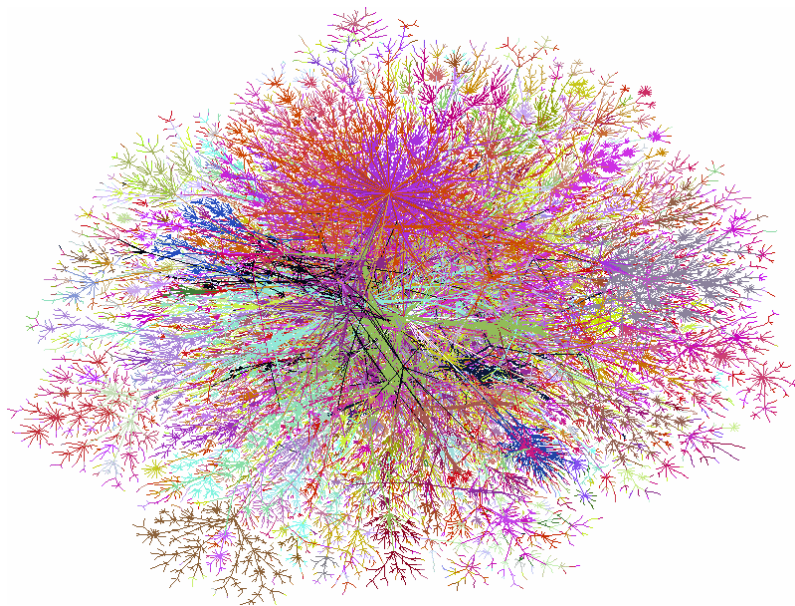




# HTML - Hypertekst Markup Language

WWW (Word Wide Web) – svetovni splet



Različne barve določene na osnovi IP naslova  
<http://research.lumeta.com/ches/map>

## Kaj sestavlja internet

- WWW
- Video konference
- ftp
- telnet
- Elektronska pošta
- klepetalnice
- ...



  
The World's Online Marketplace®





Komunikacijska infrastruktura  
Uporabnost je v izmenjavi informacij

“On-line interactive communities... will be communities not of common location, but *of common interest*.... the total number of users...will be large enough to support extensive general purpose [computers]. All of these will be interconnected by telecommunications channels... [to] constitute a labile network of networks--ever changing in both content and configuration.”

J. C. R. Licklider



## Razvoj infrastrukture

- Po 1960 - DARPA (ARPA v 1960) projekt, ki ga je vodil Licklider
- Konec šestdesetih let - ARPANET & raziskave na področju paketnega preklapljanja - Roberts
  - Prvo vozlišče nameščeno v BBN pri UCLA v septembru 1969
  - 1969 – štirje gostoteljski računalniki povezani (UCLA, SRI, UCSB, Univerza v Utahu)



1972 – E-pošta - Ray Tomlinson & Larry Roberts

Od 1970 - TCP - Vint Cerf & Bob Kahn – razvoj TCP/IP in UDP

Od 1980 – eksplozija razvoja strojne opreme (LAN, PC, in delovnih postaj)

1983 – Ethernet - Metcalfe

DNS – porazdeljen in razširljiv mehanizem za razčlenjevanje gostiteljskih imen v IP naslove

UC Berkeley vgradi TCP/IP v Unix BSD

1985 – Internet uporabljajo raziskovalci in razvijalci

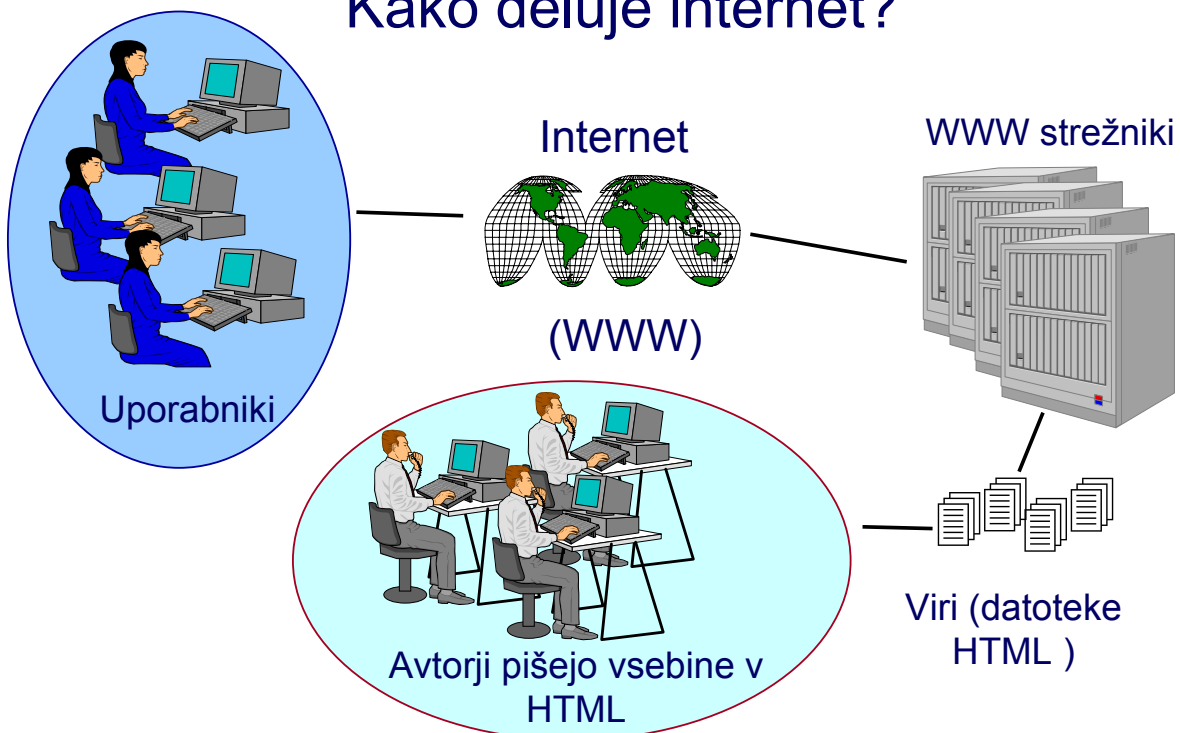
Tim Berners-Lee v CERNu 1989

Predlog za WWW v 1990

Prva spletna stran 13. novembra 1990

Hypertext – Besedilo, ki vsebuje povezave do drugih besedil.

## Kako deluje internet?





**Protokol:** predpis, ki ga uporabljajo vsi računalniški programi pri medsebojni komunikaciji

- HTTP: Hypertext Transfer Protocol – množica pravil, ki dovoljujejo brskalnikom, da komunicirajo z internetnimi (web) strežniki
- IP = Internet Protocol
- FTP = File Transfer Protocol
- SMTP = Simple Mail Transfer Protocol
- TCP = Transmission Control Protocol



## World Wide Web

- WWW je storitev interneta
- WWW povezuje informacijo, tako da lahko uporabniki enostavno prehajajo iz enega dokumenta na drugega, ki je običajno z njim vsebinsko povezan.
- WWW je zasnovan na HTML



## Brskalnik

- Je računalniški program, ki ga uporabljamo za pregledovanje internetnih strani.
- Brskalnik vključuje naslednje funkcionalnosti:
- Iskanje in dostop do web strežnika;
- Nalaganje, urejanje in prikaz HTML dokumentov;
- Detekcija hiperpovezav in prehajanje iz enega dokumenta v drugega;
- Podpora standardnih orodij.



- **WWW = World Wide Web – mreža računalnikov, ki med seboj komunicirajo**
  - porazdeljen
  - Neodvisen od platforme
  - interaktiven & realni čas
  - tekst, grafika, zvok, video
    - Mosaic 1993 - Natl. Ctr. Supercomputing Appls. (NCSA)
    - Marc Andreessen founded Netscape

## Akronim: HTTP

- Hypertext Transfer Protocol
- Definira postopek po katerem nek računalnik (brskalnik) poda zahtevo drugemu oddaljenemu računalniku (web strežniku) za HTML dokument.

## Kaj je HTML?

- HTML = HyperText Markup Language
- Markup (opis) – je postopek **dodajanja informacije** besedilu s katero označimo **logične komponente dokumenta, inštrukcije glede izgleda**, ali drugo informacijo, ki jo lahko nek avtomatski sistem ustrezno interpretira.
- HTML dokumente lahko berejo in interpretirajo internetni brskalniki
- HTML ni programski jezik



## Akronim: HTML

- **H**ypertext **M**arkup **L**anguage
    - hypertext : množica dokumentov s povezavami
    - markup (opis): k tekstu dodane značke za njegov urejen izpis
    - language : slovar, slovnica itd.
  - Jezik za opis strukture in izgleda spletne strani
  - Brskalniki berejo HTML in uporabljajo značke za določitev načina prikaza teksta
  - HTML dokumenti so zapisani v tekstovnih datotekah
- 



## HTML

- HTML 2.0 - November, 1995.
- HTML 3.0 - September 28, 1995
- HTML 3.2 - January 14, 1997
- HTML 4.0 - July 8, 1997
- W3C – World Wide Web Consortium
- <http://www.w3.org>

### Nekateri elektronsko dostopni viri za HTML

HTML:

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

Standard HTML 4.01:

<http://www.w3.org/TR/REC-html40/>

---





## Priprava www strani

Da bi lahko pripravili www stran potrebujemo:

- **Urejevalnik za pisanje HTML dokumentov.**
  - Notepad, WordPad, VI, ali drugi.
- **HTTP brskalnik** za preskušanje strani
  - Trenutne različice Internet Explorerja, FireFox, ali Netscape
- **Web strežnik** na katerega shranimo datoteke in omogočimo njihovo pregledovanje.



### Vsesplošna dosegljivost internetnih dokumentov (v nasprotju z drugimi)

- Web strani so narejene z namenom njihovega doseganja od koderkoli, ne glede na tip strojne opreme ali programske opreme klienta, s katerega dosegamo do strani.
- Načrtovalec internetnih strani nima kontrole nad karakteristikami računalnika, na katerem bo dokument prikazan.
- Formatiranje internetnih strani je zato izvedeno na način, da ostajata vsebina in logični dizajn strani enaka, čeprav je morda fizični izgled strani različen.
- Izvorno kodo dokumentov si lahko ogledate v vsakem brskalniku – tekst z dodanimi značkami.

## Datoteke HTML

- HTML dokumenti morajo biti shranjeni kot tekstovne datoteke s končnico html ali htm. Končnica pove brskalniku, da mora datoteko interpretirati skladno s standardom HTML.
- datoteke HTML so po obsegu relativno majhne.

## Kako deluje HTML?

- HTML sestavlja množica “ukazov”, ki jih imenujemo značke. Te povedo brskalniku kako naj prikaže informacijo na spletni strani.
- **Značke** so zaprte v **trikotnih oklepajih** <...>
- Značke so dodane znotraj besedila.
- Značke so **neodvisne od načina zapisa** (velike/male črke), smiselna je konsistentna raba (samo velike ali samo male črke).

## HTML

- Večina značk se pojavlja v parih, ena pred tekstem na katerega obliko izpisa vpliva in ena takoj za koncem tega teksta.
- Prva vrstica html datoteke, ki bo prikazana kot spletna stran je `<HTML>`; zadnja vrstica datoteke je `</HTML>`
- Spletna stran je običajno razdeljena v dva dela - glava (HEAD) in telo (BODY). Glava je označena z značkama `<HEAD>` in `</HEAD>`, telo pa z `<BODY>` in `</BODY>`.

## Splošna oblika HTML dokumenta



## Primer preproste strani

```
<html>

  <!-- Tu lahko pišemo poljuben komentar -->
  <!-- Brskalnik komentarje ignorira in jih ne prikaže na spletni strani -->

<head>

  <title> Tukaj je naslov strani </title>

</head>

<body>
  Tu pišemo tekst ki se bo pojavil v oknu brskalnika.
  Tu pišemo tekst ki se bo pojavil v oknu brskalnika.
</body>

</html>
```

## Pravila HTML

- Neobčutljiv na velikost črk
- Presledki:
  - Brskalnik ignorira odvečne presledke
- Značke nivoja bloka vključujejo avtomatsko deljenje vrstic
  - Primeri: P, H1, UL, TABLE

## Atributi (lastnosti) značk

- Atributi ponujajo množico možnosti
- Attribute zapišemo med ukazom in končnim simbolom >
- Posamezna značka lahko ima več atributov
- Attribute lahko zapišemo enega za drugim v poljubnem vrstnem redu

```
<IMG SRC="Image.bmp" HSPACE=5>
```

## Vrednosti atributov

- Atributom lahko pridružimo vrednosti posameznega tipa.
- Vrednosti atributov morajo biti zapisane med narekovaje.
- "" lahko izpustimo, če vrednost vključuje samo črke, številke, vezaj (-), ali piko (.)

