda koristi se - da upozori korisnika na nešto.
- Npr., na netočno unesen podatak u formu, pogrešan rezultat kalkulacije i sl...

- Zadatak1: Ispisati: *Pozdrav svima* pomoću alert okvira.
- Zadatak2: Obavijestiti korisnika:
 - a) Pozadinska boja vaše str. je:
 - b) Ime vašeg dokumenta je:

Prompt()

- Alert() metoda nam omogućava da pošaljemo poruku korisniku, ali ne i da preuzmemo odgovor od njega.

Da bi smo omogućili jednostavnu interakciju sa korisnikom, koristimo metodu prompt().

Prompt() stvara prozor u kojem se šalje poruka korisniku i formira se polje u koje korisnik može unijeti proizvoljan tekst.

Primjer:

```
prompt("dobar dan! Kako ste danas?");
```

- Moguće je također predložiti vrijednost u tom polju, koju korisnik može prihvatiti ili ne.

Primjer:

prompt("Unesite Vaše ime:", "Gost");

- Da bi iskoristili informaciju koju je korisnik napisao moramo metodu prompt() povezati sa drugom metodom ili varijablom*.

Primjer:

**ime_korisnika = prompt("Unesite Vaše ime:", "Gost");
document.write(ime_korisnika);**

Napomena:

*bit će kasnije detaljnije objašnjene

- Primjer:

```
<script type="text/javascript">  
    document.write("<h1>Dobrodošli, </h1>");  
    ime_korisnika = prompt("Unesite Vaše  
ime:", "Gost");  
    document.write(ime_korisnika);  
    document.write(" <h1> u svijet JavaScripta!</h1>");  
    document.write("<P>Čestitamo</P>");  
</script>
```

Kad bi željeli i uneseno ime ispisati na ekran u h1 tagu što moramo promijeniti u prethodnom kodu?

- Zadatak: Pitati korisnika da unese svoje ime. Obavijestiti korisnika, njegovim imenom: ***Drago nam je da ste posjetili našu stranicu.*** Nakon što klikne OK, pojavljuje se poruka da unese prezime. Obavijestiti korisnika, njegovim imenom i prezimenom, **Nikad nećete otići sa ove stranice – JavaScript je poludio 😊**

Confirm()

- Confirm metodu koristimo kada korisnik treba potvrditi ili prihvatiti nešto.
- Kada se pojavi confirm okvir, korisnik treba kliknuti ili na “OK” ili na “Cancel” da bi nastavio rad.
- Ako izabere “OK”, confirm okvir vrati vrijednost true(potvrdio). Ako klikne na “Cancel”, vraća false(nije potvrdio).

Npr. confirm(“Želite otići? ”);

JavaScript Event Handlers

Događaji

Što su događaji? (Event Handlers)

- Mali JavaScript kod koji se ne stavlja unutar `<script>` taga već direkno unutar HTML-a. Omogućuje “hvatanje” korisničkih događaja, kao što su:
 - klik na gumb
 - prelazak miša preko linka
 - klikanje POŠALJI gumba u formi
 - Učitavanje stranice u preglednik i sl.

Ovo su samo od neki od događaja koje JavaScript podržava, za potpuniju sliku i detaljniji popis možete pogledati na stranici:
<http://help.dottoro.com/ljtdqwlx.php#mouseProps>

Događaj	Odnosi se na:	Pokrenut je kad:
onClick	Gumb, Dokument, Checkbox, Link, Radio gumb, Obriši i Pošalji gumb	Korisnik klikne na objekt
onDbClick	Document, Link	Korisnik dvostruko klikne na objekt
onMouseDown	Gumb, Document, Link	Korisnik stisne tipku miša
onMouseUp	Gumb, Document, Link	Korisnik pusti tipku miša
onMouseOver	Slika, Link	Korisnik prijeđe kursorom miša preko objekta

Kako se koriste događaji?

- Događaj se najčešće stavlja direktno unutar HTML taga na objekt s kojim želite raditi. Iza imena događaja slijedi JavaScript kod koji želite da se izvrši kad se događaj ostvari.

- Primjer1 - na submit gumbu:
- `<input type="submit" name="kliknime" value="Klikni me!" onclick="alert('Hvala ti!!! ')>`
- Primjer2 - na linku
- `Google`

- Primjer3 - na radio gumbima u formi
`<input type="radio"
ondblclick="document.backgroundColor='#EE
EECC'" />`

Zadaci1:

1. Napravi link koji na prelazak miša otvori alert okvir u kojem je upisano tvoje ime, prezime i razred.
2. Napravi link koji na prelazak mišem mijenja pozadinsku boju na crvenu (#FF0000).
3. Napravi 1. i 2. zadatak sa onclick događajem
4. Napravi 1. i 2. zadatak sa ondblclick događajem
5. Napravi submit gumb koji će na klik promijeniti naslov dokumenta u “Haha jesam te!”

Funkcije

- Funkcija je dio JavaScript koda koji neprimjetno čeka dok se ne pozove. Funkcije su jako dobra stvar kod nekih zadataka koje stalno treba ponavljati.

Funkcija se ne izvršava pri učitavanju stranice zato svaka funkcija mora biti u `<head>` dijelu stranice.

Sintaksa

- Funkciju definiramo sa ključnom riječi ***function*** i dodjeljujemo joj neko ime. Zatim, unutar vitičastih zagrada, pišemo Javascript naredbe koje će ta funkcija izvršiti kada bude pozvana.

function ime()

{

Naredbe koje će se izvršiti

}

Funkciju pozivamo njenim imenom, možemo joj, a i ne moramo dodijeliti neku vrijednost(parametar).

Kako imenovati funkciju?

- Funkcije obavljaju neki posao. Prema tome, u ime funkcije dobro je uključiti glagol koji ukratko i na najbolji način opisuje njeno djelovanje.
- Primjeri dobrih imena:
 - ispisujeDatum
 - obradjujeIme
 - zbrajaBrojeve
 - pokreceRacun
- Ime funkcije obično započinje glagolom iza kojeg slijedi jedna ili više imenica koje počinju velikim slovom. Ovo pravilo imenovanja nije obavezno ali je preporučljivo radi lakše čitljivosti.

