

Spletne tehnologije, UL, FRI (VSP) ST 3-JS, jQuery

Mira Trebar

Vsebina

- ▶ JavaScript
- ▶ DOM
- ▶ jQuery

- ▶ Literatura:
 - ▶ David Sawyer McFarland, JavaScript & jQuery: The Missing Manual, O'Reilly Media, 2014
 - ▶ Brad, Dayley, jQuery and JavaScript Phrasebook, Addison-Wesley, 2014
 - ▶ W3Schools, <http://www.w3schools.com>

Problem: kombinacija loto (1-39)

Računalnik bo generiral kombinacijo Loto	
Generiraj	Izpiši kombinacijo
Kombinacija	
Kombinacija loto: 17 , 18 , 2 , 23 , 28 , 5 , 5	
Nova kombinacija	

- ▶ Kako bomo to naredili?

JavaScript (JS) - uvod

- ▶ Skriptni jezik - se interpretira ali tolmači
- ▶ Dinamična spletna stran
- ▶ Interaktivnost
- ▶ Vključen je v HTML

- ▶ Uporaba:
 - Dodajanje programskih funkcionalnosti
 - Odziva se na dogodke
 - Omogoča branje in spreminjanje elementov HTML
 - Potrditev obrazcev
 - Generiranje piškotkov ('cookies')
 - Komunikacija s strežnikom



JS - definicija

- ▶ Brskalnik ne podpira JavaScript – koda se prikaže kot vsebina spletne strani.
- ▶ Program je del kode HTML v glavi (funkcije) ali v telesu

`<script type="text/javascript">` tip zapisa je navaden tekst

...

`</script>`

`<script>`

HTML 5 – ni potrebno določiti tipa

...

`</script>`

Program je zapisan v samostojni datoteki z imenom *test.js*

`<head>`

`<script src="test.js"> ... </script>`

`</head>`

JS - enostaven program

- ▶ Spremenljivke, prireditve, operacije, izpis v oknu brskalnika
- ▶ Podpičje označuje konec stavka

```
<script>
```

```
    a=3;
```

```
    var a, B;
```

```
    document.write(a);
```

```
    var B=5;
```

```
    a=B-2;
```

```
    document.write("<br />", B);
```

```
    document.write("<br />");
```

```
    document.write(a, "<br />");
```

```
    document.write(" a=" + a);
```

```
</script>
```

JS - izrazi

- ▶ Komentar:

`// opis ali /* opis */`

- ▶ Ukazi/Stavki/Izrazi:

- `var a, B;` `// spremenljivka`
- `a=B+3;` `// priredilni operator`
- ...

- ▶ **Operand** - ena ali več vhodnih vrednosti.

- ▶ **Operator** je znak ali skupina znakov za označevanje operacije.

- ▶ **Izraz** je kombinacija operandov in operatorjev.

- ▶ Prednost in vrstni red izvajanja: operacije, operatorji, oklepaji

- `a+b*c-d`
- `(a+b)*(c-d)`

JS - spremenljivke

document.write(a) - ni deklarirana, JavaScript se izvede samo do tega

var x, y;

nedoločena spremenljivka ('undefined'):

y=x; x ni definirana

številski tip (ang. number) - (64-bitna plavajoča vejica (standard IEEE 754))

y = x - 2; - ni število - NaN - (ang. Not a Number), ker je x undefined

a=1/0 ; - neskončno (Infinity)

y = 10; - število

Boolov tip :

x = true; - x=pravilno,

y = false; - y=napačno

znakovni niz (ang. char):

y="a"; " " - dvojne navednice, ali ' ' - enojne navednice

JS - operatorji

- ▶ Unarni (1 op), binarni (2 op), ternarni (3 op)

- ▶ Aritmetični operatorji (številski tip):

$a + b$, $a - b$, $a * b$, a / b , $a \% b$, $++$, $--$

- ▶ Primerjalni operatorji (Boolov tip)

$=$, $==$, $===$, $!=$, $!==$, $<$, $>$, $<=$, $>=$

- ▶ Logični operatorji (Boolov tip)

$\&\&$ - and, $\|\|$ - or, $!$ - not

- ▶ Operatorji z znakovnimi nizi:

vsi primerjalni operatorji, in konkatencija (+)

- ▶ Priredilni in kombinirani operatorji

$=$, $+=$, $-=$, $*=$, $/=$, $\%=$

- ▶ Ternarni pogojni operator: $\text{izraz1} ? \text{izraz2} : \text{izraz3}$

if $\text{izraz1} == \text{true}$ then izraz2 else izraz3 // (izraz1 je pogoj)

- ▶ `typeof()` – določa tip operanda

JS - operatorji (Primeri)

1. Število in Boolova spremenljivka

4 + false = 4

4 + true = 5

2. Število in znak

4 + "5" = "45" // operator + pomeni konkatencijo

1 * "5" = 5

3. Operator + pomeni konkatencijo

<script>

var x=10,y=20;

document.write("
" + x-y);

Izpis: NaN

document.write("
" + (x-y));

-10

var z=x-y;

document.write("
" + z);

-10

</script>

4. Primerjalni operatorji

```
<script>
```

```
var a=3,b="3";
```

```
document.write(a==b);
```

```
document.write(a===b);
```

Izpis: true

false

```
</script>
```

5. Ternarni pogojni operator (? :)

```
<script>
```

```
var q, w, maks;
```

```
q = 3;
```

```
w = 4;
```

```
maks = q > w ? q : w;
```

```
document.write(maks);
```

pogoj ni izpolnjen -> max=w

Izpis: 4

```
</script>
```

JS – stavki

Tipi:

