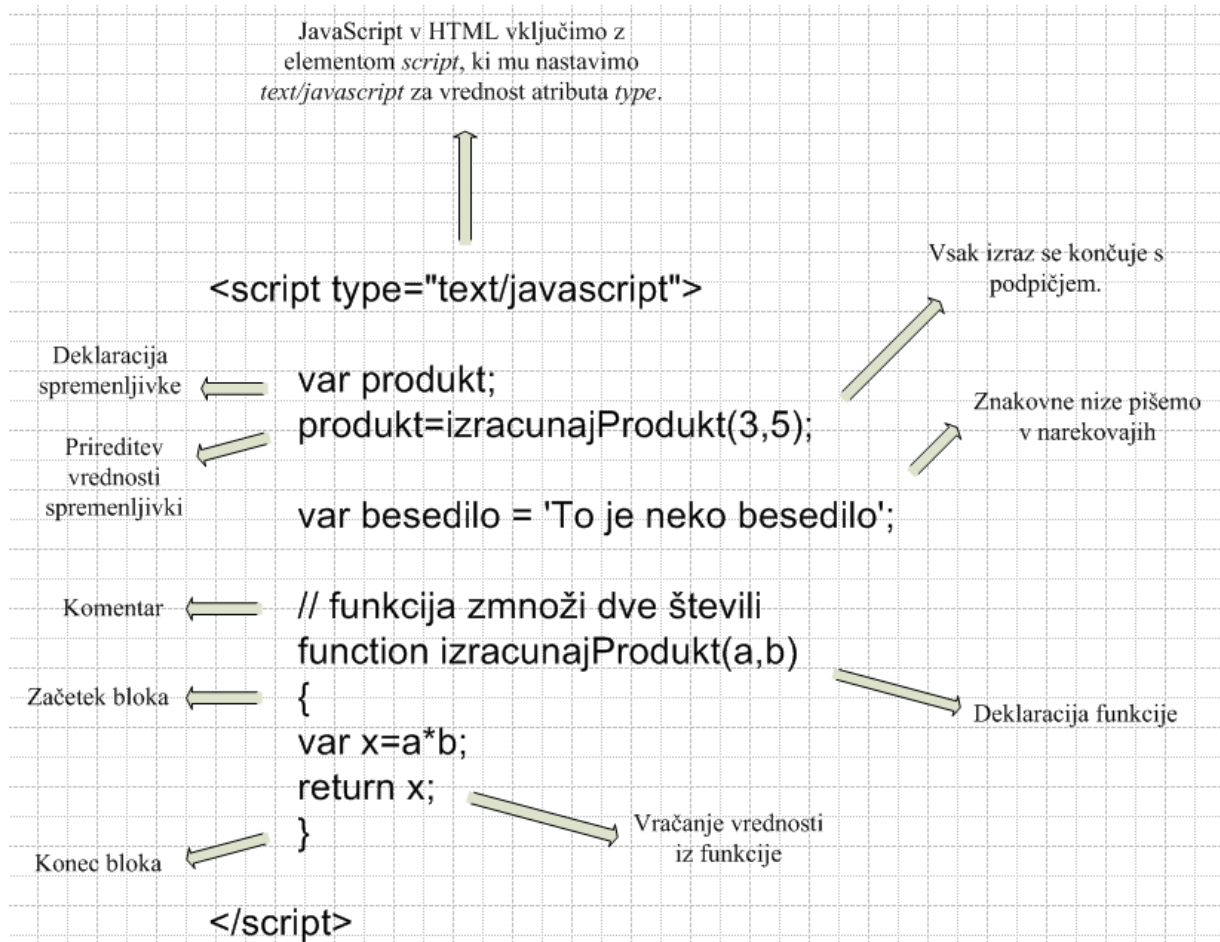


Vaja 8 - JavaScript

JavaScript

- je skriptni jezik, ki se izvaja na spletnem odjemalcu;
- je bil ustvarjen z namenom dodajanja večje interaktivnosti spletnim stranem.