Primjer:

```
<html>
  <head>
    <script type="text/javascript">
      function popUp(){
        alert("Ovo je alert okvir koji smo
pokrenuli klikom na link");
      }
    </script>
  </head>
  <body>
    <a href="#link" onclick="popUp()">Otvori
    alert okvir</a>
  </body>
</html>
```

Zadaci2

- 1. Promijeniti zadatke od 1-6 tako da koriste funkcije!
- 2. Napraviti gumb koji će na klik promijeniti pozadinsku boju dokumenta, te obavijesti korisnika o tome.
- 3. Napraviti 2. zadatak koristeći događaje
 - a) **onMouseDown**
 - b) **onMouseUp**
 - c) **onMouseOut**

Zadaci3

napomena: za kreiranje gumba koristite tag `<button></button>`
npr. `<button onclick="klikniMe()">Klikni me</button>`

1. Napravi gumb koji na klik miša obavijesti korisnika koji je naslov dokumenta.
2. Napravi gumb koji na prijelaz miša preko gumba obavijesti korisnika koji je URL dokumenta.
3. Napravi gumb koji na dvostruki klik miša obavijesti korisnika koja je pozadinska boja dokumenta.
4. Napravi gumb koji prilikom odlaska miša obavijesti korisnika koji je naslov

5. Napravi gumb koji kada korisnik pusti tipku miša promjeni naslov dokumenta u “Ovo je 5. zadatak” .
6. Napravi gumb koji kada korisnik prijeđe preko gumba promjeni pozadinsku boju dokumenta u ljubičastu .
7. Napisati u HTML bilo koju rečenicu. Napravi gumb koji na klik mijenja boju teksta na crvenu, a boju pozadine na žutu.
8. Napravi link koji će na dvostruki klik obavijestiti korisnika koji je naslov

Korišćenje dva događaja.

Primjer1:

```
<A HREF="#link"  
  onmouseover="promPozBojuCrvena()"  
  onmouseout="promPozBojuPlava()">  
Prijeđi mišem preko mene 😊</A>
```

Zadaci 4

- 1. Napravi link koji će kad miš dođe na njega promijeniti boju u crvenu i izbaciti aler okvir „Opasna boja“, a kad miš siđe sa linka boja se postavlja na početnu.
- 2. Napraviti gumb koji će na klik miša promjeniti boju pozadine na plavu i postaviti naslov dokumenta na “Plavi dokument”, a na dvostruki klik miša promjeniti boju pozadine na žutu i postaviti naslov dokumenta na “Žuti dokument”.

Otvaranje novog prozora

```
<a href="#link"  
  onclick="OtvoriProzor()">Otvori novi  
  prozor</A>
```

```
<script type="text/javascript">  
  function OtvoriProzor(){  
    open();  
  }  
</script>
```

Argumenti open() metode

- open() naredba prima 4 vrijednosti:
open('URL', 'IME', 'POSTAVKE', 'ZAMJENA');
- **URL** je putanja do stranice koju želimo otvoriti u novom prozoru
- **IME:** specizira ime prozora koji se tek otvorio
- **POSTAVKE:** dopušta postavljenje vrijednosti atributa kao što su: **width, height, scrollbar, location bar, status bar itd**

Lista važnijih postavki

- **height:** visina prozora u pixelima.
- **width:** širina prozora u pixelima
- **location:** provjerava da li postoji location bar
- **menubar:** navigacijska traka
- **scrollbars:** traka za skrolanje
- **status:** statusna traka
- **toolbar:** alatna traka
- **resizable:** omogućuje da novi prozor bude promjenjive veličine. Ako nije definirana ova vrijednost standardno je da prozor onda ne može mijenjati veličinu
- **fullscreen:** da li je prozor otvoren preko cijelog ekrana.

Primjer:

```
<a href="#link"  
  onclick="OtvoriProzor()">Otvori novi  
prozor</A>
```

```
<script type="text/javascript">  
  function OtvoriProzor(){
```

```
    open('dobrodošli.html','dobrodošli')  
    ;
```

```
  }
```

```
</script>
```

Otvora novi prozor i u njemu stranicu

Primjer:

```
<script type="text/javascript">  
  function OtvoriProzor(){  
    open('dobrodosli.html',  
    'dobrodosli', 'width=300,height=200');  
  }  
</script>
```

- Ovaj primjer otvara novi prozor dimenzija 300 pixela širine i 200 pixela visine.
- Pripazite na način pisanja ovih postavki!!
- **width i height** – zatvoreni su u jednostrukim navodnicima bez razmaka i nema navodnika oko brojčanih vrijednosti.

Primjer:

```
<script type="text/javascript">  
  function OtvoriProzor()  
  {  
    open('dobrodosli.html', 'dobrodosli',  
    'width=300,height=200,menubar=yes,  
status=yes,  
location=yes,toolbar=yes,scrollbars=y  
es');  
  }  
</script>
```

Zadaci 5

1. Napraviti link koji će otvoriti novi prozor sa sljedećim postavkama: 400 x 200 pixels, ima scroll bar, ima menu bar.
2. Koja je metoda za otvaranje prozora? Što mislite koja će biti za zatvaranje prozora?
3. Napravi link koji će otvoriti postojeću stranicu u novom prozoru, a na toj stranici je link za zatvaranje

Zadaci 6

1. Napravi gumb koji na dvostruki klik otvara novi prozor i u njemu google.com.
2. Napravi gumb koji na klik miša postavlja status “Odlazite?”
3. Napravi “galeriju” slika (slike na ozana.pgski.hr). Kad korisnik klikne na malu sličicu, otvara mu se prozor sa velikom slikom, prozor je dimenzija slike! Prozor sa velikom slikom zatvara se tako da korisnik klikne bilo gdje na velikoj slici.

- Primjer.- Želimo napraviti 10 gumba koji na klik miša promijene boju pozadine u 10 različitih boja.
- Kako bi to napravili?

Prosljeđivanje vrijednosti funkciji

- Da ne bi morali pisati 10 različitih funkcija, puno koda i velika mogućnost pogreške u pisanju, lako prosljedimo vrijednost funkciji. Tako ćemo prethodni primjer riješiti prosljeđivši vrijednost boje u funkciju `PromjenaBoje()`.
- Otvoriti `10_boja.html`!