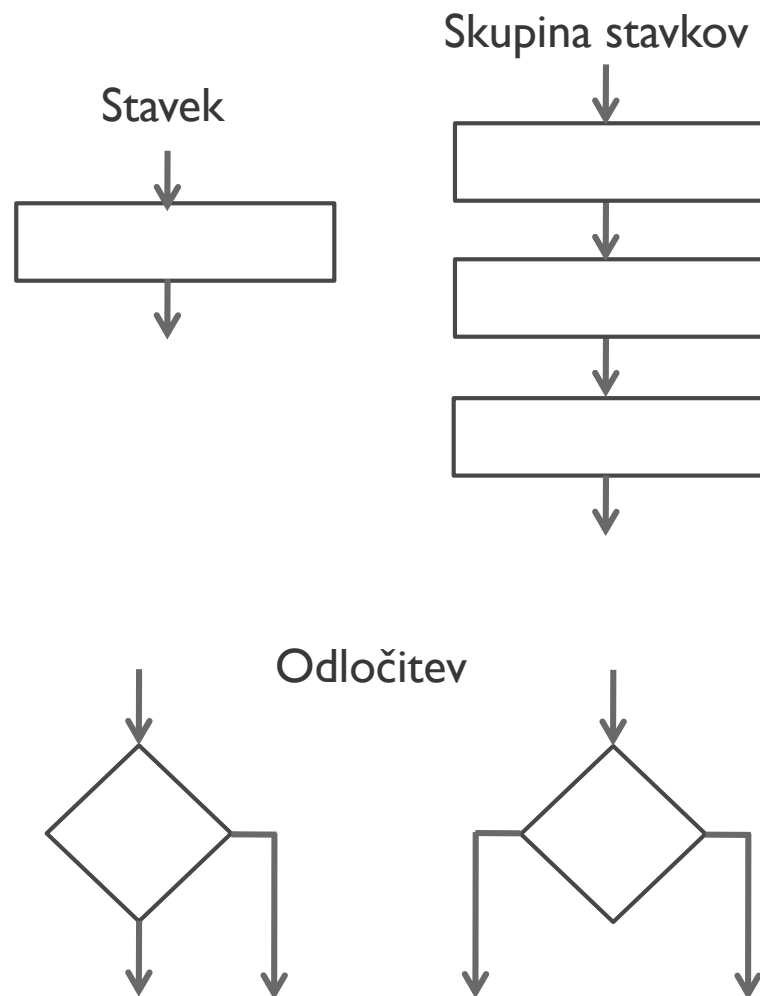
- ▶ priredilni stavek ($a=b;$)
- ▶ prazen stavek ($;$)
- ▶ deklaracija ($\text{var } a;$)
- ▶ sestavljeni stavek

Ločevanje med stavki:

- ▶ stavek v eni vrstici
- ▶ podpičje na koncu stavka
- ▶ zaviti oklepaj in zaklepaj za sestavljeni stavek

Diagram poteka:

- ▶ pravokotnik
- ▶ romb
- ▶ povezava s puščico



► **Pogojni stavki**

- if (pogoj) akcija;
- If (pogoj) akcija else alternativa;

► **Ponavljalni stavki**

- for (inicializacija; pogoj; šteje) izvajaj;
- while (pogoj) akcija;
- do akcija while (pogoj);

► **Izbirni stavek**

- switch (izraz) { case v1: st_1; break; ..., case vn: st_n; break; default: ostalo; }

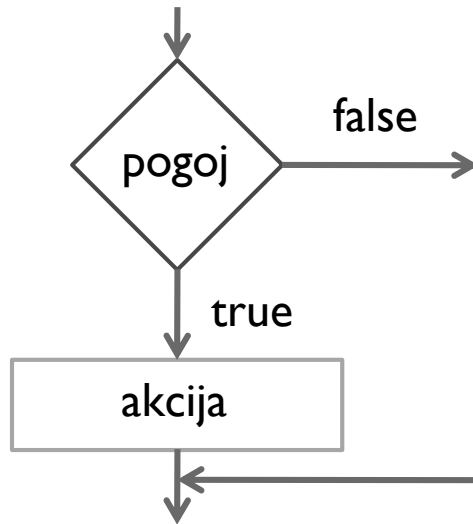
► **Drugi stavki**

- break;
- continue;

JS – pogojni stavek (if in if-else)

if (pogoj)

akcija;



Primer:

A=-3

if (A<0)

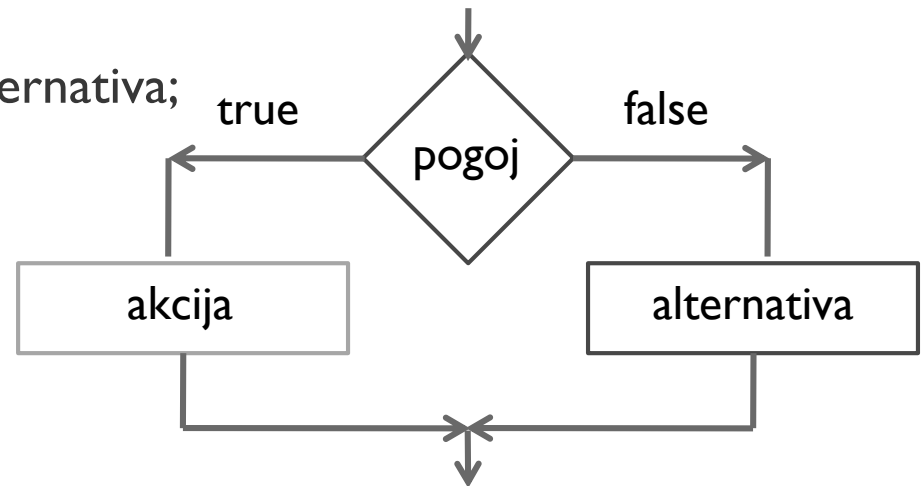
A=A*(-1);

if (pogoj)

akcija;

else

alternativa;



Primer:

a=3

if (a<0)

c=2;

else

c=-2;

JS – ponavljalni stavek (for)

```
for (inicializacija; pogoj; štej)  
    izvajaj;
```

