

Tehnologii web

Suport de curs: 2009-2010

Autor: Lect. univ. dr. Horea Oros

Universitatea din Oradea

Departamentul de Matematică și Informatică

Tehnologii web

- n HTML,
 - n HTML, XHTML, CSS, TCP/IP
- n XML
 - n XML, XSL, XSL-FO, Xpath, XQuery, XLink, XPointer, DTD, SOAP, WSDL, RDF, RSS, WAP, Web Services,
- n Browser Scripting
 - n JavaScript, HTML DOM, DHTML, VBScript, AJAX
- n Server Scripting
 - n SQL, ASP, PHP, ADO

Ce este HTML?

- n HTML = **H**yper **T**ext **M**arkup **L**anguage
- n Un fișier HTML este un fișier text care conține anumite tag-uri pentru marcarea textului
- n Tag-urile informează browser-ul cum să afișeze pagina web
- n Fișierul HTML trebuie să aibă extensia **htm** sau **html**
- n Un fișier HTML poate fi creat folosind un simplu editor text (Notepad)

HTML “Hello world”

```
<html>
  <head>
    <title>Titlul paginii</title>
  </head>
  <body>
    Hello world
    <b>Acest text va fi afișat bold</b>
  </body>
</html>
```

Browser-e

- n Internet Explorer
- n Mozilla Firefox
- n Opera
- n Amaya
- n Netscape

Elemente HTML

- n Documentele HTML sunt fișiere text formate din **elemente HTML**
- n **Elementele HTML** sunt definite folosind tag-uri HTML
- n Tag-urile HTML
 - n Sunt folosite pentru a marca elementele HTML
 - n Sunt delimitate de `<` și `>`
 - n De regulă sunt în pereche `` și ``
 - n Textul dintre cele două tag-uri (de început și sfârșit) este **conținutul elementului**
 - n `` echivalent cu ``

Tag-uri cu litere mici sau mari?

n `` == ``?

n Răspuns:

n **NU** în XHTML obligatoriu litere mici

n HTML 4 **recomandă** utilizarea literelor mici

n Versiuni mai vechi **DA**

Tag-uri cu atribute

- n Tag-urile pot avea atribute – oferă o informație suplimentară în legătură cu un element
- n `<table>` - tabel HTML
- n `<table border="0">` - tabel HTML fără chenar
- n Atributele sunt perechi (nume, valoare)
- n Sintaxa: `nume = "valoare"`
- n Sunt specificate în tag-ul de start

Tag-uri cu attribute (cont.)

n XHTML

- n obligatoriu numele atributelor cu litere mici
- n Valorile trebuie delimitate cu “ sau ‘
- n Ex. 1 nume = “John”
- n Ex. 2 nume = 'John "ShotGun" Nelson'

Tag-uri elementare

<h1>Heading 1</h1>

<h2>Heading 2</h2>

<h3>Heading 3</h3>

<h4>Heading 4</h4>

<h5>Heading 5</h5>

<h6 align = "center">

Heading 6</h6>

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

Test

Tag-uri elementare

n Paragraf

`<p>Acesta e un paragraf</p>`

`<p>Acesta e un alt paragraf</p>`

n Rupere de linie

`<p>Acesta
 e un para
graf cu ruperi de linie</p>`

n Comentarii

`<!-- Acesta e un comentariu -->`

Taguri elementare

n `<hr>` - horizontal rule

n `<hr width="50%">`

n `<body bgcolor = "yellow">`

Test

Formatarea textului

Bold

Strong

<big>Big </big>

Text

<i>Italic</i>

<small>Small</small>

Acest text conține _{subscript}

Acest text conține ^{superscript}

Text preformatat

```
<pre>
```

```
Acesta
```

```
este text preformatat.
```

```
Se vor păstra atât spațiile suplimentare  
cât și ruperile de linie.
```

```
</pre>
```

```
<pre>
```

```
for i = 1 to 10
```

```
    print i
```

```
next i
```

```
</pre>
```

Prescurtări și acronime

```
<abbr title="United Nations">UN</abbr>
```

```
<acronym title="World Wide Web">WWW</acronym>
```

Atributul **title** este folosit pentru a afișa denumirea completă atunci când mouse-ul este deasupra elementului.

Entități HTML

- n Caracterele speciale nu pot fi folosite în text:
<, >
- n O entitate caracter are trei părți:
 - n ampersand (&)
 - n nume sau # și un număr de entitate
 - n ;
- n **<**; sau **<**; pentru <

Entități HTML

| | Descriere | Nume | Număr |
|---|--------------------|-------------------|--------------|
| | non-breaking space | | |
| < | Mai mic | < | < |
| > | Mai mare | > | > |
| & | Ampersand | & | & |
| " | Ghilimele | " | " |
| ' | apostrof | ' (nu in IE) | ' |

Entități HTML

| | Descriere | Nume | Număr |
|---|------------------|-------------|--------------|
| ¢ | cent | ¢ | ¢ |
| £ | pound | £ | £ |
| ¥ | yen | ¥ | ¥ |
| § | section | § | § |
| © | copyright | © | © |
| ® | registered | ® | ® |
| × | x | × | × |
| ÷ | / | ÷ | ÷ |

Legături <a href...>

- n În exterior, în același website, în aceeași pagină
 - n `Facultatea de Stiinte`
 - n `Pagina principala`
 - n ``

n a - anchor

Legături <a href...> (cont.)