Zadaci 7

Napomena: Zadatke riješiti prosljeđivanjem vrijednosti funkciji

- 1. Napraviti 5 gumba koji mijenjaju naslov dokumenta. Zadatak napraviti prosljeđivanjem vrijednosti u funkciju.
- 2. Napraviti 6 poveznica koje na klik otvaraju stranice google.com, facebook.com, twitter.com, net.hr, monitor.hr, novatv.hr u novom prozoru.
- 3. Promjeniti 2. zadatak tako da je

Variable

- Varijable se pojavljuju u svakom programskom jeziku.
- **Variable** možemo opisati kao posude u kojima spremamo informacije s kojima JavaScript barata.
- Možemo ih definirati kao brojčane vrijednosti ili kao tekst(string).
- Primjer iz matematike: $x=5$.
- x je ime varijable kojoj je dodijeljena vrijednost 5.
- Ili $z = x+y$, varijabli z dodijeljen je izraz $x+y$.
- U JavaScriptu nije potrebno deklarirati kojeg je tipa varijabla, dovoljno je samo napisati ime varijable i vrijednost, pri deklariranju varijable često se koristi ključna riječ **var**, no ona nam nije neophodna.

Deklaracija varijabli

- Deklaracija varijabli se u JavaScriptu vrši izrazom:
var imeVariable;
- Iako JavaScript prihvaća više alternativnih načina tvorbe varijable, na ovaj način možete biti sigurni da ste kvalitetno deklarirali varijablu.
- Uz to, koristeći izraz **var** ispred imena varijable nedvojbeno ste naznačili da je to prvi put da se ta varijabla koristi u tom programu, što može uvelike olakšati snalaženje u programu.

var imeVariable;

- Ispred imena varijable nalazi se ključna riječ var, a sama varijabla naziva se **imeVariable**. Ovom linijom koda nismo dodijelili varijabli nakakvu vrijednost.

Pridruživanje vrijednosti varijabli se može napraviti u istom redu kada se varijabla deklarira:

var imeVariable = 3;

ili to može biti odvojeno:

var imeVariable;

imeVariable = 3;

Znak = se koristi za dodjeljivanje vrijednosti.

Znak = ne znači kao u matematici da su te dvije strane jednake.

= u programiranju označava da smo vrijednost sa lijeve strane pripremili da postane jednaka vrijednosti sa desne strane jednakosti. Kraće rečeno: ono što je na lijevoj strani postaje jednako desnoj.

Primjer

- **var mojIme="Ozana";** // znakovni niz (string)
- Znakovni nizovi moraju biti napisani unutar navidnika
var muskarac = false; // Logička vrijednost (Boolean true ili false)
var godine = 30; //Numerička vrijednost
- Broj mora biti napisan bez razmaka.
- Za sve varijable nije potrebno navoditi kojeg je tipa jer JavaScript prepoznaje i samostalno određuje kojeg je tipa varijabla prema vrijednosti koju smo joj dodijelili (string, numerička, logička i sl.)

Pravila za imenovanje varijabli

- Ime varijable mora započinjati slovom.
- Ime varijable može sadržavati brojeve.
- Ime varijable ne smije imati razmake ili bilo kakve druge specijalne znakove
- JavaScript je case sensitive tako da varijabla:
npr. *auto* različita je od varijable *Auto*
- Uobičajeno je:
 - prvo slovo u imenu pisati malim slovom
 - ako ime varijable ima više riječi pisati ih spojeno pri čemu se stavi veliko početno slovo druge riječi kao `introTxt` (camelback stil pisanja)

Lokalne varijable

- Varijabla koja je deklarirana u JavaScript funkciji zove se **LOKALNA**. Pristup toj varijabli ima samo ta funkcija. Lokalne varijable možemo nazivati čak i istim imenima jer se samo koriste u funkciji u kojoj je deklarirana. Lokalne varijable budu „uništene“ kad skripta izađe iz funkcije.

Globalne varijable

- Varijabla koja je deklarirana izvan funkcije naziva se **GLOBALNA**. Pristup toj varijabli imaju sve skripte i sve funkcije web stranice. Globalne varijable budu „uništene“ kad ugasimo web stranicu.

- Što će biti ispisano na ekran?

```
var moj_film = "Godfather";
```

```
document.write("Moj najdraži film je: moj_film");
```

- Što će biti ispisano na ekran?

```
var moj_film = "Godfather";
```

```
document.write("Moj najdraži film je: "+ moj_film);
```

+ operator

- Koristimo ga u dvije različite situacije
- + se koristi za zbrajanje numeričkih vrijednosti
- + se koristi za sastavljanje vrijednosti kod (stringova) nizova teksta

Primjer1:

```
var ime="Petar"  
var prezime="Petrić"  
document.write(ime+" "+prezime);
```

Primjer2:

```
var txt1="Želim vam reći";  
var txt2="dobar dan";  
var txt3=txt1+txt2;  
document.write(txt3);
```

Zadaci

1. Deklarirajte varijable za sljedeće:

Broj godina osobe, Plaća zaposlenika, Broj riječi u riječniku, Slovo abecede.

2. U zadatku1 varijablama redom dodijeliti vrijednosti: 24, 5500, 80000, A.

3. Ispisati na ekran vrijednosti varijabli!

4. Deklarirati varijable: prvaRijec, drugaRijec, trecaRijec, cetvrtaRijec i dodijeliti im redom vrijednosti: “Ova”, “rečenica je”, “ispisana spajanjem”, “varijabli”

5. Ispisati varijable iz zadatka4 u jednoj rečenici.

Aritmetički operatori

- Aritmetički operatori služe za izvođenje operacija nad vrijednostima varijabli. Recimo da nam varijabla y ima vrijednost 5.

Operator	Opis	Primjer	Rezultat
+	Zbrajanje	$x=y+2$	$x=7$
-	Oduzimanje	$x=y-2$	$x=3$
*	Množenje	$x=y*2$	$x=10$
/	Dijeljenje	$x=y/2$	$x=2.5$
%	Modul (ostatak pri dijeljenju)	$x=y\%2$	$x=1$
++	Povećanje (inkrementacija)	$y++$	6
--	Smanjenje (dekrementacija)	$y--$	4

- Kolike su vrijednosti varijabli e, f i g?

```
var a = 4;           var e = a + b + c;  
var b = 6;           var f = b - c + a;  
var c = 10;          var g = b + b / d;  
var d = 3;
```

- Kolike su vrijednosti varijabli a i b?

```
var a = 34 % 16;  
var b = 64 % 8;
```

- Kolike su vrijednosti varijabli a i b?

```
var a = 5;           var b = 10;  
a++;                 b--;
```


Zadaci

1. Deklarirati dvije varijable: a i b , te im dodijeliti vrijednosti 7 i 2. Ispisati na ekran vrijednosti: $a+b$, $a-b$, $b-a$, $a*b$, a/b , b/a , $a\%b$, $a++$, $b++$, $a--$, $b--$.
2. Tražiti od korisnika da unese 2 broja. Ispisati vrijednosti zbroja, razlike i umnoška unesenih brojeva.
3. Što se dogodilo?