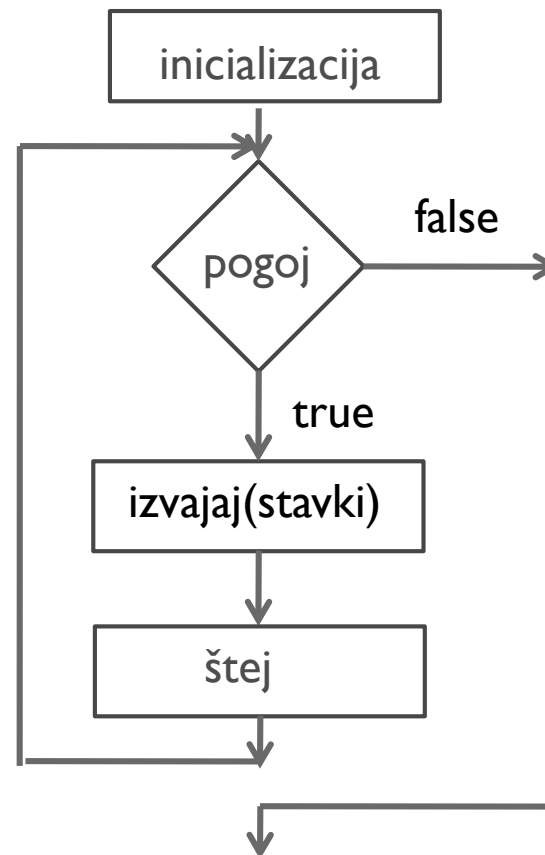
Primer:

```
a=4
```

```
for (i=1; i<= a; i++)
```

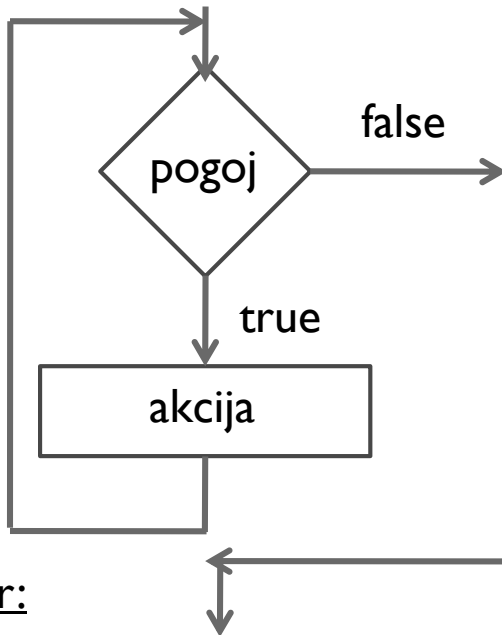
```
    b=i*2;
```

i	b
1	2
2	4
3	6
4	8



JS – ponavljalni stavek (while)

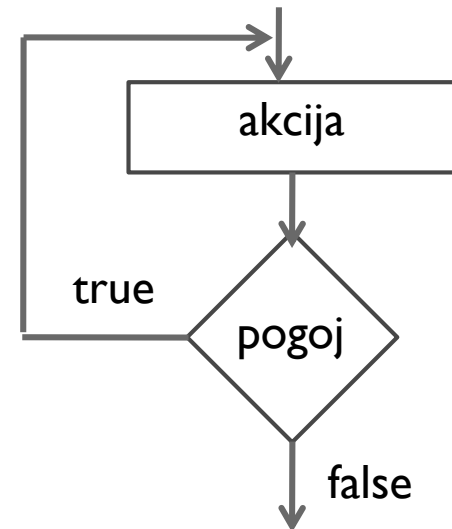
while (pogoj)
akcija;



Primer:

```
i=0;  
while (i<3)  
    i=i*2;
```

do
 akcija;
while (pogoj);



Primer:

```
i=0;  
do  
    i=i*2;  
while(i<3)
```


JS – izbirni stavek (switch)

switch () - povzroči, da se izvede ena od ponujenih možnosti.

```
switch (izraz)
```

```
{
```

```
    case v1: stavek1; break;
```

```
    case v2: stavek2; break;
```

```
    ....
```

```
    case vn: stavekn; break;
```

```
    default: ostalo;
```

```
}
```

// ni nujno potrebno, lahko opustimo!

JS – break, continue

- Zaustavitev izvajanja v ponavljalnem stavku: **break;**

```
for (i = 1; i <= 5; i++)
```

```
    if (i==3) break;
```

```
    document.write(i)
```

Izpis: 0 | 2

- Prekinitev izvajanja v ponavljalnem stavku: **continue;**

```
for (i = 1; i <= 5; i++)
```

```
    if (i==3) continue;
```

```
    document.write(i)
```

Izpis: 0 | 2 4 5

- Prazen stavek: podpičje v stavku (;)

```
for (a=3 ; a<b ; a--);
```

```
    ....
```

```
    document.write (a);
```

// ta koda se ne bo izvedla

JS – objekti

- ▶ **Objekt** – posebna vrsta podatkov

- Lastnosti
- Postopki (metode)

- ▶ Zapis objekta

imeObjekta.imeLastnosti

imeObjekta.imePostopka(seznamParametrov)

- ▶ Kreiranje novega objekta:

```
var oseba=new Object();
```

```
oseba.ime="Jaka";
```

```
oseba.priimek="Novak";
```

```
oseba.starost=55;
```

```
document.write(oseba.ime + " je star " + oseba.starost + " let.");
```

Jaka je star 55 let.

► Objekti brskalnika:

► **window** - !!! Ni obvezno !!!!

► lastnosti: document, open(), close()

► postopki: alert(), confirm(), prompt(), open()

► **document** (window.document == document)

► lastnosti: write(), writeln(), ..

► **Math** – standardna knjižnica konstant in funkcij

► lastnosti: konstante (Eulerjeva: $E=2.718282$), $PI=3.141593$, ...)