1. Osnove jezika



Slika 1: Osnove jezika JavaScript

- HTML vsebino generiramo z izrazom *document.write()*.
- Za spajanje nizov uporabljamo operator +

2. Uporaba JavaScripta v spletni strani

JavaScript kodo lahko v spletno stran vključimo na sledeče načine.

- **V glavo HTML dokumenta** - To kodo je navadno **potrebno poklicati** iz telesa dokumenta, ko se zgodi nek dogodek (npr. stran se naloži, uporabnik pritisne na gumb, spremeni se neka vrednost v obrazcu) ali pa iz druge JavaScript kode.
- **V telo dokumenta** - Ta koda **se izvede sproti** ob nalaganju dokumenta, običajno pa pripomore k dinamičnemu tvorjenju vsebine dokumenta (npr. izpis datuma)
- Če isto JavaScript kodo uporablja več spletnih strani hkrati, potem jo vključimo **v ločeno datoteko**, ki ima končnico *.js*. To datoteko povežemo s spletnimi stranmi, tako da dodamo **v glavo HTML dokumentov** vrstico

```
<script type="text/javascript" src="datotekaZJavaScriptKodo.js"></script>
```

3. Dogodki

- Izvajanje JavaScript kode v spletnih straneh navadno sprožijo *dogodki*.
- Dogodki (ang. *events*) (večinoma) ustrezajo uporabnikovim dejanjem, ki jih lahko programsko prestrežemo in pripravimo odziv nanje.
- Dogodke vedno opazujemo na nekem HTML elementu, kar nakažemo z atributom, ki ustreza opazovanemu dogodku. Nekatero pogoste dogodke prikazuje Tabela 1.

```
<ime_elementa ime_dogodka="programska_koda">
```

- Kot vrednost atributov vpišemo JavaScript kodo, ki se izvede, ko (če) se dogodek zgodi. Koda navadno vsebuje klic neke predhodno definirane funkcije. **Funkcije v JavaScriptu se navadno izvedejo le ob nekem dogodku.**

HTML atribut	pomen dogodka
onload	stran ali slika je naložena
onfocus	element je pridobil pozornost (<i>focus</i>)
onblur	element je izgubil pozornost
onchange	spremenila se je vsebina polja
onsubmit	uporabnik je sprožil odpošiljanje podatkov iz obrazca. Dogodek navadno uporabimo, če želimo preveriti podatke, preden se ti dejansko odpošljejo na strežnik.
onclick	uporabnik je »kliknil« na element
onmouseover	miškin kazalec se je premaknil nad element
onkeydown	pritisnjena je bila tipka na tipkovnici

Tabela 1: Dogodki in pripadajoči HTML atributi



Nekega dogodka ne moremo opazovati prav na vsakem HTML elementu. Dogodek *onsubmit* lahko tako na primer opazujemo samo na elementu *form*.



Celoten seznam dogodkov in elementov, s katerimi so posamezni dogodki povezani, je dostopen na spletnem naslovu http://www.w3schools.com/jsref/jsref_events.asp.

Primer uporabe dogodka

```
<form method="post" action="http://mojStreznik/blabla.asp"
  onsubmit="return preveriObrazec(this)">
```

HTML atribut, ki označuje dogodek odpošiljanja obrazca

JavaScript koda, ki se ob nastopu dogodka izvede

Slika 2: Primer uporabe dogodka pošiljanja podatkov iz obrazca

Slika 2 prikazuje primer uporabe dogodka, ki se nanaša na odpošiljanje podatkov iz obrazca:

1. Uporabnikov klik na gumb za odpošiljanje podatkov iz obrazca ustvari nov dogodek **onsubmit**.
2. Izvede se JavaScript koda, ki je vpisana kot vrednost atributa **onsubmit**.
3. V primeru na Sliki 2 ta koda vsebuje klic funkcije **preveriObrazec**, ki mora biti predhodno definirana (npr. v glavi dokumenta ali v ločeni datoteki).
4. Z uporabo rezervirane besede **return** dosežemo, da se odpošiljanje podatkov izvede le, če funkcija **preveriObrazec** vrne vrednost *true*. Če funkcija vrne *false*, je odpošiljanje preklicano.