- Napomena: Ako „zbrajamo“ znakovni niz(string) i broj rezultat će biti znakovni niz(string).

Primjer:

http://www.w3schools.com/js/tryit.asp?filename=tryjs_variables

Mogućnosti ?!?

```
var numVrijednost = +txtVrijednost;
```

```
var numVrijednost = txtVrijednost - 0;
```

```
var numVrijednost = txtVrijednost * 1;
```

```
var numVrijednost = txtVrijednost / 1;
```

```
var numVrijednost = Number(txtVrijednost);
```

```
var numVrijednost = parseFloat(txtVrijednost,10);
```

```
var numVrijednost = parseInt(txtVrijednost,10);
```

parseInt

- parseInt() funkcija prima vrijednost stringa a vraća brojčanu vrijednost!!
- **parseInt(string, radix);**
- Parametar radix koristimo kada želimo naglasiti o kojem je brojevnom sustavu riječ. Odnosno JavaScript interpretira radix parametar na sljedeći način:
- Ako string počinje sa "0x" - radix je 16 (heksadecimalni)
- Ako string počinje sa "0" - radix is 8 (oktalni).
- Ako string počinje sa bilo kojom drugom vrijednosti – radix je 10 (decimalni) i nije ga potrebno navoditi.

Primjer

`parseInt("10") → 10`

`parseInt("10.33") → 10`

`parseInt("40 years") → 40`

`parseInt("He was 40") → NaN (Not a number)`

`parseInt("10",10) → 10`

`parseInt("010") → 8`

`parseInt("10",8) → 8`

`parseInt("0x10") → 16`

`parseInt("10",16) → 16`

Number

- **var broj = Number(vrijednost);**
- Pozivanje Number() funkcije za pretvorbu tekst polja u broj je najsporiji način od svih prethodno navedenih načina. Zato jer za tu konverziju potrebno je pozivati funkciju da obavi zadatak umjesto da se pretvorba obavi direktno.
- Velika prednost ove metode je zato što je lako uočljivo što JS kod radi i ne postoji mogućnost pogreške u tumačenju. Pa ako kod nije “težak” ovo je najbolja metoda za koristiti u pretvorbi.

Primjer Forma

- Napisati u HTML-u sljedeći kod:

```
<form>
```

```
<input type="text" id="PrviBroj"/>
```

```
<input type="text" id="DrugiBroj"/>
```

```
<br />
```

```
<input type="button" value="Izracunaj" onclick="Izracunaj()"/>
```

```
<br />
```

Rezultat je:

```
<input type="text" id="Rezultat" />
```

```
</form>
```

- Napisati Javascript kod na odgovarajuće mjesto:

```
<script type="text/javascript">
```

```
function Izracunaj(){
```

```
    a=document.getElementById("PrviBroj").value;
```

```
    b=document.getElementById("DrugiBroj").value;
```

```
    a=Number(a);
```

```
    b=Number(b);
```

```
    zbroj=a+b;
```

```
    document.getElementById("Rezultat").value=zbroj;
```

```
}
```

```
</script>
```


Zadaci

1. Napravi skriptu koja traži od korisnika da unese 3 broja: a, b i n. Ispiši na ekran rezultat formule:
$$n * ((a+b) \% 2)$$
2. Napravi formu sa dva tekstualna polja. Ispiši u treće tekstualno polje umnožak ta dva broja, a u četvrto tekstualno polje ispiši modulo ta dva broja.
3. Napravi formu sa dva tekstualna polja. U prvo tekstualno polje korisnik upisuje svoje ime, a u drugo svoje prezime. Skripta ispisuje na ekran “Dobrodošli Ime Prezime!”

4. Napiši skriptu koja traži od korisnika da unese 2 broja putem forme. Skripta računa a^2+b^2 i ispisuje rezultat na ekran.
5. Promijeni 4. zadatak tako da rezultat bude ispisan u polje forme.
6. Napraviti tekstualno polje i gumb. U tekstualno polje korisnik unosi web adresu, a klik na gumb otvara tu adresu u novom prozoru.

5. Napraviti tekstualno polje i gumb. U tekstualno polje korisnik unosi vrijednost boje, a klik na gumb postavlja tu boju kao pozadinsku boju stranice.
6. Napraviti 5. zadatak tako da korisnik može odabrati želi li promijeniti boju pozadine ili boju teksta.
7. Napraviti dva tekstualna polja u koja korisnik unosi željenu širinu i visinu ekrana, a na klik gumba veličina prozora se promijeni na unesene vrijednosti.

Napomena: za 7. otvoriti DOM.ppt. Zadatak testirati u IE!

Operatori dodjele

Recimo da je **x=10** i **y=5**

Operator	Primjer	Rezultat	Kraće pišemo:
=	x=y	x=5	
+=	x=x+y	x=15	x+=y
-=	x=x-y	x=5	x-=y
*=	x=x*y	x=50	x*=y
/=	x=x/y	x=2	x/=y
%=	x=x%y	x=0	x%=y

JavaScript operatori

Aritmetički operatori

- Aritmetički operatori služe za izvođenje operacija nad vrijednostima varijabli. Recimo da nam varijabla y ima vrijednost 5.