► postopki:

► abs(x), asin(x), acos(x), atan(x), sin(x), cos(x), tan(x), exp(x), log(x)

► sqrt(x), max(x,y), min(x,y), pow(x,y)

► random(), round(x), ceil(x), floor(x)

▶ **Date** – delo z uro in datumom

▶ konstruktorja:

▶ `Date();`

▶ `Date(leto, mesec, dan, ura, minute, sekunde);`

▶ postopki:

▶ `getFullYear()` - (yyyy), `getMonth()` - (0-11), `getDate()` - (1-31), `getDay()` – (0-6)

▶ `getHours()` – (0-23), `getMinutes()` – (0-59), `getSeconds()` – (0-59)

▶ `setDate()`, ...

▶ **Array** – omogoča shranjevanje več različnih vrednosti

▶ konstruktorji:

`Array()`, `Array(n)` , `Array(v1,v2,...,vn)`

▶ lastnosti: `length`

▶ postopki: `sort()`, `concat()`, `slice()`

► **String** – znakovni niz

- konstruktorji: `String(besedilo)` - ustvari znakovni niz in mu priredi besedilo
- lastnosti: `length` - število znakov v nizu
- postopki:
 - `indexOf(vzorec)` - v znakovnem nizu išče besedilo podano s parametrom vzorec. besedila ni v nizu, vrne -1, sicer pa indeks začetka prvega najdenega besedila.
 - `charAt(indeks)` - vrne znak na mestu indeks v znakovnem nizu
 - `toLowerCase()` - vse črke so pretvorjene male črke
 - `toUpperCase()` - vse črke so pretvorjene v velike črke
 - `substring(začetek, konec)` - izpišejo se znaki od indeksa začetek, do (konec-1).

Kreiranje objektnega tipa:

`new String();` ali vpišemo tekst v oklepaju – dva načina
`var sporocilo=new String("Pojdi na kosilo!");`
`var sporocilo="Pojdi domov!"`

JS – funkcije

Definicija funkcije (*'function definition'*)

```
function imeFunkcije(seznam_parametrov) {  
    // stavki JavaScript, lahko tudi stavek return  
}
```

- v elementu <head>
- formalni parametri

Klic funkcije: (*'function call'*)

```
imeFunkcije(seznam_parametrov)
```

- v elementu <body>.
- dejanski parametri

Primer 1: Funkcije nima parametrov, ne vrne vrednosti

Primer 2: Funkcija nima parametrov, vrne vrednost (return)

Primer 3: Funkcija ima definirane parametre in/ali vrne vrednost

► **Primer 1:**

```
function imeFunkcije() {  
    document.write("Funkcija");  
}
```

imeFunkcije;

Primer 2:

```
function imeFunkcije() {  
    var vrni=5;  
    return vrni;  
}  
a=imeFunkcije();
```

► **Primer 3:**

```
function imeFunkcije(p1, p2, ..., pn)  
{  
    var vrni;  
    // stavki JavaScript  
    return vrni;  
}
```

a=imeFunkcije(v_1, v_2, ..., v_n);

- formalni parametri

- dejanski parametri

Območje spremenljivk

Lokalne spremenljivke:

- ▶ se nanašajo na območje funkcije.
- ▶ so parametri funkcije (p1,...) so lokalne spremenljivke: vrednost se jim nastavi ob klicu funkcije.
- ▶ Globalne spremenljivke:
 - ▶ se nanašajo na območje datoteke.

Obstoj spremenljivke:

- ▶ *Avtomatičen obstoj*- obstaja v času izvajanja funkcije, v kateri je deklarirana. (*lokalne spremenljivke*)
- ▶ *Statičen obstoj* - obstaja ves čas, dokler strani v brskalniku ne zapustimo ali je ponovno ne naložimo (*globalne spremenljivke*).

JS HTML DOM (Document Object Model)

- ▶ Definicija: *"The W3C Document Object Model (DOM) is*
 - *a **platform and language-neutral interface***
that allows programs and scripts to
 - ***dynamically access and update***
the content, structure, and style of a document."
- (http://www.w3schools.com/html/dom_intro.asp)

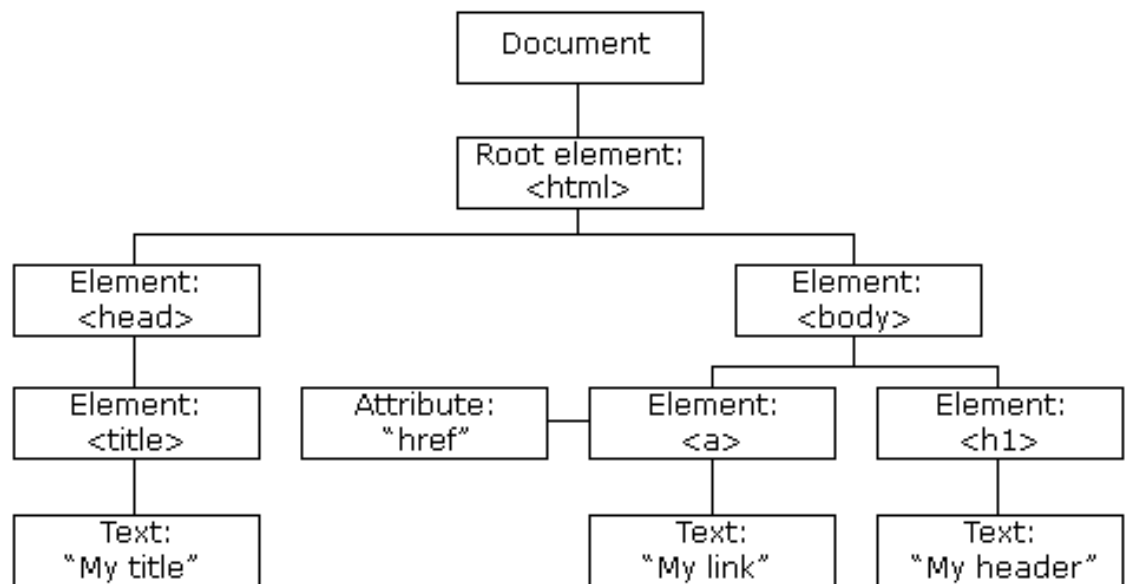
- ▶ je standard W3C (World Wide Web Consortium)
- ▶ W3C DOM standard se deli na:
 - Core DOM - standarden model za vsak strukturiran dokument
 - XML DOM - standarden model za dokumente XML
 - HTML DOM - standarden model za dokumente HTML