4. Pojavna okna

Pojavna (»popup«) okna so prikazana na Sliki 3.



Slika 3: Pojavna okna *alert*, *confirm* in *prompt*.

alert('sporočilo');	v oknu izpiše podano sporočilo
confirm('sporočilo');	v oknu izpiše sporočilo in ponudi uporabniku izbiro <i>OK</i> oziroma <i>Cancel</i> . Izbira <i>OK</i> vrne vrednost <i>true</i> , izbira <i>Cancel</i> pa <i>false</i> .
prompt('sporočilo', 'privzeta vrednost');	od uporabnika zahteva vnos vrednosti. Izbira <i>OK</i> vrne vpisano vrednost, izbira <i>Cancel</i> pa vrne prazno vrednost (<i>null</i>).



Za zapis znakovnih nizov v JavaScriptu uporabljajte enojne narekovaje, da se izognete konfliktom z dvojnimi narekovaji v HTML značkah.



Če želimo besedilo v pojavnih oknih pomakniti v novo vrstico, uporabimo t.i. »ubežno zaporedje« `\n`.



Uporaba pojavnih oken v spletnih straneh ni priporočljiva, saj je za uporabnike moteča. Pojavna okna so priročna predvsem za izpisovanje pri hitrem testiranju strani.



1. S spletnega mesta <http://www.lkn.fe.uni-lj.si/gradiva/ST/vaja8/> prenesite datoteko *obrazec.html*, lahko pa za reševanje nalog uporabite tudi lastno spletno stran.
2. S pomočjo dogodkov in pojavnih oken dopolnite prenesen dokument:
 - a. Izberite si nek HTML element. Vsakič, ko se z miško zapeljete čez izbran element, naj se v pojavnem oknu izpiše poljubno sporočilo.
 - b. Ko odpošljete podatke iz obrazca, naj se pojavi okno, ki od vas zahteva potrditev vaše odločitve.

5. Delo z »vgrajenimi« JavaScript objekti

1. **Objekt *String*** omogoča delo z znakovnimi nizi.
 - Primera:
 - *znakovni_niz.length* vrne dolžino znakovnega niza.
 - *znakovni_niz.toUpperCase()* pretvori vse črke niza v velike.



Celoten seznam metod objekta *String* je dostopen na spletnem naslovu http://www.w3schools.com/jsref/jsref_obj_string.asp.

2. **Objekt *Date*** omogoča delo z datumom in časom.
 - Pred uporabo moramo objekt *Date* ustvariti, npr.: *var mojDatum=new Date()*
 - Trenutni datum in čas lahko izpišemo tako, da izpišemo kar vrednost spremenljivke, ki smo ji priredili objekt *Date* (v našem primeru *mojDatum*).
 - Ostale metode kličemo na objektu samem. *mojDatum.getFullYear()* tako na primer vrne trenutno leto.



Celoten seznam metod objekta *Date* je dostopen na spletnem naslovu http://www.w3schools.com/jsref/jsref_obj_date.asp.

3. **Objekt *Math*** vsebuje matematične funkcije in konstante.

- Primera:
 - ***Math.PI***; vrne število *pi*.
 - ***Math.round()***; zaokroži število na najbližje celo število.



Celoten seznam metod in lastnosti objekta *Math* je dostopen na spletnem naslovu http://www.w3schools.com/jsref/jsref_obj_math.asp.



1. Na vrh spletne strani izpišite trenutni datum in razliko v minutah med lokalnim časom in časom UTC (*Coordinated Universal Time*).
2. Razliko pomnožite z naključnim številom med 0 in 1 in rezultat zaokrožite. Uporabite funkcije iz razreda *Math*.
3. Dodajte nekaj naključnosti tudi prikazu spletne strani. Na spletu poiščite dve sliki. Vsakič, ko stran naložite, naj prikaže ena izmed obeh slik, pri čemer je verjetnost prikaza vsake slike 50%.