## Gnezdenje značk

- Hierarhija – značke, ki vplivajo na celoten odstavek lahko vključujejo značke, ki vplivajo na posamezne besede ali črke
- Vrstni red – vrstni red značk se mora ujemati glede na njihovo hierarhijo in mesto postavitve.

## Gnezdenje značk (primer)

### Pravilno:

```
<H1> Information  
<I>System</I></H1>
```

### Nepravilno:

```
<H1> Information  
<I>System</H1></I>
```

## Osnovni tipi značk

- **Značke dokumenta:** določajo različne dele dokumenta (Head, Body)
- **Značke strukture besedila:** določajo izgled prikazanega besedila
- **Značke oblike:** povedo brskalniku kako naj prikaže besedilo
- **Značke slike:** omogočajo vključevanje slik
- **Značke sidranja:** za definiranje hiperpovezav

## Značka dokumenta

- **<!DOCTYPE>** nima končne značke; pove brskalniku kakšno obliko zapisa naj ta pričakuje; html/xhtml, strict, transitional ali frames.
  - Tri definicije tipa dokumentov (Document Type Definitions (DTD) - definira pričakovan tip podatkov/značk) za html:
    - `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">`
    - `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">`
    - `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">`
  - Če značka `<!DOCTYPE>` ni definirana, brskalnik predpostavi html.



## Osnovne značke HTML

<code>&lt;H1&gt; ... &lt;/H1&gt;</code>	Naslov
<code>&lt;H2&gt; ... &lt;/H2&gt;</code>	Podnaslov.
<code>&lt;H3&gt; ... &lt;/H3&gt;</code>	Podnaslov.
<code>&lt;BLOCKQUOTE&gt; ... &lt;/BLOCKQUOTE &gt;</code>	Zamik bloka besedila za razdaljo enega tabulatorja.



## Osnovne značke HTML

<code>&lt;P&gt; ... &lt;/P&gt;</code>	: Paragraf.
<code>&lt;B&gt; ... &lt;/B&gt;</code>	: Krepko
<code>&lt;I&gt; ... &lt;/I&gt;</code>	: Ležeče
<code>&lt;U&gt; ... &lt;/U&gt;</code>	: Podčrtano
<code>&lt;EM&gt; ... &lt;/EM&gt;</code>	: Poudarjeno (način izpisa odvisen od brskalnika)
<code>&lt;BR/&gt;</code>	: Nova vrstica
<code>&lt;HR/&gt;</code>	: Vodoravna črta





## Tabele

```
<table width="300" border="1">
  <tr>
    <th align="left">Vrsta gobe </th>
    <th align="center">Klobuk</th>
    <th align="center">Bet</th>
    <th align="center">Užitnost</th>
  </tr>
  <tr>
    <td align="left">Mušnica</td>
    <td align="right">Rdeč</td>
    <td align="center">Bel</td>
    <td align="center">Neužitna</td>
  </tr>
  <tr>
    <td align="left">Jurček</td>
    <td align="center">Rjav</td>
    <td align="center">Bel</td>
    <td align="center">Užitna</td>
  </tr>
</table>
```

• Opis elementa je v splošnem sestavljen iz treh delov:

- iz začetne značke, vsebine in končne značke.
- Začetna in končna značka se pričneta z znakom `<`, končata z znakom `>`, vmes pa je zapisano ime elementa.
- V končni znački je pred imenom vedno znak `/`.
- V začetni znački lahko za imenom določimo attribute elementa. Za vsak atribut napišemo presledek, njegovo ime, znak `=` in vrednost atributa.
- Vrednosti atributa je priporočljivo pisati v dvojnih narekovajih.



## Pogoste napake pri pisanju HTML dokumentov

- Značke nimajo zaključka (npr. napišemo `<ul>`, ne pa tudi `</ul>`).
- Značke niso ustrezno ugnezdene.
- Napačno poimenovanje značk (`<d>` namesto `<b>` - brskalnik nedefinirane značke ignorira).
- Vrednosti atributov niso v dvojnih narekovajih ali manjka začetni ali končni dvojni narekovaj.
- Komentarji nimajo zaključka - vsebina se ne izpiše.

Nekateri brskalniki veliko napak ignorirajo in lahko posledično prikažejo neustrezno oblikovano slovnično nepravilno vsebino.



## Struktura dokumentov HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>

  <!-- Glava -->
  <head>
    <title> Naslov dokumenta </title>
    <meta http-equiv="Content-Type" content="text/html";
charset="utf-8" />
  </head>

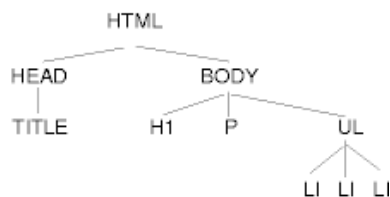
  <!-- Telo -->
  <body>
    Telo HTML dokumenta.
  </body>
</html>
```

35



## Hierarhija označb HTML

```
<html>
<head>
<title>Zgodba o gobah</title>
</head>
<body>
<h1>Zgodba o gobah</h1>
<p> V gozdu raste veliko gob.
  Da bomo dočakali naslednji dan je
  dobro zaužiti le naslednje: </p>
```



```
<ul>
<li> Goban
<li> Lisička
<li> Jurček
<li> Sivka
<li> Medvedove tačke
</ul>
</body>
</html>
```

- *html* je starš od *head* in *body*
- *head* in *body* sta otroka od *html*
- *p* je predhodnik od *ul*
- *ul* je naslednik od *p*
- vsi *li* so sorodniki
- *li* je potomec od *body*
- *body* je prednik od *li*

36



## Glava dokumenta HTML

- V glavi opišemo dokument.
  - Značka <title> določa naslov, ki ga brskalnik prikaže v naslovni vrstici.
  - Dodatne značke določajo način kodiranja (na kakšen način so zapisani znaki, ki jih ni v tabeli ASCII), ključne besede, ki iskalnikom omogočajo uvrščanje spletne strani v ustrezne skupine, obliko posameznih elementov, vključevanje pomožne datoteke ...
- Zapis v glavi v oknu brskalnika ni viden.
  - Vidnost zapisa omogoči posebna zahteva (npr. ukaz Tools, Page Info v brskalniku Mozilla Firefox).



## Označbe glave

<u>&lt;head&gt;</u>	Glava HTML dokumenta
<u>&lt;title&gt;</u>	Naslov dokumenta
<u>&lt;meta&gt;</u>	Meta podatki
<u>&lt;base&gt;</u>	Definira začetni URL sklicevanih dokumentov
<u>&lt;basefont&gt;</u>	Definira osnovni tip, barvo in velikost pisave dokumenta
<u>&lt;script&gt;</u>	Del dokumenta, ki vsebuje skript (npr. Javascript)
<u>&lt;noscript&gt;</u>	Del v katerem v katerem ni skript
<u>&lt;style&gt;</u>	Definira stil CSS

## Glava dokumenta HTML

Primer:

```
<title> Gobarski vodič </title>
<meta http-equiv="Content-Type" content="text/html"; charset="windows-1250" />
<meta name="Author" content="Janez Preživeli" />
<meta name="Keywords" content="Gobe, vodič" />
<meta name="Description" content="Gobarski vodič za preživele" />
<meta name="Generator" content="WordPad" />
<meta http-equiv="Refresh" content="15;url=http://www.dsplab.uni-mb.si">
<meta http-equiv="Robots" content="noindex" />
<meta http-equiv="Content-Language" content="en" />
<base href="http://www.feri.uni-mb.si/csferi/" />
<style type="text/css"> p {color: #0000FF;} </style>
<link rel="StyleSheet" type="text/css" href="oblika.css" />
<script type="text/javascript" src="prikaz.js"></script>
```

## Glava dokumenta HTML

- Oznaka `<title>`: določa naslov dokumenta HTML, ki se izpiše v naslovni vrstici.  
`<title> Gobarski vodič </title>`
- Oznaka `<meta>` - omogoča dodajanje informacij o sami spletni strani. Na izbiro več možnosti:
  - `<meta name="Description" content="Gobarski vodič za preživele" />` - Opis spletne strani
  - `<meta name="Keywords" content="gobe, vodič" />` - Ključne besede, pomembne pri indeksiranju - iskalniki uvrščajo dokumente s ključnimi besedami višje
  - `<meta name="Author" content="Janez Preživeli" />` - Avtor dokumenta



## Glava dokumenta HTML

- `<meta http-equiv="Content-Type" content="text/html"; charset="windows-1250" />` - Definiranje dokumenta tipa [MIME](#) (Multipurpose Internet Mail Extensions) (text/HTML) in kodiranja (windows-1250, UTF-8...)
- `<meta http-equiv="Content-Language" content="en" />` - Jezik dokumenta
- `<meta http-equiv="Refresh" content="15;url=http://www.dsplab.uni-mb.si">` - čas preusmeritve na definiran url naslov. Po preteku definiranega časa (15 s) se izvede preusmeritev na url naslov <http://www.dsplab.uni-mb.si>
- `<meta http-equiv="Robots" content="noindex" />` - Uporaba oznake Robots (noindex) prepreči avtomatsko indeksiranje strani, ki ga izvajajo različni iskalniki.



## Glava dokumenta HTML

- `<base href="http://www.dsplab.uni-mb.si/home/" />` Oznaka `<base>` določa osnovni url naslov za reference, ki imajo relativne naslove glede na ta naslov
- `<style type="text/css"> p {color: #0000FF;} </style>` - oznaka `<style>` določa stil CSS v glavi dokumenta
- `<link rel="StyleSheet" type="text/css" href="oblika.css" />` - oznaka `<link>` določa povezavo do zunanje datoteke (v tem primeru vključimo datoteko oblika.css s stilom CSS), ki se vključi v glavo dokumenta
- `<script type="text/javascript" src="prikaz.js"></script>` - z oznako `<script>` vključimo datoteko skript v glavo dokumenta (v tem primeru vključimo datoteko Javascript [prikaz.js](#))

## Telo dokumenta HTML

- Telo dokumenta vključuje vsebino, ki se prikaže v brskalnikovem oknu.

Najpogosteje jo sestavljajo: besedilo, slike, tabele, sezname, povezave in drugi elementi.

- **Primer:**

`<body>`

`<p>` Besedilo, zapisano v navadnem odstavku. Brskalnik ignorira več zaporednih presledkov v vrstici. `</p>`

`<p>` Drugi odstavek besedila. Brskalnik sam določa velikost presledka med dvema odstavkoma.

- `<br/>` Skok v novo vrstico izveden, ko zmanjka prostora v tekoči vrstici. Skok v novo vrstico drugače izveden le, če to posebej zahtevamo z značko za skok v novo vrstico. `</p>`

`<pre>` Uporabimo, ko so razlomi vrstic in presledki med besedami v besedilu pomembni. `</pre>`

`</body>`

## Posebni znaki

- Znaki v HTML, ki imajo poseben pomen: `<`, `>`, `"`, `&` ...
  - Zapis nekaterih posebnih znakov:

- `<` zapišemo z `&lt;`;

- `>` zapišemo z `&gt;`;

- `®` zapišemo z `&reg;`;

- `&` zapišemo z `&amp;`;

- `"` zapišemo z `&quot;`;

- .....

**Primer:** `x < y` zapišemo kot: `x &lt; y`.

- Imena znakov so določena s standardom SGML. Seznam posebnih znakov in ASCII znakov dosegljiva na:

- [http://www.w3schools.com/tags/ref\\_entities.asp](http://www.w3schools.com/tags/ref_entities.asp)

- [http://www.w3schools.com/tags/ref\\_ascii.asp](http://www.w3schools.com/tags/ref_ascii.asp)



## Oblikovanje besedila

- Pri oblikovanju besedila sledimo paradigmi: “to kar vidiš je to kar si mislil” (what you see is what you meant – WYSIWYM).
- Pri oblikovanju besedila z uporabo značk določimo, kako naj bo besedilo prikazano. Za to lahko uporabimo različne elemente, ki bodo vplivali na obliko črke, besede, stavka ali daljšega zapisa.
- **Uporaba različnih elementov in način izpisa besedila:**