JS HTML DOM - definicija

- ▶ Standarden model objekta za HTML
- ▶ Standarden programski vmesnik za HTML
- ▶ Definira:
 - **Objekte** – to so elementi HTML
 - **Lastnosti** vseh elementov HTML
 - **Metode** za dostop do elementov HTML
 - **Dogodke** za vse elemente HTML.
- ▶ HTML DOM je standard za:
 - pridobivanje,
 - spreminjanje,
 - dodajanje,
 - brisanje elementov HTML

- ▶ Naloži se spletna stran – brskalnik kreira DOM spletne strani.
- ▶ Drevesna struktura dokumenta HTML.
- ▶ Vozlišča: dokument, element, atribut, tekst, komentar.
- ▶ Do vseh vozlišč je možno dostopati z JavaScript programom.

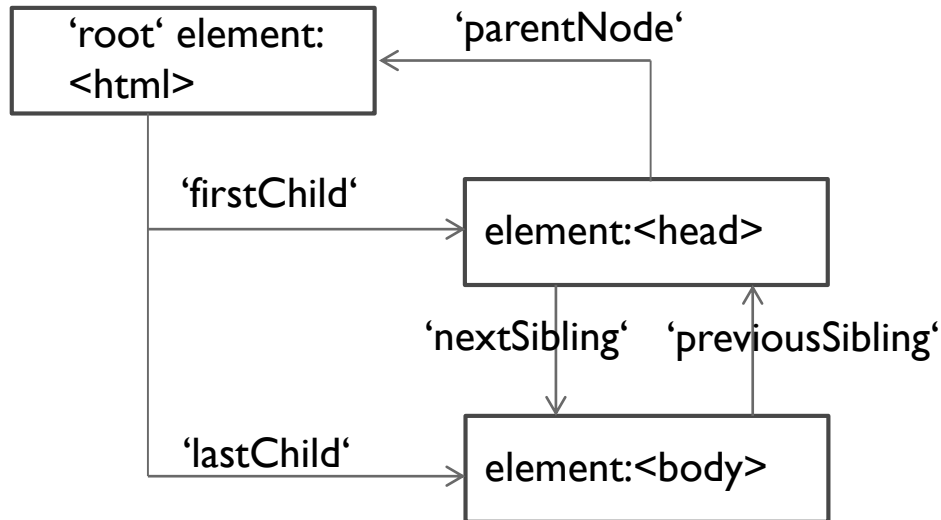
```
<!DOCTYPE html>
<html>
<head>
  <title>My title</title>
</head>
<body>
  <a href="My link"></a>
  <h1>My header</h1>
</body>
</html>
```



<http://www.w3schools.com/html/dom/default.asp>

JS HTML DOM - navigacija

- Relacije med vozlišči: 'parent', 'child', 'sibling'



```
<html>
  <head>
    <title>Uvod v DOM</title>
  </head>
  <body>
    <h1>DOM Lekcija I</h1>
    <p>Pozdrav!</p>
  </body>
</html>
```

- <html> nima vozlišča 'parent'
- <html> je 'parent' od <head> in <body>
- <head> je prvi 'child' od <html>
- <head> ima eno vozlišče 'child' : "Uvod v DOM"
- <h1> in <p> sta 'sibling'
- <h1> ima ...

JS HTML DOM - metode

- Metode HTML DOM – akcije se izvede na elementu HTML
- Lastnosti HTML DOM – vrednosti elementa HTML, se nastavi, spremeni

- Primer:

```
<p id="demo">Sedaj!</p>
```

```
<p id="demo I">Včeraj!</p>
```

```

```

Sedaj!

Včeraj!

Potem!

Danes!



```
<script>
```

```
    element=document.getElementById("demo I");
```

```
    element.innerHTML = "Danes!";
```

```
    document.getElementById("demo").innerHTML = "Potem!";
```

```
    document.getElementById("spr").src = "euro500.gif";
```

```
</script>
```

JS HTML DOM - document

➤ Iskanje elementov HTML

- `document.getElementById(id)` – najde element z element id
- `document.getElementsByTagName(name)` – najde elemente s tag name
- `document.getElementsByClassName(name)` – najde elemente s *class name*

○ Spreminjanje elementov HTML

- `element.innerHTML` = nova vsebina – spremeni notranji HTML elementa
- `element.attribute` = nova vrednost – spremeni vrednost atributa HTML elementa
- `element.setAttribute(attribute, value)` – spremeni vrednost atributa HTML elementa
- `element.style.property` = *nov stil* – spremeni stil HTML elementa

○ Dodajanje in brisanje elementov HTML

- `document.createElement(element)` – ustvari element HTML
- `document.appendChild(element)` – doda element HTML
- `document.removeChild(element)` – odstrani element HTML
- `document.replaceChild(element)` – zamenja element HTML
- `document.write(text)` – piše tekst v HTML izhodni tok

► Iskanje HTML objektov

- `document.anchors` – vrne vse elemente `<a>`, ki imajo ime atributa
- `document.cookie`
- `document.doctype`
- `document.domain`
- `document.forms`
- `document.images`
- `document.URL`
- ...

HTML Tutorial

CSS Tutorial

XML Tutorial

Klikni - število `anchors` v dokumentu.