Pomoč:

- V HTML vključite element, ki predstavlja sliko, ter mu dodelite *id*. V JavaScriptu lahko nato pot do slike nastavljate z izrazom `document.getElementById('id_ki_ste_ga_dodelili').src='pot_do_slike'`
- Za ustvarjanje naključnosti in odločanja, katero sliko boste prikazali, si pomagajte z `Math.random()` in *if* stavkom.

6. Validacija na odjemalcu

Validacija (preverjanje vsebine podatkov) se navadno izvede pred pošiljanjem podatkov na strežnik.



Preverite podatke iz obrazca, preden so ti poslani strežniku:

1. Ugotovite, kateri dogodek ustreza odpošiljanju obrazca.
2. Dogodek vključite kot atribut v tisti HTML element, kamor sodi.
3. Vrednosti atributa priredite rezultat izvajanja funkcije, ki bo preverjala obrazec (glej *Dogodki / Primer uporabe dogodka*).
4. V glavi HTML dokumenta ustvarite ogrodje funkcije, v kateri boste izvedli preverjanje.
5. Podatke iz obrazca v funkcijo posredujete tako, da pri klicu funkcije kot parameter uporabite ključno besedo **this**, torej *ime_vase_funkcije(this)*.

Rezervirana beseda **this** (»ta«) predstavlja programski objekt, ki je programska predstavitev HTML elementa, v katerem je ključna **this** besedna uporabljena. V našem primeru bo torej **this** predstavljal programski objekt vrste **form**, saj bo ta ključna beseda uporabljena v znački istoimenskega HTML elementa. Objekt vrste **form** predstavlja programsko predstavitev obrazca skupaj z vsemi njegovimi gradniki in vrednostmi, ki so v gradnike trenutno vpisane.

6. Znotraj funkcije lahko do podatkov iz obrazca dostopate preko imena spremenljivke, ki ste ga navedli v glavi funkcije (torej *obrazec* v primeru na Sliki 4). Do vrednosti posameznega polja v obrazcu lahko dostopate prek imena polja v obrazcu, ki mu dodate lastnost **value**. Na Sliki 4 je prikazan izraz, s pomočjo katerega dostopate do vrednosti, ki je bila vpisana v polje »ime«.

```
function preglejobrazec(obrazec)
{
  //spodnji izraz vrne vrednost, vpisano v polje 'ime':
  obrazec.ime.value;
}
```

Slika 4: Prebiranje vrednosti iz obrazca

7. Dopolnite funkcijo tako, da bo preverila vrednosti polj v obrazcu. Če vsaj ena vrednost manjka (prazen niz), naj funkcija vrne *false*, v nasprotnem primeru pa *true*.
8. V primeru, da je uporabnik pozabil izpolniti določeno vnosno polje, naj se ob njem izpiše obvestilo.

7. Uporaba časovnika

Z uporabo metode *setTimeout()* je mogoče zakasnjeno izvesti programsko kodo. V omenjeno metodo posredujemo kodo in časovno zakasnitev v milisekundah.



1. Ustvarite funkcijo, ki bo vsako sekundo spremenila sliko na spletni strani. Izvedete lahko ciklično preklapljanje med slikama, ki ste ju že uporabili pri naključnem prikazu. Ciklično preklapljanje naj se sproži, ko kliknete na sliko.

Pomoč: Prvič naj se funkcija pokliče ob nekem dogodku. Nato naj v časovnih intervalih ene sekunde funkcija z uporabo metode *setTimeout()* kliče sama sebe.



Več informacij o metodi *setTimeout()* in časovnikih lahko dobite na spletni strani http://www.w3schools.com/js/js_timing.asp.

Naslednjič

- Pregledali bomo hierarhijo vgrajenih objektov v JavaScriptu in nekaj teh objektov uporabili.
- Obravnavali bomo objektni model dokumenta (t.i. DOM, *Document Object Model*) in ga uporabili za branje in nastavljanje lastnosti obstoječih elementov na spletni strani.

Temeljito se seznanite s hierarhijo vgrajenih objektov in principom objektnega modela DOM!

Literatura

1. Spletni tečaji JavaScripta:
 - <http://www.w3schools.com/js/default.asp>
 - <http://www.howtcreate.co.uk/tutorials/javascript/important>
2. Prosojnice s predavanj