Operator	Opis	Primjer	Rezultat
+	Zbrajanje	$x=y+2$	$x=7$
-	Oduzimanje	$x=y-2$	$x=3$
*	Množenje	$x=y*2$	$x=10$
/	Dijeljenje	$x=y/2$	$x=2.5$
%	Modul (ostatak pri dijeljenju)	$x=y\%2$	$x=1$
++	Povećanje (inkrementacija)	$y++$	6
--	Smanjenje (dekrementacija)	$y--$	4

Operatori dodjele

Recimo da je **x=10** i **y=5**

Operator	Primjer	Rezultat	Kraće pišemo:
=	x=y	x=5	
+=	x=x+y	x=15	x+=y
-=	x=x-y	x=5	x-=y
*=	x=x*y	x=50	x*=y
/=	x=x/y	x=2	x/=y
%=	x=x%y	x=0	x%=y

Usporedni operatori:

Recimo da je **x=5**

Operator	Opis	Primjer
==	Je jednako	x==8 netočno x==5 točno
===	Je točno jednako	x===5 točno x=== "5" netočno
!=	Nije jednako	x!=8 točno
>	Je veće od	x>8 netočno
<	Je manje od	x<8 točno
>=	Je veće ili jednako od	x>=8 netočno
<=	Je manje ili jednako od	x<=8 točno

Logički operatori

Recimo da je **x=6** i **y=3**

Operator	Opis	Primjer
&&	i	(x < 10 && y > 1) točno
 	ili	(x==2 y==5) netočno
!	ne	!(x==y) točno

Uvjetni operatori

- JavaScript ima operatore koji dodijeljuju vrijednost varijabli na temelju nekog uvjeta
- **Sintaksa**
- `imeVarijable=(uvijet)?vrijednost1:vrijednost2`
- **Primjer**
- `pozPoruka= (titla=="prof") ? "Dragi profesore " : "Dragi učenice";`
- Ako varijabla **posjetitelj** ima vrijednost „prof“ tada će vrijednost varijable **pozdravnaPoruka** biti Dragi profesore inače će biti Dragi učenice.

IF grananje

- IF grananje je jedno od najčešće korištenih naredbi u JavaScriptu, a i mnogim drugim programskim jezicima.
- If grananje je jednostavan logički sklop za donošenje odluka.
- Sintaksa:

```
if (uvjet){  
  napravi to i to  
}
```

- **if** kao ključna riječ ovog grananja web pregledniku govori:
- ukoliko je ono što je između zagrada (**istinito**) -> izvrši naredbe unutar vitičastih zagrada {}.
- Ako nije istina -> web preglednik će preskočiti taj dio i nastaviti provođenje koda koji slijedi nakon grananja.

Primjer

- Napisati skriptu koja traži od korisnika da unesu dva broja, a ukoliko su ti brojevi jednaki obavijesti korisnika “Unešeni brojevi su jednaki”.

```
var a=prompt("Unesi prvi broj: ");  
var b= prompt("Unesi drugi broj: ");  
if (a==b){  
    alert("Unešeni brojevi su jednaki!");  
}
```

if... else

- if... else naredba je proširenje if naredbe kojim se definira što će zbiti ukoliko je tvrdnja u zagradama neistinita.
- Sintaksa

```
if (uvjet) {  
    napravi to i to;  
}  
else{  
    napravi ono;  
}
```

Primjer

- Napisati skriptu koja traži od korisnika da unesu dva broja, skripta provjerava da li su ti brojevi jednaki, a zatim obavijesti korisnika o tome.

```
var a=prompt("Unesi prvi broj: ");  
var b= prompt("Unesi drugi broj: ");  
if (a==b){  
    alert("Unešeni brojevi su jednaki!");  
}  
else{  
    alert("Unešeni brojevi su različiti!");  
}
```


Primjer1.

```
var odgovor = confirm("Razumiješ li pojam varijable?");  
if (odgovor) {  
    alert("Bravo! Možeš učiti dalje ");  
}  
else{  
    alert("Moraš ponoviti gradivo");  
}
```

Primjer 2.

```
var n = prompt("Unesi broj: ");  
if ((n%2)==0){  
    alert("Broj je paran!");  
}  
else {  
    alert("Broj je neparan!");  
}
```

- Primjer3. Što će biti ispisano u alert okviru?

```
var a = 40;  
var b = 60;  
if (!(a > b)) {  
    alert("Zdravo!");  
}  
else {  
    alert("Doviđenja");  
}
```

- Primjer4.

```
var pime = navigator.appName;  
if (pime == "Microsoft Internet Explorer") {  
    window.location="explorer_verzija.html";  
}  
else {  
    window.location="netscape_verzija.html";  
}
```

if... else if..else

- **if... else if... else** naredba je kombinacija if i else naredbe. Ona zapravo - u slučaju da prvi uvjet nije istinit, provjerava drugi uvjet na potpuno jednak način.

```
if (uvjet1) {  
    napravi to i to;  
}  
else if (uvjet2){  
    Napravi to i to - ako nije prvi  
    uvjet zadovoljen  
}  
else{  
    napravi ono;  
}
```

- Primjer5.

```
var n = prompt("Unesi broj: ");  
if (n < 20 || n > 50) {  
    document.write("Kvadrat broja je: " + n*n);  
} else{  
    alert("Broj mora biti manji od 20 ili veći od 50!");  
}
```

- Primjer 6

```
var god = prompt("Unesi broj godina: ");  
if (god>12 && god<20) {  
    alert("Ti si tinejđer")  
}  
else if(god > 20) {  
    alert("Ti si odrasla osoba")  
}  
else{  
    alert("Ti si dijete");  
}
```

Zadaci

1. Skripta pita korisnika da li mu se sviđa film “Planet majmuna”. Ako korisnik potvrdi, skripta obavijesti “Bravo sviđaju ti se klasici” , ako korisnik odbaci tvrdnju skripta obavijesti “Vi ste više za modernu tematiku”.
2. Napisati skriptu koja traži korisnika da unese broj, a nakon toga obavijesti korisnika da li je unešeni broj pozitivan ili negativan.
3. Napisati skriptu koja promjeni pozadinsku boju stranice i to u rozu ako se koristi IE, maslinastu ako se koristi Firefox.
4. Napisati skriptu koja traži od korisnika da u tekstualno polje forme unese broj svojih godina. Ako je taj broj manji od 19 obavijesti korisnika “Vi ste školarac”, ako je broj veći obavijesti “Teško je biti odrastao 😊 ”

Objekt Date i String

Objekt Date

- Da bi koristili vrijeme i datum sa korisničkog računala koristimo Javascript Date() objekt.
- Objekt Date se ravna prema satu na računalu korisnika. Stoga, ukoliko je sat na računalu pogrešno namješten, to nam može poremetiti rad skripte.