```
<b> Besedilo bo izpisano krepko </b>
<i> Besedilo bo izpisano nagnjeno </i>
<tt> Izgled besedila natipkanega na pisalnem stroju </tt>
<big> Besedilo bo izpisano z večjimi znaki </big>
<big><big> Besedilo bo izpisano s še večjimi znaki </big></big>
<big><big><big> Besedilo bo izpisano s še večjimi znaki </big></big></big>
<small> Besedilo bo izpisano z manjšimi znaki </small>
<small><small> Besedilo bo izpisano s še manjšimi znaki </small></small>
<small><small><small> Besedilo bo izpisano s še manjšimi znaki
</small></small></small>
```



## Oblikovanje besedila

Besedila ponavadi oblikujemo hierarhično: razdelimo jih v poglavja, podpoglavja, podpodpoglavja ... Z uporabo ustreznih elementov značk določimo hierarhijo besedila. Brskalnik sam določi končni izgled besedila na zaslonu.

- Bločni elementi za hierarhično oblikovanje besedila so različne vrste naslovov (elementi [h1](#), [h2](#), [h3](#), [h4](#), [h5](#) in [h6](#)), odstavek (element [p](#)) in besedilo s predpisanimi prelomi vrstic (element [pre](#)).
- Vrstični elementi so na primer indeks (element [sub](#)), eksponent (element [sup](#)) in drugi, ki jih običajno redkeje uporabljamo (na primer elementi [cite](#) [address](#), [em](#), [code](#),...).

### Primeri uporabe elementov za oblikovanja besedila:

```
<h1> Naslov</h1>
<h2> Podnaslov </h2>
<h3> Podpodnaslov</h3>
<h4> Naslov na 4. nivoju</h4>
<h5> Naslov na 5. nivoju</h5>
<h6> Naslov na 6. nivoju</h6>
<p> Odstavek </p>
<pre> Besedilo z določenimi
prelomi vrstic </pre>
```

```
<p>P<sub>R</sub>=U<sup>2</sup>/R</p>
```



## Osnovne značke

### Te so:

[<h1> do <h6>](#) - Naslovi in podnaslovi

[<p>](#) - Odstavek

[<br />](#) - Skok v novo vrstico

[<hr />](#) - Vodoravna črta

[<!--...-->](#) Komentar

[<pre>](#) Če želimo, da brskalnik upošteva presledke in skoke v novo vrstico, uporabimo značko [<pre>](#). Sicer zapisano besedilo spletni brskalnik prikaže enako, ne glede na to, koliko presledkov in skokov v novo vrstico naredimo.



## Značke za formatiranje besedila

<a href="#"><u>&lt;b&gt;</u></a>	Krepko izpisani znaki	<a href="#"><u>&lt;dfn&gt;</u></a>	Definicija
<a href="#"><u>&lt;i&gt;</u></a>	Ležeče izpisani znaki	<a href="#"><u>&lt;samp&gt;</u></a>	Vzorec
<a href="#"><u>&lt;u&gt;</u></a>	Podčrtano izpisani znaki	<a href="#"><u>&lt;xmp&gt;</u></a>	Neprelomljen
<a href="#"><u>&lt;center&gt;</u></a>	Na sredino pomaknjeno besedilo	<a href="#"><u>&lt;acronym&gt;</u></a>	Akronim
<a href="#"><u>&lt;em&gt;</u></a>	Poudarjeno izpisani znaki	<a href="#"><u>&lt;abbr&gt;</u></a>	Okrajšava
<a href="#"><u>&lt;big&gt;</u></a>	Večje izpisani znaki	<a href="#"><u>&lt;address&gt;</u></a>	Naslov
<a href="#"><u>&lt;strong&gt;</u></a>	Poudarjeno izpisani znaki	<a href="#"><u>&lt;blockquote&gt;</u></a>	Citat
<a href="#"><u>&lt;small&gt;</u></a>	Pomanjšano izpisani znaki	<a href="#"><u>&lt;q&gt;</u></a>	Kratka okrajšava
<a href="#"><u>&lt;sup&gt;</u></a>	Nadpisani znaki (eksponent)	<a href="#"><u>&lt;cite&gt;</u></a>	Citat
<a href="#"><u>&lt;sub&gt;</u></a>	Podpisani znaki (indeks)	<a href="#"><u>&lt;ins&gt;</u></a>	Dodam tekst
<a href="#"><u>&lt;bdo&gt;</u></a>	Krmiljenje izpisa črk od desne proti levi ali obratno	<a href="#"><u>&lt;del&gt;</u></a>	Prečrtan tekst
<a href="#"><u>&lt;code&gt;</u></a>	Besedilo programske kode	<a href="#"><u>&lt;s&gt;</u></a>	Prečrtan tekst
<a href="#"><u>&lt;tt&gt;</u></a>	Znaki izpisani kot na pisalnem stroju	<a href="#"><u>&lt;strike&gt;</u></a>	Prečrtan tekst
<a href="#"><u>&lt;kbd&gt;</u></a>	Znaki izpisani kot na tipkovnici	<a href="#"><u>&lt;font&gt;</u></a>	Velikost, tip in barva pisave
<a href="#"><u>&lt;var&gt;</u></a>	Spremenljivka		



## Seznami

- Seznane uporabimo, ko želimo izvesti naštevanje. Poznamo več vrst seznamov, ki jih uporabljamo za oštevilčenje, naštevanje ali opisovanje.
- Vrste seznamov:
  - oštevilčeni,
  - neoštevilčeni in
  - opisni.

## Oštevilčeni seznam

- Oštevilčenje seznama omogoča uporaba elementa `<ol>` (ordered list), znotraj katerega naštejemo elemente `<li>` (list item), ki predstavljajo posamezne točke seznama. Točko zaključimo z elementom `</li>`. Številčenje zaključimo z elementom `</ol>`

- Primer:

```
<ol>
```

```
  <li> Lisička </li>
```

```
  <li> Snežka </li>
```

```
  <li> Jurček </li>
```

```
  <li> Mušnica </li>
```

```
</ol>
```



## Različna možna oštevilčenja seznamov

`<h4>` Uporaba malih črk za označitev alinej`</h4>`

```
<ol type="a">
  <li>Lisička</li>
  <li>Snežka</li>
  <li>Jurček</li>
  <li>Mušnica</li>
</ol>
```

`<h4>` Uporaba velikih črk za označitev alinej`</h4>`

```
<ol type="A">
  <li>Lisička</li>
  <li>Snežka</li>
  <li>Jurček</li>
  <li>Mušnica</li>
</ol>
```

`<h4>` Uporaba velikih rimskih črk za označitev alinej`</h4>`

```
<ol type="I">
  <li>Lisička</li>
  <li>Snežka</li>
  <li>Jurček</li>
  <li>Mušnica</li>
</ol>
```

`<h4>` Uporaba malih rimskih črk za označitev alinej`</h4>`

```
<ol type="i">
  <li>Lisička</li>
  <li>Snežka</li>
  <li>Jurček</li>
  <li>Mušnica</li>
</ol>
```



## Neoštevilčeni sezname

- Tvorjenje neoštevilčenega seznama omogoča uporaba značke `<ul>` (unordered list). V nadaljevanju naštejemo elemente `<li>` (list item), ki predstavljajo posamezne točke seznama. Značka `<ul>` mora biti zaključna z značko `</ul>`

- **Primer:**

```
<ul>
  <li> Užitne gobe </li>
  <li> Neužitne gobe </li>
  <li> Pogojno užitne gobe </li>
  <li> Obisk pri zdravniku </li>
</ul>
```

## Različni neoštevilčeni sezname

<h4>Pike:</h4>

```
<ul type="disc">
  <li>Lisička</li>
  <li>Snežka</li>
  <li>Jurček</li>
  <li>Mušnica</li>
</ul>
```

<h4>Krogi:</h4>

```
<ul type="circle">
  <li>Lisička</li>
  <li>Snežka</li>
  <li>Jurček</li>
  <li>Mušnica</li>
</ul>
```

<h4>Kvadrati:</h4>

```
<ul type="square">
  <li>Lisička</li>
  <li>Snežka</li>
  <li>Jurček</li>
  <li>Mušnica</li>
</ul>
```

## Opisni sezname

- Opisni seznam tvorimo z uporabo elementa `<dl>` (definition list). V nadaljevanju naštejemo elemente z uporabo značke `<dt>` (definition term) in `<dd>` (definition description). Prvi predstavljajo pojme, ki jih opisujemo, drugi pa njihove opise. Seznam zaključimo z značko `</dl>`

- **Primer:**

```
<dl>
  <dt> PGP </dt>
  <dd> poljudni gobarski priročnik </dd>
  <dt> UG </dt>
  <dd> Užitne gobe </dd>
  <dt> SG </dt>
  <dd> Strupene gobe </dd>
  <dt> PP </dt>
  <dd> Prva pomoč </dd>
</dl>
```



## Povzetek značk za tvorjenje seznamov

<u>&lt;ol&gt;</u>	Oštevilčen seznam
<u>&lt;ul&gt;</u>	Neoštevilčen seznam
<u>&lt;li&gt;</u>	Alinea seznama
<u>&lt;dir&gt;</u>	Imenik – neoštevilčen seznam
<u>&lt;dl&gt;</u>	Seznam definicij
<u>&lt;dt&gt;</u>	Definicija
<u>&lt;dd&gt;</u>	Opis definicije
<u>&lt;menu&gt;</u>	Meni - neoštevilčen seznam



## Vstavljanje slik v dokumente

- Značka `<img>` omogoča vstavljanje slik v dokument. Značka nima zaključne značke.
- Elementu **img** lahko sledi več atributov, ki so zapisani vsi znotraj trikotnih oklepajev, ki sicer oklepajo element **img**.
- Pravimo, da je element **img** vrstični element – slika se v dokumentu zato obnaša kot velik znak.
- Atributi elementa **img**:
  - **src** določa naslov in ime datoteke s sliko – zadošča zgolj ime, če se dokument html in slika nahajata v isti mapi. Formati zapisa slike v datoteki so lahko: JPG, GIF ali PNG.
  - **width** in **height** določata velikost ( širino in višino) pravokotnega področja, v katerem bo prikazana slika. Ob odsotnosti obeh atributov bo slika izrisana v svoji naravni velikosti. Ob zapisu enega izmed atributov bo parameter drugega atributa avtomatsko določen glede na naveden atribut, tako da slika ne bo popačena.
  - **alt** omogoča izpis kratke informacije o vsebini slike. Ta se izpiše, če kurzor miške nekaj sekund miruje v področju slike. Če brskalnik ne more prikazati slike se na mestu predvidenem za sliko izpiše kratka informacija izpiše.

## Vstavljanje slik v dokumente

- **Primeri:**

```
<img src= "eiff.jpeg" width="60" alt="Eifelov stolp – določena širina" />
```

```
<img src= "eiff.jpeg" height="60" alt="Eifelov stolp – določena višina" />
```



## Poravnavanje slike z besedilom

- Atribut **align** omogoča poravnavanje slike z besedilom.
- Vrednosti atributa so lahko:
  - **left** – sliko poravna levo,
  - **right** – sliko poravna desno,
  - **middle** – sliko poravna na sredino,
  - **top** - sliko poravna na vrh strani,
  - **bottom** – sliko poravna na dno strani.
- Atributa **hspace** in **vspace** določata število pikslov levo in desno oziroma nad in pod sliko, ki pomenijo velikost presledka med sliko in besedilom.



## Poravnavanje slike z besedilom

- **Primer:**
- Poravnava slike levo od besedila:

```
<body>  
<p> Posnetek  
Eiffelovega stolpa v Parizu ponoči. Slika je na spletni strani poravnana levo glede na besedilo, ki  
obdaja sliko z desne strani. Za to poskrbi atribut <b> align </b> z vrednostjo <b> left </b>.  
</p>  
</body>
```

- Poravnava slike na desno od besedila:

```
<body>  
<p>  Posnetek  
Eiffelovega stolpa v Parizu ponoči. Slika je na spletni strani poravnana desno glede na besedilo.  
Za to poskrbi atribut <b> align </b> z vrednostjo <b> right </b>.  
</p>  
</body>
```



## Povezave znotraj dokumenta

- Osnovna značilnost hiperteksta je, da omogoča vključevanje povezav na druge dele dokumenta ali druge dokumente.
- Povezavo na druge dele dokumenta ali drugi dokument izvedemo z aktiviranjem dela besedila, kar pomeni, da bo ta del besedila postal aktiven oziroma uporabniku s klikom nanj omogočil prehod na druge dele dokumenta.
- Tako lahko izvedemo prehod (se skličemo) na nek element v dokumentu. Da se lahko prehod pravilno izvede mora biti ta element ustrezno označen. Označitev elementa izvedemo z atributom **id**. Vrednost atributa je lahko poljubno ime. Element je lahko tudi značka **<a>**, ki ji določimo vrednost atributa **name**, ki je lahko poljubno ime.

**Primer:**