Preveri

Primer: `x=document.anchors.length;` (3 vrstice določene z `<a>`)

JS HTML DOM - dogodki

▶ Dogodki z miško - lastnosti:

- onclick - uporabnik je z miško kliknil na objekt
- ondblclick (ondoubleclick): - dvojni klik na objekt
- onmouseover - prelet miške, ko se zapelje miško nad objekt
- onmouseout – izstop, ko se izstopi z miško iz objekta

▶ Dogodki s tipkovnico - lastnosti:

- onkeydown/onkeyup – ob pristiku na tipko/sprostitvi tipke
- onkeypress - ob držanju tipke

▶ Dogodki z obrazci:

- onchange – ob spremembi vsebine (<input>, <select>, <textarea>)
- onreset - ko je obrazec obnovljen v začetno stanje
- onsubmit – ko je obrazec poslan

-
- ▶ JavaScript se izvede šele ob dogodku (klik elementa HTML).
 - ▶ Primeri dogodkov:
 - Ko uporabnik klikne z miško
 - Ko se miška giblje nad elementom
 - Ko se naloži spletna stran
 - Ko se naloži slika
 - Ko se spremeni vhodno polje
 - Ko je oddan obrazec HTML
 - Ko uporabnik pritisne tipko
 - ▶ Zamenjava vsebine elementa `<h1>`:
`<h1 onclick="this.innerHTML='Ooops!' ">Klikni na tekst!</h1>`

-
- ▶ Zamenjava vsebine elementa `<h1>` – klic funkcije

```
<script>
    function zamenjajtekst(id) {
        id.innerHTML="OHHH!";
    }
</script>
<body>
    <h1 onclick="zamenjajtekst(this)">Klikni na tekst!</h1>
</body>
```

- ▶ Uporaba atributov – klik na gumb (element `<button>`)
`<button onclick="displayDate()">Datum</button>`
- ▶ Dogodek ob spremembi vnosnega polja
`<input type="text" id="fname" onchange="upperCase()">`

JS HTML DOM – sprememba CSS

► Sintaksa

`document.getElementById(id).style.property=new style`

Primer: `<p id="p2"> ... </p>`

`document.getElementById("p2").style.color = "blue";`

○

► Uporaba dogodkov:

`<h1 id="id1">Naslov1 </h1>`

`<button type="button"`

`onclick="document.getElementById('id1').style.color = 'red'">`

`Klikni!</button>`

jQuery - uvod

- ▶ JavaScript knjižnica
- ▶ Poenostavljeno programiranje z JavaScript
- ▶ Kaj moramo poznati: HTML, CSS, JavaScript

- ▶ Omogoča:
 - Upravljanje HTML/DOM
 - Upravlja CSS
 - HTML metode dogodkov
 - Efekte in animacije

- ▶ Zakaj uporabiti jQuery?
 - ▶ Najbolj popularen
 - ▶ Najbolj razširljiv
 - ▶ Uporabljajo ga velika podjetja: Google, Microsoft, ...



jQuery is a JavaScript Library.

jQuery greatly simplifies JavaScript programming.

jQuery is easy to learn.

jQuery - namestitev

Knjižnica je nameščena je v elementu <head>

1. Lokalna namestitev knjižnice v mapi (<http://jquery.com>)

```
<head>
  <script src="jquery-1.12.0.min.js">
  </script>
</head>
```

2. Direktna vključitev v dokument (Google, Microsoft):

```
<head>
  <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js">
  </script>
</had>
<head>
  <script src="http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.12.0.min.js">
  </script>
</head>
```

jQuery - sintaksa

- ▶ Izbira elementa HTML in izvedba akcije nad njim
- ▶ Sintaksa: **`$(selector).action()`**
 - `$` definicija/dostop do knjižnice jQuery
 - `(selector)` dostop ali iskanje elementov HTML
 - `action()` določa nalogo, ki se izvede na elementih HTML
- ▶ Metode jQuery so znotraj dogodka :

```
$(document).ready(function(){  
    // metode jQuery ...  
});
```

ali krajše

```
$(function(){  
    // metode jQuery ...  
});
```
- ▶ Izvajanje kode se prične šele, ko je dokument naložen.

jQuery – selektorji

- ▶ Selektor je element
 - `$("p")` – vsi elementi `<p>`.
- ▶ Selektor je `.class`
 - `$(".test")` – vsi elementi s specifičnim `class="test"`
- ▶ Selektor je `#id`
 - `$("#test")` - element s specifičnim `id="test"`
- ▶ Drugi selektorji:
 - `$("*")` – vsi elementi
 - `$(this)` – trenutni element HTML
 - `$("p:first")` – prvi element `<p>`
 - `$("ul li:first")` – prvi element `` od prvega elementa ``
 - `$("[href]")` – vsi elementi z atributom `href`
 - `$("tr:odd")` – vsi lihi elementi `<tr>`

jQuery - metode dogodkov

► Večina dogodkov DOM ima ekvivalentno metodo jQuery

- Premik miške nad elementom
- Izbira gumba 'radio'
- Klik na element

► Dogodki DOM:

- Miška: click, dblclick, mouseenter, mouseleave
- Tipkovnica: keypress, keydown, keyup
- Obrazec: submit, change, focus, blur
- Dokument: load, resize, scroll, unload

► Sintaksa: `$("p").click();`

► določimo kaj naj se zgodi – posredujemo funkcijo

```
$("p").click(function(){  
    // akcija !!  
});
```

jQuery - efekti

- ▶ `hide()` - skrij
- ▶ `show()` - prikaži
- ▶ `toggle()` - izmenoma izvede `hide()` in `show()`
- ▶ `fadeIn` – slabiti, izginevati
 - ▶ `fadeIn()`,
 - ▶ `fadeOut()`,
 - ▶ `fadeToggle()`,
 - ▶ `fadeTo()`
- ▶ `sliding` - drseti
 - ▶ `slideDown()`,
 - ▶ `slideUp()`,
 - ▶ `slideToggle()`