Metode

Metoda	Opis
<i>getDay()</i>	vraća brojčano dan u tjednu [0, 6]
<i>getDate()</i>	vraća brojčano dan u mjesecu
<i>getMonth()</i>	vraća brojčano mjesec [0,11]
<i>getFullYear()</i>	vraća tekuću godinu(četveroznamenasti broj)
<i>getHours()</i>	vraća sate (24)
<i>getMinutes()</i>	vraća minute (60)
<i>getSeconds()</i>	vraća sekunde (60)
<i>getMilliseconds()</i>	vraća milisekunde

Primjeri

- Sa: `var datum = new Date();`
Deklariramo varijablu ***datum*** u koju će se očitati današnji datum iz korisnikova računala.
- Sa: `var sati = datum.getHours();`
Deklariramo varijablu ***sati*** u koju se očitava koliko je sati na korisnikovom računalu.
- Sa: `var godina = datum.getFullYear();`
Deklariramo varijablu ***godina*** u koju se očitava koja je godina na korisnikovom računalu.

Zadaci

1. Napiši skriptu koja provjerava ako je više od 20h ispisuje poruku „Dobra večer“, a ako je manje ispisuje „Dobar dan“
2. Napiši skriptu koja provjerava sate i ispisuje odgovarajuću poruku.
 - Ako je manje od 10h ispisuje „Dobro jutro“
 - Ako je više od 10h, a manje od 20h ispisuje „Dobar dan“
 - Ako je više od 20h ispisuje „Vrijeme je za odmor“
3. Napiši skriptu koja ispisuje dan, mjesec i godinu, te ako su dan i/ili mjesec jednoznamenasti broj dodaje „0“ ispred broja. *Npr. 22.01.2020.*
4. Napiši skriptu koja ispisuje dan, mjesec i godinu, te tekstualno ispiše dan u tjednu, te ako su dan i/ili mjesec jednoznamenasti broj dodaje „0“ ispred broja. *npr. Ponedjeljak, 22.10.2020.*
5. Napisati skriptu koja traži od korisnika da u tekstualno polje forme upiše godinu svog rođenja, te ovisno o tome otvara novu web stranicu.
 - Ako korisnik ima manje od 12 godina otvara: <http://www.igre123.net/>
 - Ako korisnik ima više od 12 godina, a manje od 18 otvara: <http://www.teen385.com/>
 - Ako korisnik ima više od 18 godina otvara: <http://www.posao.hr/>

6. Napisati skriptu koja daje korisniku mogućnost odabira kao na slici. Nakon odabira korisniku se datum ispisuje na ekran u odabranom obliku!
7. Napisati skriptu koja traži od korisnika da u tekstualno polje forme upiše godinu svog rođenja, te ovisno o tome ispisuje na ekran.
 - Ako korisnik ima manje od 14 godina : "Ti ideš u osnovnu školu"
 - Ako korisnik ima više od 14 godina, a manje od 18 : "Ti ideš u srednju školu"
 - Ako korisnik ima više od 18 godina otvara mu se pitanje "Da li studiraš?" Ako odgovori potvrdno ispisuje se na ekran "Ti si student" u suprotnom se ispisuje "Ti si radnik!".
8. Napiši skriptu koja pita korisnika da odabere želi li datum prikazati:
 - u standardnom obliku (Npr. 22.01.2020)
 - uz datum dodatno piše dan u tjednu (Npr.Ponedjeljak, 22.10.2020.)
 - uz datum dodatno piše sat i minute (Npr. 19:02, 22.10.2020.)

Odaberi u kojem formatu želiš ispisati datum

dan/mjesec/godina	<input type="checkbox"/>
dan-mjesec-godina	<input type="checkbox"/>
dan. mjesec. godina.	<input type="checkbox"/>
mjesec/dan/godina	<input type="checkbox"/>
<input type="button" value="Ispiši"/>	

Objekt String

- Objekt string koristimo pri radu s tekstom.
- Najčešće korištene metode i osobine string objekta odnose se na traženje određenih znakova, kombinacija znakova ili mjerenje dužine teksta.
- Te mogućnosti se izdašno koriste pri validaciji formi. Npr, ukoliko adresa e-pošte nema znak @ i . i/ili nema najmanje šest znakova ne može biti valjana te stoga nećemo dopustiti slanje takvih podataka.

Metoda	Opis
indexOf()	Vraća položaj prvog pojavljivanja određenog znaka u stringu. Ako ga ne nađe, vraća vrijednost -1.
match()	Traži određenu kombinaciju znakova u stringu. Ako ju ne nađe vraća vrijednost null.
replace()	Mijenja znak unutar stringa. Ili mijenja string sa zadanim stringom.
search()	Traži određen znak u stringu. Ako ga ne nađe, vraća vrijednost -1.
substr()	Izdvađa zadani broj znakova iz stringa počevši od zadanog početka.

Svojstvo	Opis
length	Vraća broj znakova u stringu.
prototype	Omogućava dodavanje metoda i osobina objektu.

Primjeri

- ```
var txt="Pozdrav sivma!";
var duljinaTeksta =txt.length;
document.write(duljinaTeksta);
```
- ```
var txt="Pozdrav svima!";  
document.write(txt.indexOf("d") + "<br />");  
document.write(txt.indexOf("svima") + "<br  
>");  
document.write(txt.indexOf("Svima"));
```

- Mogućnosti uređivanja teksta pomoću Javascripta:

http://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_str_style

- Valjana email adresa:

Zadaci

1. Napisati skriptu koja traži od korisnika da putem forme unese svoje ime i prezime. Skripta provjerava da li je korisnik dobro popunio podatke. Ako nije obavijesti ga – “Morate popuniti podatke”. Ako ispuni na ekran se ispiše poruka “Ime Prezime, hvala! Ugodan dan!”
2. Dopuniti 1. zadatak tražeći od korisnika da unese e-mail adresu. Skripta provjerava da li je adresa duža od 6 znakova, ako nije ispisuje se poruka “E-mail adresa nije valjana!”
3. Dopuniti 2. tako da skripta provjerava postoji li @ znak.
4. Dopuniti 3. tako da skripta provjerava da li je znak @ na točnoj poziciji(min 3 znaka moraju biti prije unaka @).
5. Dopuniti 4. tako da skripta provjerava postoji li .
6. Dopuniti 5. tako da skripta provjerava da li iza znaka . postoje min 2 znaka.
7. Dopuniti 4. tako da skripta provjerava da li se znak @ nalazi

SWITCH

Zadatak 1

- Napiši skriptu koja će pitati posjetitelje da ocjene ljepotu dana ocjenom od 1 do 7. Za svaku ocjenu skripta daje odgovor: “Danas vam je dan za (broj koji je upisao)!”