```
<h1 id="kazalo">Kazalo</h1>  
<br> <a name="poglavje_1">Poglavje 1 1 - užitne gobe </a>  
<br> <a name="poglavje_2">Poglavje 2 2 - NEužitne gobe </a>
```

- Na izbran element oziroma točko se skličemo tako, da na ustreznem mestu v dokumentu vstavimo znak **#** in dodamo ime elementa oziroma točke. Če se izbran element nahaja znotraj istega dokumenta, kot aktivna povezava, lahko ime ciljne strani izpustimo.

- Primeri:

```
<b>Kazalo</b>  
<ol>  
  <li><a href="#poglavje_1">Poglavje 1 - užitne gobe </a>  
  <li><a href="#poglavje_2">Poglavje 2 - NEužitne gobe </a>  
</ol>
```



## Povezave na druge dokumente

- Osnovna značilnost hiperteksta je, da omogoča vključevanje povezav na druge dele dokumenta ali druge dokumente.
- Povezavo na drugi dokument izvedemo z aktiviranjem dela besedila, ki bo uporabniku s klikom nanj omogočil prehod na druge dokumente. Ti so lahko druga spletna stran, različni tipi dokumentov (doc, xls, pdf ...), slika, video ....
- Značka **<a>** naredi besedilo aktivno. Atribut **href** določa ime datoteke, v kateri se nahaja nov dokument oziroma nova vsebina. Aktivno besedilo je največkrat podčrtano in obarvano modro. Ko smo povezavo enkrat obiskali se aktivno besedilo obarva vijolično. Z izbiro različnih stilov lahko te oblikovne lastnosti spremenimo.

- **Primeri:**

```
<body>
```

```
Prehod na dokument <a href="gobe.html">v isti mapi.</a>
```

```
Prehod na dokument <a href="podmapa/gobe.html">v podmapi.</a>
```

```
Prehod na dokument <a href="podmapa1/podmapa2/gobe.html">dva nivoja nižje.</a>
```

```
</body>
```

```
Prehod na dokument <a href="../gobe.html"> nivo višje.</a>
```

```
Prehod na dokument na spletnem naslovu
```

```
<a href="http://www.dsplab.uni-mb.si/index.html">v korenski mapi</a> strežnika.
```

Če imena datoteke v spletnem naslovu ne navedemo se uporabi privzeta datoteka.



## Slika kot aktivni element

- Če želimo predstaviti sliko kot aktivni element, moramo navesti njeno ime za značko **<a>**.
- Aktivna slika je uokvirjena v okvir.

**Primer:**

```
<a href="pariz.html"></a>
```

## Atributi povezav značke <a>

- Z atributom **target** določimo, v katerem oknu ali okvirju se bo izpisala nova vsebina.
- Vrednost atributa je lahko poljubno ime. V primeru, da okno ali okvir z definiranim imenom obstaja, bo nova vsebina izpisana v njem. V nasprotnem se izpiše v novem oknu.
- Če ima atribut **target** vrednost **\_blank**, se nova vsebina zmeraj izpiše v novem oknu.
- **Primer:**

Izpiši vsebino `<a href="http://www.paris.fr" target="eiffel">strani</a>` v oknu poimenovanem eiffel.

Izpiši vsebino `<a href="http://www.paris.fr" target="_blank">strani</a>` v novem oknu.

## Uporaba strežniške aktivne mape na sliki

- Aktivni sliki lahko dodamo atribut **ismap**.
- Če atributu ne pripišemo vrednosti, bo brskalnik strežniku ob zahtevi po prikazu vsebine posredoval tudi koordinate točke na sliki, kjer je uporabnik izvedel kliknil.
- Točka (0,0) je definirana v levem zgornjem kotu slike; koordinata **x** narašča proti desni, koordinata **y** pa navzdol.

### Primer:

```
<a href="http://www.paris.fr">  
</a>
```

Uporabnik izvede klik v sliki na koordinatah  $x=64$ ,  $y=32$ ; brskalnik pošlje strežniku zahtevo po prikazu dokumenta  
<http://www.paris.fr?64,32>





## Možnosti aktivne mape

- Aktivna slika lahko vključuje območja, ki določajo različne cilje.
- Da to dosežemo, moramo v naprej definirati posamezna aktivna območja slike.
- Vsakemu območju slike moramo pridružiti ustrezne cilje.
- Za to uporabimo element `<map>`, ki ga poimenujemo, da se lahko nanj sklicujemo. (da se lahko skličemo nanj),
- Z vsebino elementa `<area>` opišemo posamezno območje
- Območje opišemo z atributi:
  - **shape** - določa obliko območja. Izbiramo lahko med:
    - **default** (cela slika), **rect** (pravokotnik), **circle** (krog) in **poly** (poligon).
  - **coords** določa položaj in velikost območja na sliki. Število potrebnih parametrov je odvisno od izbrane oblike območja, to je vrednosti atributa `shape`:
    - **rect** - podamo koordinate levega zgornjega in desnega spodnjega oglišča,
    - **circle** - podamo koordinate središča in polmer,
    - **poly** - podamo zaporedje oglišč, ki so definirana z ustreznim parom koordinat (x,y).
  - **href** - določa naslov cilja - datoteke, ki jo brskalnik odpre, ko uporabnik klikne na območje.
  - **alt** – podaja kratke predstavitve območij slike in informacijo, kakšna akcija sledi kliku na posamezno območje.



## Značke za delo s slikami

<code>&lt;img&gt;</code>	Slika
<code>&lt;map&gt;</code>	Slikovna mapa
<code>&lt;area&gt;</code>	Področje znotraj slikovne mape

## Značke za delo s povezavami

<code>&lt;a&gt;</code>	Povezava na oznako znotraj dokumenta ali zunanji dokument
<code>&lt;link&gt;</code>	Definicija relacije med dvema povezanima dokumentoma



## Uporaba tabel

- Za definiranje tabele uporabimo značko `<table>` .
- Atributi značke `table` so:
  - `<caption>` - komentar tabele izpisan sredinsko poravnan preko celotne širine tabele,
  - `<col >` (column) in `<colgroup>` (column group) - določata lastnosti stolpcev.
- Za urejanje izgleda tabele lahko uporabimo naslednje attribute:
  - `<td>` (table data) – uporabimo pri definiranju podatkov v celicah tabele,
  - `<th>` (table heading) – uporabimo pri definiranju prve vrstice ali stolpca tabele,
  - `<tr>` (table row) – združevanje podatkovnih celic v vrstice,
  - `<thead>`, `<tfoot >` in `<tbody>` - vrstice združujemo v skupine (glava, noga, telo). Vrstice, ki predstavljajo glavo tabele se ponovijo na vrhu vsake strani, tiste, ki predstavljajo nogo pa na koncu vsake strani. Tabela ima lahko več teles.
  - Značk za telo tabele ne pišemo, če sta noga in glava tabele prazni in imamo eno samo telo.



## Tabela

```
<table cellpadding="1" cellspacing="1" border="1" rules="all">
  <caption> <i>Ocena osebnih vozil </i> </caption>
  <colgroup width="100">
    <col align="left">
      <col span="1">
    </colgroup>
    <thead>
      <tr>
        <th colspan="2">Vrsta avtomobila </th>
        <th colspan="4"> Parametri ocenjevanja</th>
        <th rowspan="1"> Ocena </th>
      </tr>
      <tr>
        <th align="left"> Znamka </th>
        <th align="left"> Tip </th>
        <th> Teža </th>
        <th> Štev. vrat </th>
        <th> kW </th>
        <th> Poraba </th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td> </td><td>Škoda</td><td>Poprečje</td><td> 1272 </td>
        <td> 5 </td><td> 84 </td><td> 6,98 </td><td> **** </td>
      </tr>
    </tbody>
  </table>
```

```
<tbody>
  <tr>
    <td> Škoda </td><td> Oktavia </td><td> 1320 </td>
    <td> 5 </td><td> 84 </td><td> 7,1</td><td> **** </td>
  </tr>
  <tr>
    <td> Volkswagen </td><td> Golf </td><td> 970</td>
    <td> 5 </td><td> 75 </td><td> 6,8 </td><td> ****</td>
  </tr>
  <tr>
    <td> Audi </td><td> A3 </td><td> 1160 </td>
    <td> 5 </td><td> 82 </td><td> 7,0 </td><td> **** </td>
  </tr>
  <tr>
    <td> Renault </td><td> Laguna</td><td> 1470 </td>
    <td> 5 </td><td> 92 </td><td> 6,5 </td><td>**** </td>
  </tr>
  <tr>
    <td> Peugeot </td><td> 407</td><td> 1440 </td>
    <td> 5 </td><td> 87 </td><td> 7,5 </td><td>**** </td>
  </tr>
</tbody>
</table>
```



## Drugi atributi značke `table`

- **rules** – določimo vidnost črt znotraj tabele
  - **all** – prikažemo vse črte znotraj tabele
  - **cols** - prikažemo samo navpične črte med stolpci tabele
  - **groups** – prikažemo črte med posameznimi skupinami celic znotraj tabele. Skupine celic določimo z značkami `thead`, `tbody`, `tfoot` in `colgroup`
  - **none** – ne prikažemo nobene črte znotraj tabele
  - **rows** – prikažemo samo vodoravne črte med vrsticami tabele
  - **hspace** – določimo vodoravni zamik med tabelo in besedilom, ki jo obdaja
- **vspace** – določimo razmik med zgornjim in spodnjim robom tabele ter besedilom, ki jo obkroža
- **width** – določimo širino tabele. Vrednost podana v točkah ali v odstotkih širine dokumenta.



## Drugi atributi značke `table`

- **background** – prikaže sliko, ki prekrije ozadje tabele
- **bgcolor** – določimo barvo ozadja tabele. Pri ukazih `th`, `tr` in `td` lahko z istim atributom spremenimo barvo ozadja posameznih vrstic, stolpcev ali celic.
- **border** – določimo debelino robov tabele. Pri vrednosti 0 se robovi v tabeli ne izrišejo.
- **bordercolor** – določimo osnovno barvo robov tabele
- **bordercolordark** – določimo temnejšo barvo robov tabele
- **bordercolorlight** – določimo svetlejšo barvo robov tabele
- **cellpadding** – določimo razmik med robovi celic in besedilom v njih. Privzet razmik je ena točka.
- **cellspacing** – določimo razmik med posameznimi celicami tabele. Privzet razmik je 2 točki.
- **frame** – določimo vidnost zunanjih robov tabele
  - **above** – prikažemo samo črto vzdolž zgornjega roba tabele
  - **below** - prikažemo samo črto vzdolž spodnjega roba tabele
  - **border** – prikažemo vse robove (privzeta nastavitve)
  - **box** – prikažemo vse robove polnih celic
  - **hsides** – prikažemo črti vzdolž zgornjega in spodnjega roba tabele
  - **lhs** – prikažemo črto vzdolž levega roba tabele
  - **void** – odstranimo vse črte vzdolž zunanjih robov tabele
  - **vsides** – prikažemo črte vzdolž levega in desnega roba tabele
- **height** - nastavimo višino tabele (vrednost podana v točkah ali v odstotkih višine dokumenta)
- **hspace** – določimo vodoravni zamik med tabelo in besedilom, ki jo obdaja



## Urejanje celic tabele

```
<table width="600" border="1">
  <tr>
    <th align="center">Goba </th>
    <th align="right">V bolnišnici</th>
    <th align="right">Preživeli</th>
  </tr>
  <tr>
    <td align="center">Zelena mušnica</td>
    <td align="right">39</td>
    <td align="right">21</td>
  </tr>
  <tr>
    <td align="center">Rdeča mušnica</td>
    <td align="right">14</td>
    <td align="right">3</td>
  </tr>
  <tr>
    <th align="center">Skupaj</th>
    <th align="right">53</th>
    <th align="right">24</th>
  </tr>
</table>
```

Podatke v celicah lahko urejamo z naslednjimi atributi:

- vodoravna poravnava – atribut **align**:
  - vrednosti **left** (levo), **right** (desno), **center** (sredinsko) ali **justify** (obojestransko),
- navpična poravnava- atribut **valign**:
  - vrednosti **top** (navzgor), **bottom** (navzdol), **middle** (sredinsko) ali **baseline** (na osnovno črto).



## Določanje razmika v tabeli

Širina tabele je odvisna od razmikov med celicami in od širine celic.

Razmik med celicami določa atribut **cellspacing** - privzeta vrednost je 1.

Razmik med vsebino celice in robovi celice določa atribut **cellpadding**.

### Primer:

```
<table border="1" cellpadding="2" cellspacing="2">
  <tr>
    <td> 11 </td>
    <td> 12 </td>
  </tr>
  <tr>
    <td> 21 </td>
    <td> 22 </td>
  </tr>
</table>
```



## Določanje ozadja

```
<table border="2" bgcolor="green">
<caption> <i>Določanje ozadja v tabeli </i> </caption>

  <tr>
    <td >osnovno zeleno ozadje celice</td>
    <td bgcolor="blue">modro ozadje celice</td>
  </tr>
  <tr>
    <td bgcolor="red"> Rdeče ozadje celice</td>
    <td background="ozadje.jpg"> Slika kot ozadje celice</td>
  </tr>
</table>
```



## Združevanje celic

- Celice v tabeli lahko združimo v navpični ali vodoravni smeri.
- Vrstice združujemo navpično z značko **rowspan**; pri tem podamo število celic, ki jih želimo združiti.
- Stolpce združujemo z značko **colspan**, ki združi dva ali več stolpcev v en stolpec.