▶ **animate():**

`$(selector).animate({params},speed,callback);`

- **params** lastnost CSS za animacijo
- **speed** “slow“, “fast“
- **callback** funkcija, ki se izvede ob zaključku animacije

▶ **stop()** – zaustavi animacijo predno se konča

`$(selector).stop(stopAll,goToEnd);`

▶ **callback**

`$(selector).hide(speed,callback);`

▶ **chaining** - združevanje akcij/metod na istem elementu

`$("#p1").css("color","red").slideUp(2000).slideDown(2000);`

jQuery- HTML

- ▶ jQuery upravljanje elementov in atributov
- ▶ Pridobi/postavi vsebino in vrednost
 - ▶ `text()` – tekstovna vsebina izbranega elementa
 - ▶ `html()` – vsebina elementa
 - ▶ `val()` – vrednost polj v obrazcu
- ▶ Pridobi/postavi vrednost atributa
 - ▶ `attr()`
- ▶ Dodaj novo vsebino HTML
 - ▶ `append()` – vstavi vsebino na konec izbranih elementov
 - ▶ `prepend()` - vstavi vsebino na začetek izbranih elementov
 - ▶ `after()` - vstavi vsebino za izbrane elemente
 - ▶ `before()` - vstavi vsebino pred izbrane elemente
- ▶ Odstrani elemente
 - ▶ `remove()` – odstrani izbrani element (in njegove elemente 'child')
 - ▶ `empty()` - odstrani elemente 'child' od izbranega elementa

jQuery- metoda css()

► Vrne lastnost CSS

```
css("propertyname","value");
```

Primer:

```
$("p").css("background-color");
```

► Postavi lastnost CSS

```
css("propertyname","value");
```

Primer.

```
$("p").css("background-color", "yellow");
```

► Postavi več lastnosti CSS

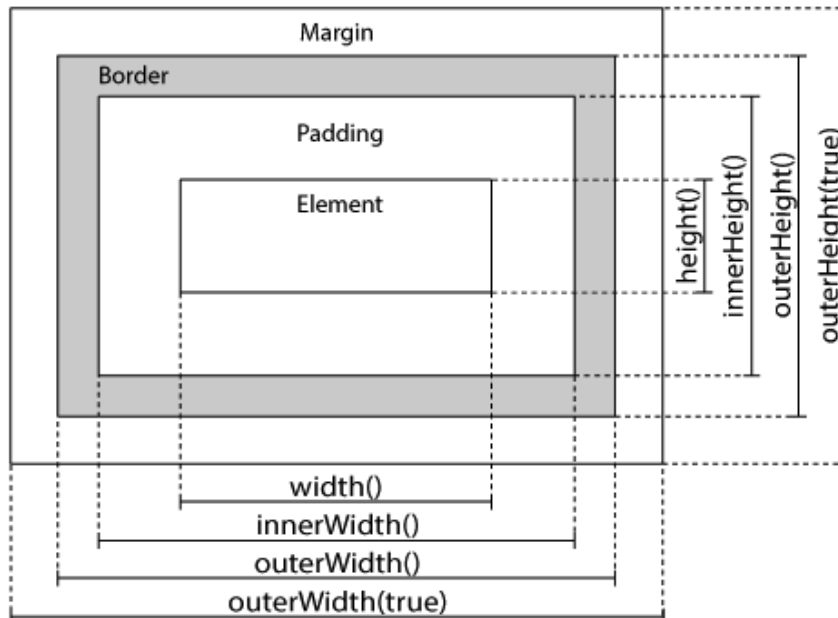
```
css({"propertyname":"value","propertyname":"value",...});
```

Primer:

```
$("p").css({"background-color": "yellow", "font-size": "200%"});
```

jQuery- metode dimenzij

- ▶ width()
- ▶ height()
- ▶ innerWidth()
- ▶ innerHeight()
- ▶ outerWidth()
- ▶ outerHeight()



Primer:

```
<div id="dim"></div>
```

```
$("#dim").width()
```

```
$("#dim").height()
```

...

```
Width of div: 299.8  
Height of div: 121.8  
Inner width of div: 319.8  
Inner height of div: 141.8  
Outer width of div: 323  
Outer height of div: 145
```

Display dimensions of div

Naloga 1: Deljitelji števil

Največji skupni deljitelj dveh števil:

Primer: $x=65, y=15$, Rešitev: 5

- Vnos števila x in y
- Odštevanje manjšega števila od večjega števila, dokler sta števili različni.

Deljitelji števila n .

Primer: $n=20$ - izpis: 1 2 4 5 10

- število n delimo z $i=1, 2, 3, \dots, n-1$,
- če je ostanek $=0$, potem je i deljitelj števila n

Naloga 2: Poštevanka - izpis

```
var i, j;  
for (j = 1; j <= 5; j++) {  
    for (i = 1; i <= 5; i++)  
        document.write(" " + (i * j));  
    document.write("<br />");  
}
```

1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25

Primeri:

→ i

j ↓

1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25

1				
2	4			
3	6	9		
4	8	12	16	
5	10	15	20	25

6
12
18
24
30

Naloga 3: Vodilne ničle v izpisu. Rezultat=?