Switch sintaksa

```
switch(n){  
    case 1:  
        napravi to i to;  
    break;  
    case 2:  
        napravi to i to;  
    break;  
    default:  
        kod koji će se izvršiti ako nije izvršen uvjet 1 ili 2  
}
```

Zadatak 1 - switch

```
var ocjena = prompt("Ocijeni ljepotu dana ocjenom od 1 do 7");
```

```
switch(ocjena){  
    case"1": alert("Danas vam je dan za 1");  
    break;  
    case"2": alert("Danas vam je dan za 2");  
    break;  
    case"3": alert("Danas vam je dan za 3");  
    break;  
    case"4": alert("Danas vam je dan za 4");  
    break;  
    case"5": alert("Danas vam je dan za 5");  
    break;  
    case"6": alert("Danas vam je dan za 6");  
    break;  
    case"7": alert("Danas vam je dan za 7");  
    break;  
    default: alert("Programiranje je zakon");  
}
```

Zadaci:

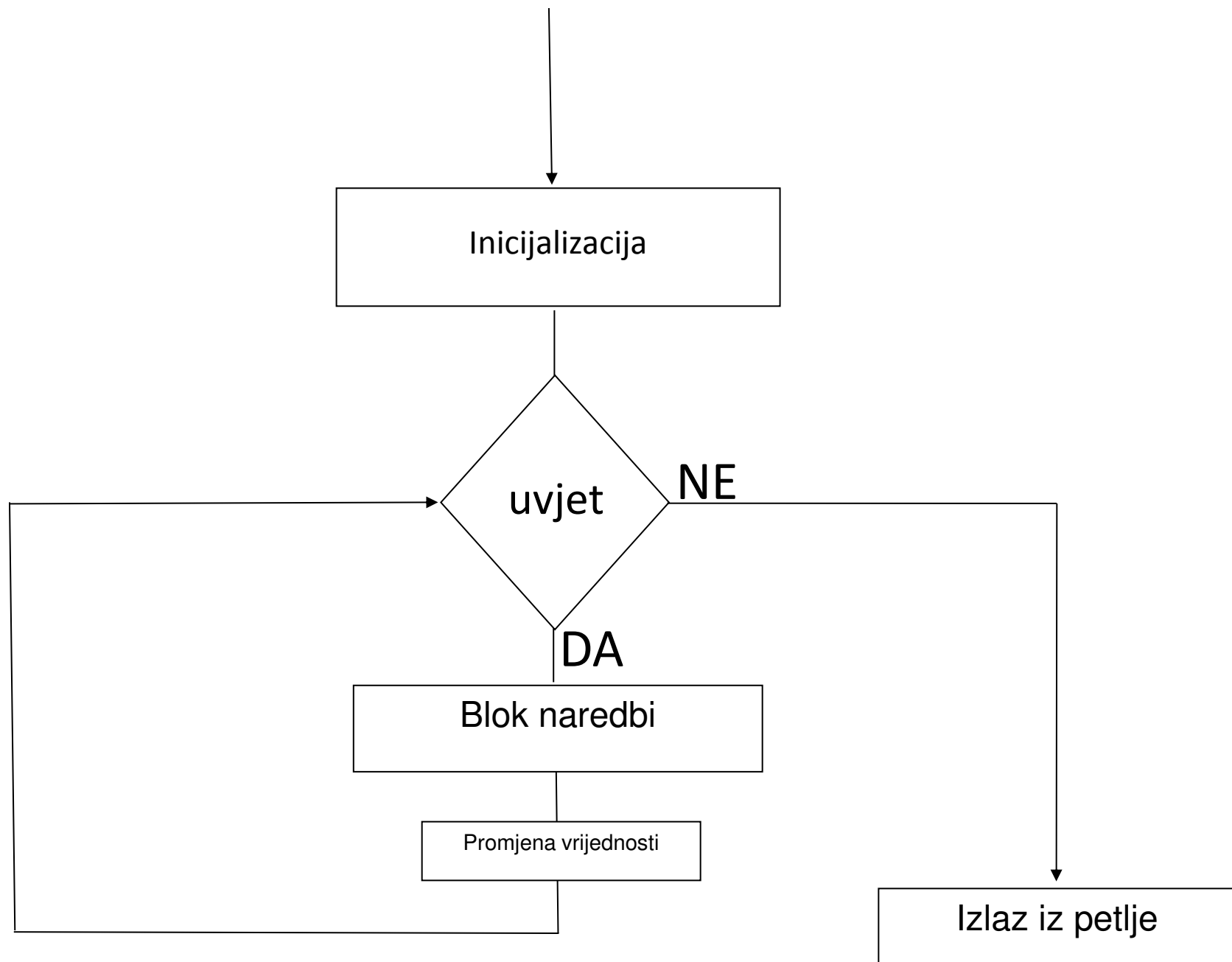
1. Napiši skriptu koja ovisno o danu u tjednu ispisuje poruku:
 - Ako je petak ispiše “Jeeej napokon petak”
 - Ako je subota ispiše “Večeras ideeeem vaaan!”
 - Ako je nedjelja ispiše “Spavam cijeli dan!”
 - Ako je radni dan ispiše “Kad će taj petak!”
2. Napiši skriptu koja ispisuje dan, mjesec i godinu, te tekstualno ispiše dan u tjednu.
3. Napiši skriptu koja ispisuje dan, mjesec i godinu, te tekstualno ispiše tekući mjesec. Npr. 1. Ožujak, 2012.
4. . Napiši skriptu koja pita korisnika da upiše najdražu boju (crvena/zelena/plava/žuta/crna/bijela). Ovisno o tome koju boju upiše u tu boju se promijeni pozadinska boja
5. Dopuni 4. zadatak tako da se boja teksta promijeni u kontrastnu boju u odnosu na odabranu pozadinsku boju.

Petlje

- Često nam je potrebno istu naredbu ili grupu naredbi izvršavati veći broj puta ili sve dok je neka tvrdnja istinita. U takvim slučajevima koristimo petlje. Petlje imaju sposobnost da istu radnju ponavljaju dok je neki uvjet zadovoljen.
- Postoje for, for... in, while i do... while petlje.

Što se to u **for** petlji zbiva?

- Sintaksa **for** petlje izgleda ovako:
 - **for** (početni izraz; uvjet; promjena vrijednosti)
 {
 naredbe;
 }
- Dakle, u **for** petlji postoje tri parametra koji su odvojeni točkom sa zarezom.