```
<body>
<h1> Združevanje vrstic </h1>
<table border>
<caption align=top> <i>Definiranje izgleda tabele
</i> </caption>
  <tr>
    <th >1. vrstica</th>
    <td rowspan=2>Celica 1a in 2a</td>
    <td>Celica 1b</td>
    <td>Celica 1c</td>
  </tr>
  <tr>
    <th >2. vrstica</th>
    <td>Celica 2b</td>
    <td>Celica 2c</td>
  </tr>
</table>
</body>
```

```
<body>
<h1> Združevanje stolpcev </h1>
<table border>
<caption align=top> <i>Definiranje izgleda tabele
</i> </caption>
  <tr>
    <td colspan=2>1. in 2. stolpec</td>
  </tr>
  <tr>
    <td>Celica 1</td>
    <td>Celica 2</td>
  </tr>
  <tr>
    <td>Celica 3</td>
    <td>Celica 4</td>
  </tr>
</table>
</body>
```



## Gnezdenje tabel

- Tabele predstavljajo pogosto osnovni gradnik pri pripravi spletne strani.
- Celica tabele lahko vsebuje besedilo, seznam, sliko, ali celo tabelo.
- Pravimo, da lako tabele gnezdimo, kjer posamezne celice tabele vsebujejo eno ali več tabel.
- Pri gnezdenju tabel je potrebna zmernost, da ne izgubimo preglednosti.



## Okviri

- Z okvirji razdelimo okno brskalnika na več samostojnih oken, v katerih lahko prikazujemo več HTML dokumentov hkrati.
- Posameznim razdeljenim oknom pravimo okviri. Pogosto razdelimo okno na dva ali tri okvire. V enem so prikazane povezave, v drugem pa poljubna vsebina.
- Datoteko HTML, ki opisuje velikost in položaj okvirov, sestavlja:
  - značka `<head>`
  - značka `<frameset>`, ki nadomesti značko `<body>`
- Značka `<frameset>` ima več atributov; najpomembnejša sta `cols` in `rows`
  - `rows` - opisuje razdelitev na vrstice
  - `cols` - opisuje razdelitev na stolpce
  - uporaba obeh atributov daje mrežo okvirov

## Določanje velikosti okvirov

Velikost okvira lahko določimo absolutno ali relativno:

- `rows="20%,80%"` - dve vrstici, prva zavzame 20% prostora, druga 80%
- `rows="300,*"` - dve vrstici, prva zavzame 300 pikslov, druga ostanek
- `cols="35%,*"` - dva stolpca, prvi zavzame 35%, drugi pa ostanek
- `cols="250,25%,*"` - trije stolpca, prvi 250 pikslov, drugi 25%, tretji preostanek



## Pomembnejše značke za delo z okviri

<code>&lt;frame&gt;</code>	Definira okvir
<code>&lt;frameset&gt;</code>	Definira več okvirov v dokumentu HTML
<code>&lt;noframes&gt;</code>	Definira izpis vsebine za brskalnike, ki ne podpirajo izrisa okvirov
<code>&lt;iframe&gt;</code>	Definira vrinjeni okvir



## Okviri

Atributi značke `<frame>` so:

- `bordercolor` – določa barvo robov okvira.
- `frameborder` – določa prikaz robov samostojnih področij dokumenta. Vrednost `yes` (1) izriše 3D robove, vrednost `no` (0) robov ne izriše.
- `marginheight` – določa zamik podatkov od zgornjega in spodnjega roba okvira (merilo: piksli)
- `marginwidth` – določa zamik podatkov od levega in desnega roba okvira (merilo: piksli)
- `name` – zapiše ime okvira.
- `noresize` – prepreči spreminjanje velikosti okvira
- `scrolling` – določimo prikaz drsnikov v okviru
  - `yes` – drsniki bodo vidni
  - `no` – drsniki ne bodo prikazani
  - `auto` – drsniki bodo vidni le v primeru, ko bodo podatki presegli višino in širino okvira
- `src` – vpišemo pot do podatkov oziroma spletne strani, ki bo prikazana v okviru.



## Primer skupine okvirjev v dveh stolpcih

```
<html>
<title> Postavitev okvirov</title>
<frameset cols="20%,*">
  <frame marginwidth=10 marginheight=100 noresize
  src="o_file1.html" frameborder=1>
  <frame scrolling=yes src="o_file2.html" name="okno2"
  frameborder=1>
</frameset>
</html>
```



## Kombinacija okvirov v stolpcih in vrsticah

```
<frameset rows="40%,*">

  <frame src="o_file0.html">

  <frameset cols="30%,50%,20%">
    <frame src="o_file1.html">
    <frame src="o_file2.html">
    <frame src="o_file3.html">
  </frameset>

</frameset>
```





## Prikaz brez okvirov

<noframes>

Vsebina, ki se naj bo izpisana brez okvirov

</noframes>

- Značko <noframes> uporabimo v primeru, ko vsebino z okviri zahteva brskalnik, ki ne prikaza okvirov ne podpira.
- Zapišemo jo kot vsebino značke <frameset> .
- Značko <noframes> in ustrezno vsebino bodo brskalniki, ki omogočajo prikaz okvirov ignorirali.
- Brskalniki, ki ne omogočajo izpisa okvirov bodo vsebino izpisali.



## Atributa target in name

- Vsebino spletnih strani prikažemo v poljubnem okviru s pomočjo dveh atributov.
- Z atributom name značke `frame` določimo ime okvira
- Z atributom target in značke `a` ter atributa `href` <a href ...> določimo ime oziroma okvir v katerega se bo izpisala vsebina.

```
<a href="o_file1.html" target="okno1">
```

- V primeru, da okvir z danim imenom ne obstaja, se bo vsebina prikazala v novem oknu.
- Atributu `target` lahko pripišemo tudi eno od vnaprej definiranih vrednosti:
- `_blank` - odpre dokument v novem oknu
- `_self` - odpre dokument v istem oknu
- `_parent` - odpre dokument v oknu, ki je oče trenutnega okna
- `_top` - odpre dokument v celem oknu - ne zgolj v okviru



## Oblikovanje obrazcev

- Zelo pogosto spletne strani omogočajo vnos podatkov v obrazce.
- Obrazec definiramo z značko `<form>`.
- Obliko obrazca določa značka `input type`
- Izbiramo med štirimi osnovnimi oblikami:
  - polje za vpis besedila (`text` in `textarea`),
  - kvadratke za potrditev (`checkbox`),
  - izbirne gumbе (`radiobutton`)
  - sezname (`selectionlist`).
- Uporabimo lahko tudi dva gumba:
  - `submit`, podatke posreduje oziroma shrani
  - `reset`, vrne vsebino vseh polj v obrazcu na začetne vrednosti



## Oblikovanje obrazcev

- Značka `<form>` predstavlja ogrodje obrazca.
- Najpomembnejši atributi značke `<form>` so:
  - `method`, določa način prenosa podatkov do strežnika
    - možni vrednosti sta `get` in `post`,
  - `action`, določimo naslov, kjer je shranjen program oziroma skripta, ki podatke vnesene v obrazec obdela oziroma shrani.
  - `enctype`, določa način kodiranja podatkov,

### Primer:

```
<form method= "post" action= "pot/program.pl">  
    ... zapis ukazov za zapis ali izbiro podatkov ...  
</form>
```



## Oblikovanje obrazcev

- Značka `<input type>`
- Z ukazom `input type` določimo vrsto polja oziroma elementa v obrazcu.
- Izbiramo med naslednjimi vrednostmi:
  - `text`, prikaže okence za vnos besedila. Pri tem gre za krajše besedilo (ime, priimek, naslov...).
  - gumba `submit` in `reset`;
  - `checkbox`, v obrazec postavimo kvadratega za potrditev. Ta uporabniku dopušča, da potrdi eno ali več izbir, ki jih ima na voljo. Izbiro potrdimo s klikom na kvadratega, v katerem se izriše križec.

**Primer:**

.....

Izberite ena ali več področij, ki vas zanimajo:<br>

```
<input type="checkbox" name="vrednost1" checked>prosojnice<br>
```

```
<input type="checkbox" name="vrednost2" checked>učbeniki<br>
```

```
<input type="checkbox" name="vrednost3" checked>brošure<br>
```

```
<input type="checkbox" name="vrednost4" checked>izročki<br>
```

.....

- `radio`, izbirni gumb omogočajo izbiro ene izmed več možnosti. Označena izbira je izrisana s črno piko v središču okroglega gumba.



## Oblikovanje obrazcev

- izbirni gumb `radio`, prikažemo s pomočjo značke `<input type="radio">`. Značka ima naslednja atributa
  - `name`, določa ime spremenljivke
  - `value`, določa ali je izbirni gumb privzeto označen.

**Primer:**

.....

Izberite ena ali več področij, ki vas zanimajo:<br>

```
<input type="radio" name="skupina1" value=1izbira  
checked>prosojnice<br>
```

```
<input type="radio" name="skupina1" value=2izbira>učbeniki<br>
```

```
<input type="radio" name="skupina1" value=3izbira>brošure<br>
```

```
<input type="radio" name="skupina1" value=3izbira>izročki<br>
```

.....



## Oblikovanje obrazcev

- za polje `password` se odločimo, ko želimo, da uporabnik vpiše geslo. Namesto besedila se v okencu izpisujejo zvezdice. prikažemo s pomočjo značke `<input type="radio">`.

Primer:

```
.....  
<input type= "password" name= "geslo">  
.....
```



## Oblikovanje obrazcev

- Seznam elementov v meniju tvorimo z značko `<select>`. Odločimo se lahko za eno ali več izbir v seznamu. Določimo lahko, katera izbira bo označena kot privzeta.
- Seznam elementov prikažemo v bloku `<select>` in `</select>` .
- Posamezne izbire naštejemo z značko `<option>`.
  - Atribut `name` določa ime spremenljivke.
  - Atribut `multiple` omogoča izbiro več elementov v seznamu.
  - Atribut `multiple size` določa število elementov v seznamu.



## Oblikovanje obrazcev

### Primer:

.....  
Izberite eno izmed področij, ki vas zanimajo:<br>  
<select name= "spisek "  
<option selected>prosojnice  
<option>revije  
<option>brošure  
<option>izročki  
</select><br><br><br>

.....  
Izberite eno ali več področij, ki vas zanimajo:<br>  
<select name= "spisek1" multiple size="6">  
<option selected>prosojnice  
<option>revije  
<option>brošure  
<option selected>izročki  
<option>učbeniki  
</select><br>

.....



## Oblikovanje obrazcev

- Značka **textarea** prikaže polje, v katerega lahko vpišemo več vrstic besedila.
- Atributi značke **<textarea>** so.
  - **cols** določa širino polja za vpis besedila,
  - **name** določa ime spremenljivke, kamor shranimo zapisano besedilo po kliku gumba **submit**
  - **rows** določa višino polja za vpis besedila.

### Primer:

...  
<textarea name="zapis" rows=50 cols=50>  
Polje za vpis obsežnejšega besedila.  
</textarea>

...



## Značke za tvorjenje obrazcev

<code>&lt;form&gt;</code>	definira obrazec
<code>&lt;input&gt;</code>	definira polje v obrazcu
<code>&lt;textarea&gt;</code>	definira polje za daljše besedilo
<code>&lt;button&gt;</code>	definira gumb
<code>&lt;select&gt;</code>	definira element izbirnega seznama
<code>&lt;optgroup&gt;</code>	definira več opcij
<code>&lt;option&gt;</code>	definira opcijo v obrazcu
<code>&lt;label&gt;</code>	definira označbo v obrazcu
<code>&lt;fieldset&gt;</code>	definira polje
<code>&lt;legend&gt;</code>	definira napis v obrazcu



## Primer bolj zapletenega obrazca