1. Dodajanje vodilnih ničel – število x izpišite na N mest

R: x=32, N=7: IZPIS: 0000032

2. Kaj se izpiše? (Lokalne in globalne spremenljivke)

```
<script>
function test(x) {
    var y = 2;
    x++;
    y++;
    z++;
}
var x = 0, y = 0; z = 0;
test(x);
document.write("x= " + x + " y= " + y + " z= " + z);
</script>
```

Naloga 4: Izpis datuma

Izpis: `Danes je Sreda,16.3.2016`

Rešitev:

- ▶ Definiramo polje z imeni dni v tednu:

```
dnevi=new Array["Nedelja","Ponedeljek",...]
```

- ▶ Uporabimo objektni tip `Date()`;

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

```
danes.getDay() – vrne dan v tednu (0-6)
```

```
dnevi[danes.getDay()] – zapis dneva v tednu
```

- ▶ Izpis:

```
document.write("Danes je " + dnevi[danes.getDay()] + ", "  
+ danes.getDate() + ". " + (danes.getMonth()+1) + ". " +  
danes.getFullYear() + ".");
```

Nedelja

Ponedeljek

Torek

Sreda

Četrtek

Petek

Sobota

Naloga 5: Spreminjanje elementov HTML DOM

► Spreminjanje vsebine elementa HTML

```
<p id="p0">Lep pozdrav!</p>
```

```
<script>
```

```
    document.getElementById("p0").innerHTML="Nasvidenje!";
```

```
</script>
```

► Spreminjanje atributa v elementu HTML

```

```

```
<script>
```

```
    document.getElementById("image").src="landscape.jpg";
```

```
</script>
```

► Spreminjanje stila v elementu HTML

```
<p id="p1">Lep pozdrav!</p>
```

```
<script>
```

```
    document.getElementById("p1").style.color="red";
```

```
</script>
```

Naloga 6: JS HTML DOM event

```
<html><head>
```

```
<script>
```

```
function spremeni()  
{
```

V funkciji spremeni() spremenljivki d priredimo **sklic na obstoječi** objekt z lastnostjo `id= "odstavek "`

```
    var d = document.getElementById("odstavek");  
    d.innerHTML="hahaha!!!";
```

Ustvarjeni objekt ima lastnost, ki predstavlja vsebino elementa `p`

```
}
```

```
</script></head>
```

```
<body>
```

```
    <p id="odstavek" onclick="spremeni()">Nikar ne klikni tega besedila!</p>
```

```
</body></html>
```

V elementu `p` se kliče funkcija spremeni() z uporabo dogodka z miško (onclick)

► Pred klikom miške

Nikar ne klikni tega besedila!

► Po kliku miške

hahaha!!!

Naloga 7: JS HTML DOM Document

```
<!DOCTYPE html>    <script>
<html>              x=document.getElementById("main").getElementsByTagName("p");
<head>              document.write("Prvi odstavek v div: " + x[0].innerHTML);
<style>              y=document.getElementsByTagName("center");
p.center {           document.write("<br />Class: " + y[0].innerHTML);
text-align:center;   </script>
}                   </body></html>
</style>
<body>
<p>Pozdrav!</p>
<p class="center">Naslov I</p>

<div id="main">
<p>DOM metode in lastnost</p>
<p>Primeri:</p>
</div>
```

Pozdrav!

Naslov 1

DOM metode in lastnost

Primeri:

Prvi odstavek v div: DOM metode in lastnost
Class: Naslov 1

Naloga 8: Dodajanje elementov HTML DOM

- Primer: `document.getElementById().appendChild();`

```
<ul id="myList1"><li>Kava</li><li>Mleko</li></ul>
```

```
<ul id="myList2"><li>Voda</li><li>Sok</li></ul>
```

```
<p id="demo">Klikni za premik iz prvega v drug seznam.</p>
```

```
<button onclick="myFunction()">Klikni!</button>
```

```
<script>
```

```
function myFunction()
```

```
{
```

```
var node=document.getElementById("myList2").lastChild;
```

```
document.getElementById("myList1").appendChild(node);
```

```
}
```

```
</script>
```

- Kava
- Mleko
- Voda
- Sok

Klikni za premik iz prvega v drug seznam.

Klikni!

Naloga 9: jQuery - efekti

```
<head>
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js">
</script>
<script>
    $(document).ready(function(){
        $("tr:even").css("background-color","yellow");
    });
</script></head>
<body>
    <h1>Dobrodošli</h1>
    <table border="1">
        <tr>... </tr> ... <!-- tabela in podatki -->
    </table>
</body>
```

Dobrodošli

Ime	Priimek
Janez	Novak
Peter	Klepec
Jože	Petek
Franc	Jožef

Naloga 10: jQuery - dogodki

```
<!DOCTYPE html>
<html><head>
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js">
</script>
<script>
    $(document).ready(function(){
        $("p").click(function(){
            $(this).hide();
        });
    });
</script></head>
<body>
    <p>Klikni me - ?</p>
    <p>Klikni me!</p>
    <p>Še enkrat klikni!</p>
</body>
</html>
```

Rešitev: Klik na tekst – 3 krat

Klikni me - ?

Klikni me!

Še enkrat klikni!

Klikni me!

Še enkrat klikni!

Še enkrat klikni!

Naloga 11: Metode dogodkov jQuery

► Določitev akcije, ki se naj zgodi:

```
$(document).ready(function(){ - Skrije trenutni element <p> (zapis je v <head>
    $("p").click(function(){
        $(this).hide();
    });
});
```

<p> Skrij me!</p>

- Zapis je v elementu <body>

```
$(document).ready(function(){
    $("#p1").mouseenter(function(){
        alert("vstopili ste v p1!");
    });
});
```

<p id="p1">Vstopi v odstavek.</p>

- Zapis je v elementu <body>



Naloga 12: HTML, CSS, JavaScript, jQuery

► Primer 1: JS

JavaScript
Primer 1

Datum

Briši tekst!

Izpis

Vnesi ime.
Brisanje teksta

FR!
Vpiši:

Dodaj

SPLETNE TEHNOLOGIJE 2013/2014

► Primer 2: jquery

JavaScript, jQuery
Primer 1

Zamenjaj

Briši

Predavanja

Pozdravljeni.
To je še eno poglavje.

FR!
Vpiši:

Dodaj

SPLETNE TEHNOLOGIJE 2013/2014

Naloga 13: Ura

Ura - risanje delujoče ure

- ▶ HTML5 element: `<canvas>`
- ▶ JavaScript