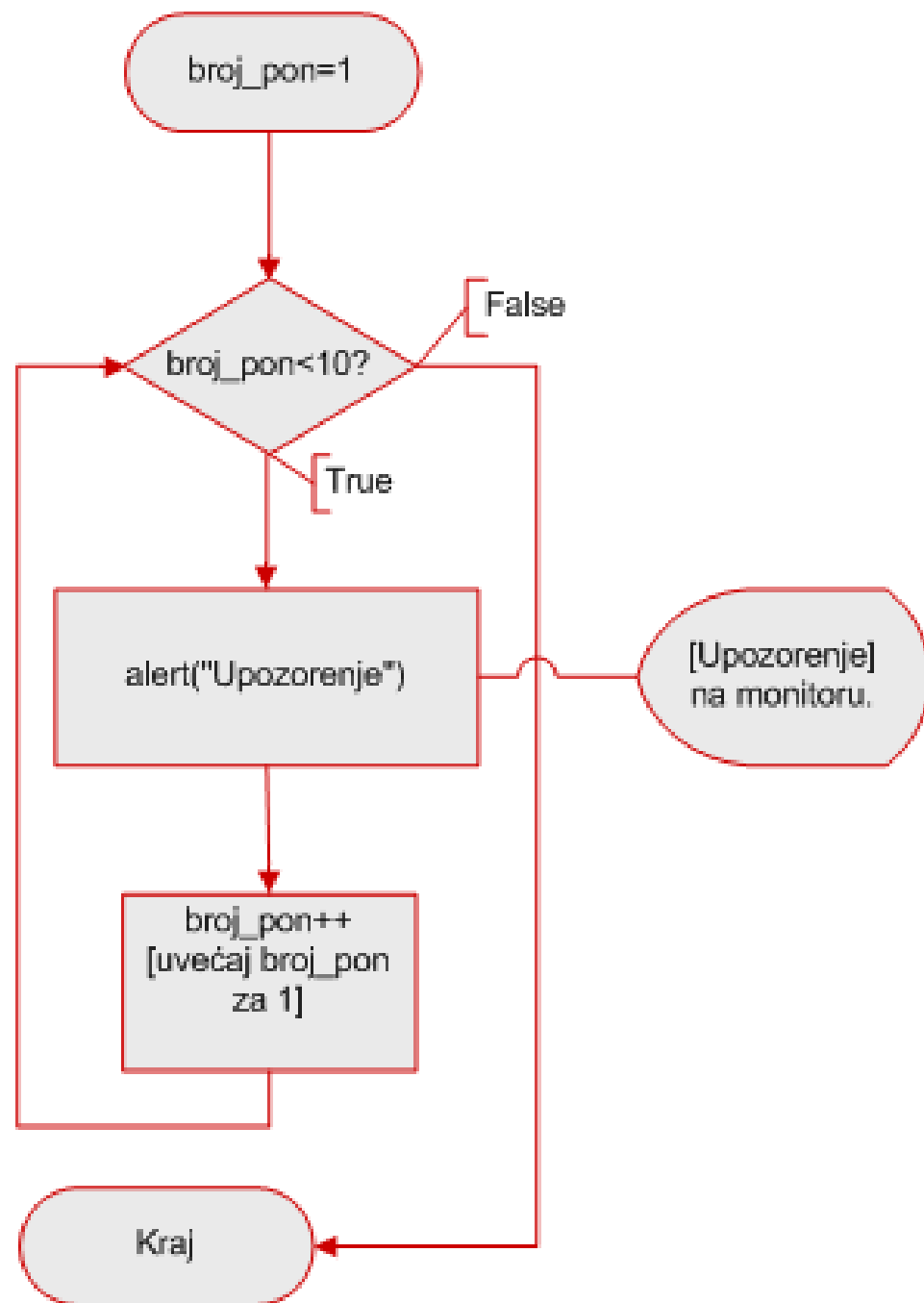


Primjer

```
for(br_ponavljanja=1; br_ponavljanja<=10;  
    br_ponavljanja++){  
    alert("Ovo je upozorenje pod rednim  
    brojem" + br_ponavljanja + ". ");  
}
```

Primjer

- **Početni izraz**
 - pridodaje varijabli `br_ponavljanja` vrijednost 1.
- **Uvjet.**
 - U drugom parametru postavljamo uvjet (`br_ponavljanja <= 10`) koji mora biti zadovoljen da petlja nastavi rad.
- **Promjena vrijednosti.**
 - Treći parametar (`br_ponavljanja++`) uvećava vrijednost varijable `br_ponavljanja` za 1.



Primjeri

```
1 . for (i=0;i<=5;i++)  
  {  
    document.write("Broj prolaska petlje je: " + i);  
    document.write("<br />");  
  }
```

Zadatak: Nacrtaj dijagram toka za prvi primjer.

```
2. for (i = 1; i <= 6; i++){  
    document.write("<h" + i + ">Ovo je naslov  
    razine h" + i + "</h" + i + ">");  
}
```

Zadatak: Nacrtaj dijagram toka za drugi primjer.

```
3. for (var x = 1; x <= 12; x++) {  
    rezultatMnozenja = 12 * x;  
    document.write("12 * " + x + " = " +  
    rezultatMnozenja + "<br>");  
}
```

4. Koliko puta će se sljedeća skripta izvršiti?

```
for (a = 0; a <= 10; a = a + 2) {  
    ... naredbe ...  
}
```

5.

```
Suma=0;
```

```
for(i=0;i<10;i++){  
    broj=prompt("Unesi "+(i+1)+". broj: ");  
    broj = Number(broj);  
    Suma = Suma + broj;  
}
```

```
document.write(Suma);
```

Zadaci

1. Napiši skriptu koja računa aritmetičku sredinu 5 brojeva koje je korisnik upisao u prompt okvira.
2. Promijeni 1. zadatak tako da korisnik sam odredi koliko će brojeva unositi.
3. Napisati skriptu koja ispisuje tablicu množenja od 1 do 10 s brojem koji korisnik odabere.
4. Napiši skriptu koja računa sumu brojeva koje korisnik unese. Korisnik odlučuje koliko brojeva unosi.
5. Napiši skriptu koja ispisuje na ekran vrijednost prvih 50 Celzijevih stupnjeva u *Fahrenheit*. Formula za pretvaranje C u F je **$F = (9/5 * C) + 32$**
6. Promijeni 5. zadatak tako da skripta pita korisnika u kojem rasponu želi konverziju.