```
<body>
<h1>Vprašalnik</h1>
<p>Prosimo izpolnite spodnji vprašalnik.</p>
<form method = "post" action = "/cgi-bin/formmail">
  <p>
    <input type = "hidden" name = "recipient"
      value = "dsplab@uni-mb.si" />
    <input type = "hidden" name = "subject"
      value = "vprašalnik" />
  </p>
  <p><label>Ime:
    <input name = "ime" type = "text" size = "25" />
  </label></p>
  <p><label>Priimek:
    <input name = "priimek" type = "text" size = "30" />
  </label></p>
  <p><label>Komentarji:<br />
    <textarea name = "komentarji" rows = "6"
      cols = "36"></textarea>
  </label></p>
  <p><label>Elektronski naslov:
    <input name = "email" type = "text"
      size = "25" /></label></p>
  <p>
    <strong>Kaj vam je bilo všeč:</strong><br />
    <label>Dizajn
      <input name = "vsec" type = "checkbox"
        value = "Dizajn" /></label>
    <label>Povezave
      <input name = "vsec" type = "checkbox"
        value = "Povezave" /></label>
    <label>Uporabnost
      <input name = "vsec" type = "checkbox"
        value = "Uporabnost" /></label>
    <label>Slike
      <input name = "vsec" type = "checkbox"
        value = "slike" /></label>
    <label>Izvorna koda
      <input name = "vsec" type = "checkbox"
        value = "Izvorna koda" /></label>
  </p>
</form>
</body>
```



## Primer bolj zapletenega obrazca

```
<!-- <input type = "radio" /> ustvari radijski -->
<!-- gumb. Razlika med radijskim gumbom in -->
<!-- checkboxes je, da je z radijskim gumbom -->
<!-- mogoče izbrati le en element v skupini. -->
<p>
  <strong>Kako ste nas našli?:</strong><br />

  <label>Iskalnik
    <input name = "najti" type = "radio"
      value = "iskalnik" checked = "checked" />
  </label>

  <label>Povezava iz druge strani
    <input name = "najti" type = "radio"
      value = "link" /></label>

  <label>Domača stran
    <input name = "najti" type = "radio"
      value = "deitel.com" /></label>

  <label>Referenca v knjigi
    <input name = "najti" type = "radio"
      value = "book" /></label>

  <label>Drugo
    <input name = "najti" type = "radio"
      value = "other" /></label>
</p>
```



## Slogi za spletne strani

- Slogi vključujejo mnoge oblikovne lastnosti besedila.
- V slogu so lahko shranjene oblikovne značilnosti, kot so na primer vrsta in velikost pisave, barva itd.
- Za oblikovanje dokumentov html imamo na voljo sloge, ki so določeni v zbirki slogov (style sheet).
- Značka `<style>`, s katero definiramo sloge, mora biti zapisana v glavi dokumenta HTML za ukazom `<html>` in med ukazoma `</title>` ter `<body>`.

### Primer:

```
<style type="text/css">
  <!--
  body { background: green; margin-left: 20px }
  p { font-family: helvetica; font-size: 12pt; color:red}
  h1, h2 { font-family: helvetica, arial }
  // -->
</style>
```



## Slogi za spletne strani

- `<style type="text/css">` - najava uporabe slogov
- `body { background: green; margin-left: 20px }` – definiran je zelenega ozadja in levega roba, odmaknjenega za 50 točk.
- `p { font-family: helvetica; font-size: 12pt; color:red }` – vsak tekst, ki se prične z ukazom `<p>`, oblikovani s pisavo vrste Helvetica ali Arial velikosti 12 točk in rdeče barve.
- Komentar se zaključi in zbirko slogov zapremo z značko `</style>`.

**Primer:**

```
<style type="text/css">
  <!--
    body { background: green; margin-left: 20px }
    p { font-family: helvetica; font-size: 12pt; color:red}
    h1, h2 { font-family: helvetica, arial }
  // -->
</style>
```



## Slogi za spletne strani

- `<style type="text/css">` - najava uporabe slogov
- `body { background: green; margin-left: 20px }` – definiran je zelenega ozadja in levega roba, odmaknjenega za 50 točk.
- `p { font-family: helvetica; font-size: 12pt; color:red }` – vsak tekst, ki se prične z ukazom `<p>`, oblikovani s pisavo vrste Helvetica ali Arial velikosti 12 točk in rdeče barve.
- Komentar se zaključi in zbirko slogov zapremo z značko `</style>`.

**Primer:**

```
<style type="text/css">
  <!--
    body { background: green; margin-left: 20px }
    p { font-family: helvetica; font-size: 12pt; color:red}
    h1, h2 { font-family: helvetica, arial }
  // -->
</style>
```





## Slogi za spletne strani

- V sloge lahko vključimo mnoge atribute – naštejmo najpogosteje uporabljane:

Atribut	Pomen
font-family: helvetica, arial	Vrsta pisave
font-size: 14 pt (1cm)	Velikost pisave
color: red (blue)	Barva
background: silver (yellow)	Barva ozadja
font-weight: bold (bold, light, normal)	Debelina pisave, npr. krepka, navadna ...
font-variant: small-caps	Različica pisave, npr. male črke
font style: italic	Slog pisave, npr. ležeče
line-height: 12pt (2 cm)	Višina vrstice, v točkah ali cm
margin: 10px (2cm)	Rob, v točkah ali cm
margin-right: 2cm	Desni rob
margin-left: 15px	Levi rob
margin-top: 10px	Zgornji rob
margin-bottom: 10px	Spodnji rob



## Slogi za spletne strani

### Oblikovanje odstavkov

- Ko sloge definiramo jih lahko uporabimo pri oblikovanju vsebine.
- Nov odstavek, ki smo ga v slogu definirali z značko `<p>` lahko uporabimo na naslednji način:  
`<p class="quote"> Odstavek oblikovan s slogom </p>`  
ali  
`<p class="shade"> Senčeno besedilo v odstavku.</p>`
- Sloga `quote` in `shade` lahko definiramo na naslednji način:  
`p.quote { font-family: times; font-size: 10pt; color: blue }`  
`p.shade { font-family: arial; font-size: 10pt; background: gray }`
- Na podoben način lahko oblikujemo sezname z značko `<li>`  
`<li class="circle">`
- Slog `circle` lahko definiramo na naslednji način:  
`li.circle { list-style-type: circle }`



## Slogi za spletne strani

### Oblikovanje znakov

- Besedilo med značkama <b> in </b> želimo oblikovati hkrati s krepko in ležečo pisavo.
- Besedilo med značkama <i> in </i> želimo obarvati modro.

```
b { font-style: italic }  
i { color: blue }
```

#### Primer:

```
<html>  
<head>  
<title>Primer uporabe sloga</title>  
</head>  
<style type="text/css">  
p.quote { font-family: times; font-size: 10pt; color: blue }  
p.shade { font-family: arial; font-size: 10pt; background: gray }  
</style>  
<body>  
<p class="quote"> Besedilo izpisano s slogom.</p>  
<p class="shade"> Besedilo izpisano s senčenjem.</p>  
</body>  
</html>
```



## Slogi za spletne strani

### Povezave na zunanje zbirke slogov

- Zbirke slogov lahko shranimo v posebno tekstovno datoteko, do katere dostopa html dokument.
- Prednost: iste zbirke slogov je mogoče uporabiti v različnih dokumentih, ne da bi morali sloge zapisovati vanje vsakokrat znova.

- **Postopek:**

- najprej ustvarimo tekstovno datoteko, ki jo shranimo s končnico .css – na primer slog1.css
- vanjo zapišemo definicije slogov in nobenih ukazov v jeziku html.

#### Primer:

```
h1 { font-family: arial, helvetica; font-size: 20pt;}  
h2 { font-family: arial, helvetica; font-size: 16pt;}  
h3 { font-family: arial, helvetica; font-size: 14pt;}  
p { font-family: arial, helvetica; font-size: 12pt; padding-left: 30px }
```

- dokument html povežemo z datoteko slogov. To izvedemo v glavi datoteke.

```
<link rel="stylesheet" type="text/css" href="slog1.css"
```

- v dokumentu html sedaj ni potrebno pisati značk <style> in </style>.



## Slogi za spletne strani

Zapis celotne kode z vključitvijo sloga:

### Primer:

```
<html>
<head>
<title>Primer uporabe sloga v datoteki </title>
</head>
<link rel="stylesheet" type="text/css" href="slog1.css" >
<body>
<h1> Naslov razdelka 1 </h1>
<h2> Naslov razdelka 2.</h2>
<h3> Naslov razdelka 3 </h3>
<p> Besedilo razdelka.</p>
</body>
</html>
```



## JavaScript

JavaScript je:

- skriptni jezik medmrežja,

JavaScript ni JAVA (JAVA je objektno orientiran programski jezik, ki so ga razvili v podjetju Sun Microsystems)

JavaScript je vključen direktno v HTML

JavaScript je "interpretacijski jezik" - izvaja se brez predhodnega prevajanja – enako kot HTML)



## JavaScript

- Razvil ga je Netscape leta 1995
- Prva podpora: Netscape Navigator 2.0
  - JavaScript 1.0
- Objektno zasnovan
- Teče na strani klienta – teče v okviru brskalnika
- Prihranek na obsegu komunikacije med klientom in strežnikom.
- JavaScript je med najpopularnejši skriptni jezik med razvijalci internetnih strani in storitev.
- Podpirajo ga vsi pomembnejši brskalniki: Internet Explorer, Mozilla, Firefox, Netscape, Opera.



JavaScript najpogosteje uporabljamo za:

- izvajanje za izračunov,
- dodajanje interaktivnosti,
- izboljšanje oblike oziroma izgleda strani,
- kreiranje in uporaba piškotkov.

JavaScript je brezplačen.



Z JavaScriptom lahko:

- programiramo v HTML-ju (definiranje spremenljivk,...),
- dodamo dinamične in interaktivne lastnosti,
- spreminjamo elemente HTML,
- izvedemo reagiranje na dogodke,
- Izvajamo posebne grafične in tekstovne učinke,
- potrdimo/zavrnamo izpolnjene obrazce.



- Programiranje v JavaScriptu sloni na posebni sintaksi.
- Razumevanje izrazov JavaScript je osnova za razumevanje kode zapisane v JavaScriptu.
  - Objekti, lastnosti, metode, dogodki, funkcije, vrednosti, spremenljivke, izrazi, operatorji.
- Objekti: objekti se nanašajo na okna, dokumente, slike, tabele, formularje, gumbе, povezave ...
- Objekti morajo imeti svoje ime.
- Objektivom so pripisane lastnosti, s pomočjo katerih lahko nanje vplivamo.
- Lastnosti so atributi objektov.



- Lastnosti so atributi objektov.
- Lastnosti objektov definiramo z uporabo imena objekta, pike in imena lastnosti.
  - Npr.: barvo ozadja določimo s sintakso:
    - `document.bgcolor`
    - *document* je objekt
    - *bgcolor* je lastnost

## Metode

- Metode so akcije izvedene nad posameznimi objekti.
- Metode so kar objekti lahko naredijo:
  - npr., `document.write("Dober dan!")`
    - *document* je objekt
    - *write* je metoda



## Dogodki:

- Dogodek povezuje objekt z akcijo:
  - npr., akcija dogodka `OnMouseover` lahko spremeni sliko
  - npr., akcija dogodka `on Submit` pošlje obrazec.
- Uporabnikove akcije prožijo dogodke.

## Funkcije:

- Funkcije izvajajo posamezna opravila.
  - npr., `function doWhatever (){... sledijo ukaze vrstice ...}`
  - zaviti oklepaji vključujejo ukazne vrstice funkcije.
- JavaScript ima že vgrajene funkcije, uporabnik lahko definira svoje funkcije.



### Vrednosti:

- Vrednosti lahko definiramo na naslednji način:
  - števila: 1,2,3 ...
  - znaki: znaki zajeti v narekovajih
  - Boolean: true ali false
  - objekt: slika, obrazec
  - funkcija: validate, doWhatever ...

### Spremenljivke:

- Spremenljivke vsebujejo vrednosti. Spremenljivki priredimo vrednost s pomočjo enačaja.
- Spremenljivke definiramo z uporabo ukaza var z ali brez prireditve začetne vrednosti.
  - npr., var pi
  - npr., var pi= 3,141592653589793238462643383279502884



### Izrazi:

- Izrazi so ukazi, ki priredijo spremenljivkam vrednosti.
- Izraz zmeraj uporablja prireditveni operator, takšen kot enačaj
  - npr., var mesec = maj; je izraz
- Zapis izraza zaključimo s podpičjem.

### Operatorji:

- Operatorje uporabljamo za izvajanje operacij nad spremenljivkami.
- Tipi operatorjev:
  - aritmetični operatorji (plus)
  - operatorji primerjanja (enačaj)
  - logični operatorji (in)
  - kontrolni operatorji (if)
  - znakovni operatorji



## Načini uporabe JavaScripta

- JavaScript je lahko zapisan na ločeni strani.
- JavaScript je lahko vgrajen v HTML dokumente
- Atributi objekta JavaScript so lahko vključeni kot elementi v značke HTML
  - npr., <body onLoad="alert('Dobrodošli')">.



## JavaScript in HTML:

```
<html>
<body>
  <script type="text/javascript">
    document.write("Izpis z ukazom JavaScript")
  </script>
</body>
</html>
```

Ukaz JavaScript





## Možna mesta zapisa kode JavaScript v dokument HTML:

- v glavo (<head>):

```
<html>
<head>
    <script type="text/javascript">
        document.write("Izpis z ukazom
JavaScript.")
    </script>
</head>
</html>
```

- izvede se pred naložitvijo celotne strani.



- v telo <body>

```
<html>
<body>
    <script type="text/javascript">
        document.write("Izpis z ukazom
JavaScript")
    </script>
</body>
</html>
```

- koda se izvede, ko je stran naložena.



- Oboje: glava (<head>), telo (<body>)

```
<html>
<head>
  <script type="text/javascript">
    ....
  </script>
</head>
<body>
  <script type="text/javascript">
    ....
  </script>
</body>
</html>
```



Uporaba skripte zapisane v ločeni datoteki:

```
<html>
<head>
  <script src="primer.js"> </script>
</head>
<body>
</body>
</html>
```



## Uporaba spremenljivk:

```
var ime_spremenljivke =  
    vrednost  
    ali  
ime_spremenljivke = vrednost
```

```
var ime_spremenljivke = "avto"  
    ali  
ime_spremenljivke = "avto"
```



## Pogojni stavki

```
if (pogoj)  
{  
    koda  
}
```

```
if (pogoj)  
{  
    koda 1  
}  
else  
{  
    koda 2  
}
```

```
if (pogoj_1)  
{  
    koda 1  
}  
else if (pogoj_2)  
{  
    koda 2  
}  
else  
{  
    koda 3  
}
```



## Stavek SWITCH:

```
switch(n)
{
  case 1: (če n=1) izvedi kodo 1...
  break
  case 2: (če n=2) izvedi kodo 2...
  break
  default: izvede se, če n ni ne 1 ne 2
}
```



## Stavek FOR

```
for (i=0;i<=20;i++)
{
  koda
}
```

## Stavek WHILE

```
var n=0
while (n<3)
{
  document.write("Stevilo " + n)
  document.write("<br>")
  n++
}
```



## Primer kode:

```
<html>
<body>

<script type="text/javascript">
for (i = 1; i <= 6; i++)
{
  document.write("<h" + i + ">Poglavje " + i)
}
</script>

</body>
</html>
```



## Matematični operatorji

+	(plus)	x=y	(enakost)
-	(minus)	x+=y	(x=x+y)
*	(množenje)	x-=y	(x=x-y)
/	(deljenje)	x*=y	(x=x*y)
%	(ostanek pri deljenju)	x/=y	(x=x/y)
++	(povečaj za 1)		
--	(zmanjšaj za 1)		
==	(je enako – npr. številka)		
===	(je enako – številka IN tip)	&&	(logični IN)
!=	(neenakost)		(logični ALI)
<	(manjše)	!	(logični NE)
>	(večje)		



## Uporabniške funkcije

```
<script type="text/javascript">
function primerfunkcije(parameter1, parameter2,...)
{
  ...
}
</script>
```

### Funkcija, ki vrne parameter:

```
<script type="text/javascript">
function primerfunkcije(parameter1, parameter2,...)
{
  return parameter1+parameter2
}
</script>
```



## Primer uporabe funkcije

```
<html>
<head>
<script type="text/javascript">
function primerfunkcije(parameter)
{
  alert(parameter)
}
</script>
</head>

<body>
<form>
<input type="button"
  onclick="primerfunkcije('Sit!')"
  value="Užitne gobe">

<input type="button"
  onclick="primerfunkcije('Bolnica!')"
  value="Neužitne gobe">
</form>

</body>
</html>
```



## Primer uporabe funkcije, ki vrne parameter

```
<html>
<head>
<script type="text/javascript">
function deljenje(m,n)
{
  return m/n
}
</script>
</head>

<body>
<script type="text/javascript">
document.write(deljenje(12,4))
</script>

</body>
</html>
```



## Opozorilna okna:

```
<html>
<head>

<script type="text/javascript">
function disp_alert()
{
  alert("Opozorilna škatla!! ")
}
</script>
</head>

<body>
<form>
<input type="button" onclick="disp_alert()" value="Prikaži opozorilo">
</form>

</body>
</html>
```



### Opozorilna okna (2):

```
<html>
<head>
<script type="text/javascript">
function disp_confirm()
{
var name=confirm("Izberite gumb!")
if (name==true)
{
document.write("Pritisnili ste OK!")
}
else
{
document.write("Pritisnili ste Prekliči!")
}
}
</script>
</head>

<body>
<form>
<input type="button" onclick="disp_confirm()" value="Prikaži potrdilo">
</form>
</body>
</html>
```

---

127



### Opozorilna okna (3):

```
<html>
<head>
<script type="text/javascript">
function prikaz_poziva()
{
var name=prompt("Vnesite ime:", "")
if (name!=null && name!="")
{
document.write("Pozdravljeni " + name + "! Koliko je ura?")
}
}
</script>
</head>

<body>
<form>
<input type="button" onclick="prikaz_poziva()" value="Prikaži vnos">
</form>
</body>

</html>
```

---

128





## Večpredstavnost

- Spletne strani lahko vključujejo različne tipe podatkov oziroma informacij.
    - Tekst, hipertekst, strukturirana besedila zapisana v določenem formatu (pdf, doc,...),
    - zvok, zapisan v datotekah v različnem formatu,
    - Slike, video, animacija.
  - Za predvajanje nekaterih vsebin potrebujemo dodatne programe, kot je na primer video predvajalnik.
  - Za izvajanje interakcije pa uporabimo posebni zapis v datotekah (flash datoteke) ali programe zapisane v programskem jeziku java.
- 



## Večpredstavnost

Spletne strani lahko vključujejo statične in dinamične večpredstavne tipe podatkov.

Statični večpredstavni tipi so:

besedilo,  
slika,  
grafika.

Dinamični večpredstavni tipi so:

digitalni zvok,  
digitalni video,  
animacija,  
interaktivne vsebine.

## Večpredstavnost

Podatki so lahko glede na tip podatka zapisani v različnih formatih.

- Besedila so v txt, pdf, Tex, dvi, postscript (ps, ai, eps), rtf ali doc formatu.
- Zvočna informacija je v au, aiff, wav, mpeg2 (mp2), mp3, ra (real audio), snd, mid formatu.
- Slike so v gif, png, tiff, jpeg (jpg), bmp, pcx, tga in pbm formatu.
- Filmi v mpeg (mpg, mpeg, mpe), quicktime (qt, mov), rv, ram (realvideo), ms-video (avi) formatu, flash (swf) .

Podatki so lahko še dodatno stisnjeni: zip, gz, tar, arj, arc, Z ....

Microsoft je za multimedijske podatke definiral več različnih formatov:

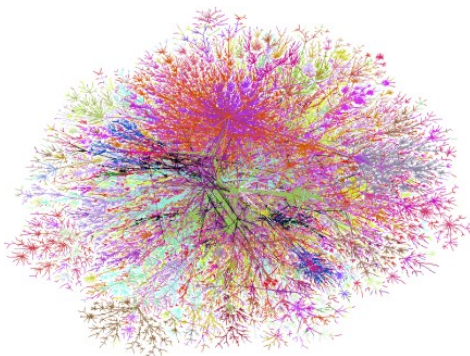
- Windows media formati v asf (Advanced Streaming Format), wma (Windows Media Audio), asx (meta podatki), wmv (Windows Media Video), wax (Windows Media Audio Redirector – meta podatki za datoteke wma).

## Različni tipi podatkov

### Slika

Bitna slika - točke slike so predstavljene z barvo.

Vektorska slika – slika je predstavljena z obliko in barvo.



## Spremljajoči podatki digitalne slike

- Barvni model: ločimo dva barvna modela
  - aditivni model: želen barvni odtenek določimo s seštevanjem barvnih komponent – model RGB.
  - subtraktivni barvni model: želen barvni odtenek določimo z odštevanjem barvne komponente od bele barve – model CMY.
- Ločljivost digitalne slike: podaja število slikovnih elementov na palec (dpi).
- Število kanalov: število barvnih komponent, ki jih vključuje opis barve slikovnega elementa. Določeno z uporabo barvnega modela. Najpogosteje ločimo tri kanale.
- Barvna globina: število bitov, ki jih uporabimo za kodiranje barve posameznega slikovnega elementa. Najpogosteje je posamezni kanal kodiran s tremi biti.
- Prepletanje (interlacing) določa organizacijo barvnih komponent slikovnih elementov. Lahko je neprepletana (posamezne komponente so podane skupaj za vse slikovne elemente) ali prepletana (komponente za posamezni slikovni element si sledijo ena za drugo).

## Izvajanje operacij nad slikami

- Urejanje:
  - spreminjanje barve posameznega piksla, rezanje, kopiranje, lepljenje skupine pikslov.
- Geometrijske transformacije:
  - Razteg, rotacija, zrcaljenje in ukrivljanje.
- Filtriranje:
  - izostrenje, poudarjanje robov objektov, spreminjanje barvne slike v sivinsko sloko.
- Zlivanje:
  - zlitje dveh ali več slik v novo sliko. Zlivanje izvedeno z uporabo Boolovih operacij.
- Pretvorba: operacija ne spremeni videza slike.
  - zamenjava barvnega modela (RGB – CMY).



## Postopki kompresije digitalnih slik

Poznamo postopke izgubne in brezizgubne kompresije.

Postopki brezizgubne kompresije:

- **Bitna mapa (BitMaP, BMP)** - MS Windows - aparaturno neodvisen format zapisa podatkov digitalne slike. Slikovni element je zakodiran s 24 biti. Uporablja preproste postopke kompresije podatkov ali pa sploh ne. Rezultat so zelo velike datoteke. Format ni primeren za arhiviranje podatkov.
- **Graphics Interchange Format (GIF)** – uporablja postopek kompresije LZW. Primeren je za zapis sivinskih in barvnih slik z največ 256 različnimi barvnimi odtenki.
- **Tag Image File Format (TIFF)** – načrtovan za izmenjavo slik med uporabniškimi programi in rasterskimi razbirnimi napravami. Uporabljan za namizno založništvo, video aplikacije, medicinske in satelitske posnetke ter shranjevanje dokumentov. Podpira barvna modela CMY in RGB. Primeren za zapis sivinskih slik (do 256 vrednosti sive) in slike v resničnih barvah (true color) (24 bitov za vsak piksel). Uporablja različne metode stiskanja, med njimi LZW, RLL in prilagodljivo Huffmanovo kodiranje.



## Postopki izgubne kompresije

- Stopnja kompresije je večja
- Kvaliteta restavrirane slike je slabša
- Možen nadzor stopnje izgube kvalitete – iščemo kompromis med kvaliteto slike in stopnjo kompresije (velikostjo zapisa).
- Najpogosteje uporabljan format je JPEG.



## Avdio signal

- Človek zaznava akustične signale v frekvenčnem področju med 20 Hz in 20kHz.
- Digitalni avdio signal tvorimo z vzorčenjem (A/D pretvorbo) analognega signala.
- Pri predvsjanju digitalnega avdio signala najprej izvedemo pretvorbo iz digitalne v analogno obliko (D/A pretvorbo).
- Najpogostejše frekvence vzorčenja analognega avdio signala: 8, 16, 44.1, 48, 96 kHz.



## Digitalni zvok

- **Število kanalov** - dva kanala (stereo zvok), pet kanalov (prostorski zvok), v profesionalnih aplikacijah uporaba vse do 64 kanalov.
- **Prepletanje:** Način zapisa podatkov o posameznem kanalu - neprepleteno ali pa prepleteno. Pri slednjem je sinhronizacija kanalov zagotovljena, omogoča tudi učinkovitejše stiskanje.
- **Kodiranje:** Kot pri sliki obstajajo tudi pri avdio signalu brezizgubni in izgubni postopki kopresije podatkov. Najpogosteje uporabljeni postopki so postopki iz družine MPEG (Moving Picture Expert Group). Omogočajo štiri različne načine predstavitve avdio signala: mono, dvojni mono (dva neodvisna kanala), stereo (dva prepletena kanala) in sestavljeni stereo (upoštevanje redundance med levim in desnim kanalom).



## Vključitev digitalnega zvoka v HTML

Obstaja več načinov vključevanja zvoka, kjer pa precej teh načinov podpira samo nekaj brskalnikov.

Najpreprostejši način vključevanja zvoka je naslednji:

```
<html>
<head>
  <title> Zvok</title>
</head>
<body>
  <a href="zvok.wav">
    Klik za poslušanje zvoka!</a>
</body>
</html>
```



## Vključitev videa v HTML

Med video datoteke uvrščamo animirane in video filme, ki so lahko posneti s kamero, preneseni z video rekorderja, televizije ali nosilca video vsebin (CD, DVD).

Najpreprostejši način vključevanja videa je naslednji:

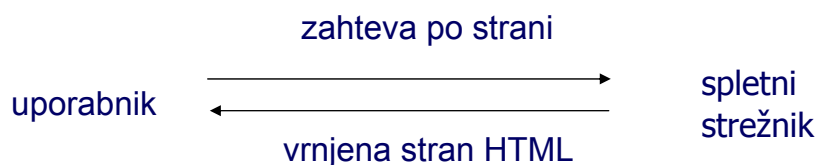
```
<html>
<head>
  <title> Video </title>
</head>
<body>
  <a href="film.avi">
    Klik za ogled videa!</a>
</body>
</html>
```



## Aktivne spletne strani

### Značilnosti statične spletne strani

- Vsebina je statična
- Vsebuje le kodo HTML
- Ob vsakem dostopu je stran enaka
- Interaktivno omejena
- Potek komunikacije:



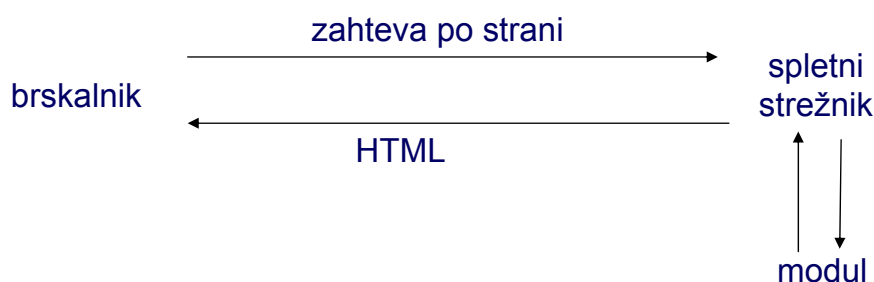
## Dinamične spletne strani

- Vsebina je dinamična – dinamični elementi se kreirajo na zahtevo
- Statični del kode HTML kode je napisan v naprej
- Dinamični elementi so napisani v skriptnem ali programskem jeziku
- Ločimo dva primera:
  - Stran strežnika
  - Stran klienta

## Stran strežnika

- Dinamične elemente generira spletni strežnik.
- Prednosti:
  - strani so enake na vseh brskalnikih
  - koda dinamičnih elementov je uporabniku skrita
- Slabosti:
  - obremenitev strežnika

Delovanje:



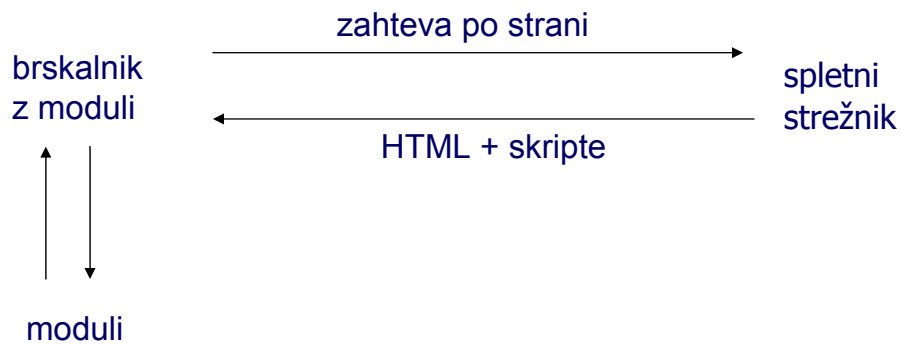
## Stran klienta

- Brskalnik na strani uporabnika kreira dinamične elemente
- Prednosti:
  - brskalnik s svojimi zahtevami ne obremenjuje strežnika
- Slabosti:
  - mogoč dolgotrajen prenos datotek
  - slaba varnost
  - različen prikaz dinamičnih elementov





- Delovanje:



## Tehnologije strani klienta

- JavaScript
- VBScript
- ActiveX
- Java applets



## JavaScript

- Prvi skriptni jezik
- Razvila sta ga Netscape in Sun
- Primeren za manjše aplikacije
- Microsoft je razvil svojo različico pod imenom JScript



## VBScript

- Razvil ga je Microsoft
- Prvič implementiran v Internet Explorer 3.0
- Temelji na programskem jeziku Visual Basic
- Prednost - lahek za učenje
- Slabosti - deluje samo z brskalnikom Internet Explorer



## Stran strežnika

- CGI
- ASP
- PHP
- ASP.NET
- JSP
- ColdFusion



## CGI Common Gateway Interface

- CGI je bila prva tehnologija, ki je omogočala razvoj dinamičnih spletnih strani
- Vmesnik, ki odjemalcu omogoča klicanje CGI programov na strežniku
- CGI programe lahko izdelamo v različnih jeziki
  
- Prednosti: deluje na vseh strežnikih in platformah, možnost uporabe pravih programskih jezikov,
  
- Slabost: ob vsakem klicu programa se zažene nov proces

## ASP (Active Server Pages)

- Tehnologija podjetja Microsoft
- V začetku je ASP podpiral le Microsoft IIS strežnik, sedaj tudi drugi
- Kodo ASP je bilo mogoče na začetku pisati le v VBScript in Jscript, danes pa tudi v drugih skriptnih jezikih
- Prednosti: enostaven za učenje
- Slabost: uporaba le skriptnih jezikov

## PHP: *Hypertext Preprocessor*

- Prvo različico je izdelal Rasmus Lerdorf leta 1994
- Kompatibilen skoraj z vsemi spletnimi strežniki
- Podpira veliko podatkovnih baz (Oracle, Informix, MySQL, Solid, PostgreSQL ...)
- Deluje na večini platform (Windows, Linux, Unix, etc.)
- Prednosti:
  - je brezplačen
  - odprtokodni (OSS)
  - enostaven za učenje

## ASP.NET

- Je najnovejša Microsoftova tehnologija za izdelavo dinamičnih strani
- Je neke vrste združitev ASP in .NET
- Najnovejša verzija je 2.0
- Prednosti:
  - programsko kodo lahko pišemo v VB.NET, J++ ali C#
  - dobro razvojno okolje
  - Html koda je ločena od programske kode
- Slabost: deluje samo na Microsoftovih strežnikih (IIS)

## Orodja za svetovni splet

- ActiveX je Microsoftovo orodje za avtorje spletnih strani in neposredna konkurenca Javi. Z njim je mogoče spletne strani relativno enostavno opremiti z večpredstavnostjo ali programi.
- VRML (*Virtual Reality Modeling Language*) je programski jezik namenjen "3-D svetu" (3D tekstu in slikam, animacijam, zvoku...)

## Internetni brskalniki

- Na trgu je več kot 50 različnih brskalnikov.
- Najbolj popularni brskalniki so:



Microsoft Internet Explorer



Opera



SlipKnot



BeConn



Netscape Navigator

Firefox ...

## Orodja za izdelavo spletnih strani



Microsoft FrontPage



Agile HTML Editor



Coffee cup HTML Editor Pro



HotDog Professional



Ace Expert



Netscape Composer

## “Plug-in” programi

- Plug-in programi dodajo spletnim brskalnikom možnost ponujanja multimedijskih vsebin, tako da uporabnikom omogočijo gledanje novih vrst dokumentov in slik.
- Multimedijško podporo zagotavljajo tudi različni pomožni programi in predvajalniki, ki jih potrebujemo, če želimo pognati različne tipe datotek na internetu.

## “Plug-in” programi za tekstovne dokumente



Acrobat Reader\_ omogoča pregledovanje datotek formata .PDF. Fonti in grafika so pri Acrobatu vdelani v sam dokument kar pomeni, da se vedno ohrani originalni izgled dokumenta.



Word Viewer omogoča pregledovanje vseh Microsoft Wordovih dokumentov v brskalniku v originalnem formatu.

## “Plug-in” programi za slike in grafiko

- Večina brskalnikov podpira le grafične datoteke formatov JPEG in GIF. Za ogled vsebine datotek drugih formatov moramo namestiti ustrezen “plug-in” (vtičnik).
- Programi, ki omogočajo gledanje datotek vektorskih formatov so koristni, saj je slika v vektorskem formatu vedno prave velikosti in prikazana v maksimalnem številu barv, ki jih podpira računalnik.

## “Plug-in” programi za slike in grafiko



*Lightning Strike* zmanjša velikost bitnih slik, 10 do 100 krat bolj od GIF formata ter ponuja dosti večjo kvaliteto slik, kot JPEG format.



CMX omogoča prikaz Corel-ovih standardnih vektorskih formatov internetnem brskalniku. CMX datoteke so krajše ter jih lahko tako hitreje prenesemo prek interneta na naš računalnik.





InterCAP InLine podpira vektorski format CGM ter omogoča gledanje, povečavo in animacijo CGM datotek v brskalniku.




FreeHand Shockwave je program za prikaz in popraviljanje slik. Omogoča vstavljanje URL povezav v objekt posamezne slike.



ViewDirector omogoča gledanje in popraviljanje slik formatov TIFF G3 in G4, CALS Type 1, JPEG, PCX/DCX ter BMP.



Quick Vector podpira formata WMF in EMF in ima vse prednosti vektorske grafike.

 Autodesk Whip! je “plug-in”, ki omogoča prikaz slik v formatu DWF (Drawing Web Format). To je 2-D vektorski format, ki omogoča učinkovit prenos podatkov iz interneta na uporabnikov računalnik, ne da bi pri tem prišlo do sprememb na risbah.

## “Plug-in” programi za zvok

- Digitalizirane zvočne datoteke so na spletu na voljo v različnih formatih (WAV, AIF, AU, MIDI).
- Govorne datoteke je možno skrčiti ter jih z veliko hitrostjo poslati prek omrežja ter jih na drugi strani poslušati v prvotni obliki.



## “Plug-in” programi za zvok



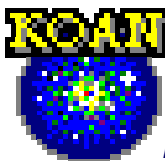
Crescendo je LiveUpdatov produkt, ki predvaja MIDI datoteke iz spletnih strani. Datoteka se začne predvajati takoj, ko začnemo dobivati podatke prek omrežja.

**eCHO**

EchoSpeech je dekorder, ki predvaja govorne datoteke skrčene 18,5 : 1. Te 16 bitne datoteke imajo vzorčenje 11025 Hz in so lahko predvajane v realnem času.



## “Plug-in” programi za zvok



Koan je program za predvajanje razpoloženske glasbe. Njegove datoteke so lahko manjše od 1KB pa vendar zadostujejo za do 8 ur glasbe. V datoteki so zapisani le osnovni parametri, samo glasbo pa ustvari računalnik. Tako nikoli ne moremo slišati dvakrat enake glasbe pa čeprav zaženemo isto datoteko.

## “Plug-in” programi za zvok



RealPlayer je zelo razširjen program, ki omogoča predvajanje zvoka in slike preko interneta v realnem času.



Talker je program za računalnike Macintosh, ki spremeni navadno tekstovno datoteko (spremenimo ji samo končnico) v govor.

## “Plug-in” programi za zvok



TrueSpeech je audio “plug-in” za predvajanje govora. Omogoča predvajanje prek interneta že pri prenosu 9600 bps.



Too/Vox predvaja govorne datoteke, pri kompresijskem razmerju 50:1 (15 sekund govora zasede približno 4.5 KB).

## “Plug-in” programi za video in animacijo

- Poznamo več vrst video formatov, v katerih so zapisani video posnetki. Najpogostejši formati so QuickTime, AVI in MPEG.
- Največja težava pri predvajanju videa in animacije preko spleta je hitrost prenosa podatkov, zato moramo doseči zadovoljiv kompromis med kvaliteto in količino podatkov.

## “Plug-in” programi za video in animacijo



ClearFusion je program za ogled datotek AVI formata.



MacZilla je “plug-in” za računalnike Machintosh, ki omogoča ogled filmov QuickTime, AVI in MPEG ter glasbe v formatih AU, WAV, MIDI, AIFF in MP2.

## “Plug-in” programi za video in animacijo



QuickTime proizvajalca Apple omogoča ogled filmov zapisanih v formatu QuickTime.



PreVU je “plug-in” za ogled videa zapisanega v formatu MPEG.

## “Plug-in” programi za video in animacijo



Shockwave je zgrajen iz treh delov (director, flash in authorware). Omogoča pregled predstavitev v brskalniku. Z njim lahko spletni strani dodamo animacije, zvok in video.



Sizzler omogoča ogled animacij in drugih multimedijskih datotek (format QuickTime pretvori v format SPRITE ali SPR).