- n Legături în aceeași pagină sau la o zonă marcată din altă pagină
 - n `Text` (În fișierul index.htm)
 - n `Click me!` (din alt fișier)
 - n `Click me!` (în fișierul index.htm)

Legături <a href...> (cont.)

n Atributul target

n <a href=<http://www.microsoft.com>
target="_blank">Microsoft

n target {"_blank", "_self", "_parent", "_top"}

n mailto link

n <a
href="mailto:someone@microsoft.com?subject
=Hello%20again">Send Mail

mailto link mai complicat

n <a

```
href="mailto:someone@microsoft.com?cc=someoneelse@microsoft.com&bcc=andsomeoneelse2@microsoft.com&subject=Summer%20Party&body=You%20are%20invited%20to%20a%20big%20summer%20party! ">Send mail!</a>
```

Ce urmează

- n Cadre (frame)
- n Inline frame
- n Tabele
- n Liste
- n Imagini
- n Formulare
- n Layout
- n meta

Tehnologii web

Curs 2

Conținuturi

- n Cadre (frame)
- n Inline frame
- n Tabele
- n Liste
- n Formulare
- n Imagini
- n Hărți de imagini
- n Layout
- n meta

Cadre (Frame)

```
<html>  
<frameset cols="25%,50%,25%">  
  <frame src="frame_a.htm">  
  <frame src="frame_b.htm">  
  <frame src="frame_c.htm">  
</frameset>  
</html>
```

Exemplu

Frameset

n Valorile pentru dimensiunea coloanelor/liniilor se pot preciza în pixeli sau procentual

n

cols="100,100" ===== două coloane de 100px

n cols="100,*" ===== * = restul

Frame-uri de navigare

```
<html>  
<frameset cols="120,*">  
<frame src="cuprins.htm">  
<frame src="frame_a.htm"  
name="showframe">  
</frameset>  
</html>
```

Exemplu cuprins

iframe (inline-frame)

```
<iframe src="http://stiinte.uoradea.ro"></iframe>
```

atribute: width, height, align, scrolling="yes, no sau auto" etc.

Exemplu iframe

Tabele

```
<table border="1">  
<tr>  
  <td>row 1, cell 1</td>  
  <td>row 1, cell 2</td>  
</tr>  
<tr>  
  <td>row 2, cell 1</td>  
  <td>row 2, cell 2</td>  
</tr>  
</table>
```

Tabele (cont.)

n Tabel cu antet: <th> în loc de <td>

n Celule goale:

n Tabel cu explicație

```
<table>
```

```
<caption>My Caption</caption>
```

```
<tr>...
```

n rowspan, colspan (atribute ale td)

n cellpadding, cellspacing, bgcolor, background (atribute al table)

Liste

- n Listă enumerată (type="disc", "circle", "square")

```
<ul>
```

```
<li>Coffee</li>
```

```
<li>Tea</li>
```

```
<li>Milk</li>
```

```
</ul>
```

n Coffee

n Tea

n Milk

- n Listă numerotată (type=A,a,l,i)

```
<ol>
```

```
<li>Coffee</li>
```

```
<li>Milk</li>
```

```
</ol>
```

1. Coffee

2. Milk

Formulare

- n Formularele se folosesc pentru a culege date de la utilizator
- n Un formular conține o serie de controale
 - n Casete de text, casete pentru parole
 - n Textarea
 - n Liste drop-down
 - n Butoane radio
 - n Casete de validare
 - n Butoane de apăsare, [Exemplu](#)

Formulare (cont.)

n Un formular este definit de tagul form

<form>

 <input>

 <input>

</form>

n <input> este cel mai uzual tag.

n Tipul de “input” este definit de atributul type

Formulare (cont.)

n Casete de text

```
<input type="text" name="firstname">
```

n Butoane radio

```
<form>
```

```
<input type="radio" name="SO" value="Windows"> Windows
```

```
<br>
```

```
<input type="radio" name="SO" value="Linux"> Linux
```

```
</form>
```

Formulare (cont.)

n Casete de validare

```
<form>
```

Sunt de acord cu condițiile:

```
<input type="checkbox" name="acord"  
value="DeAcord" />
```

```
</form>
```

Formulare (cont.)

n Atributul "action" și butoane submit

```
<form name="input"
```

```
  action="html_form_action.asp"
```

```
  method="get">
```

Username:

```
<input type="text" name="user">
```

```
<input type="submit" value="Submit">
```

```
</form>
```

Formulare (cont.) - Exemple

- n Casete de validare
- n Butoane radio
- n Lista dropdown
- n Textarea
- n Fieldset
- n Formular cu buton submit
- n Formular mailto

Imagini

- n Se introduce cu tag-ul `img`
 - n `` atributul `src` stabilește url-ul imaginii care va fi afișată
 - n `src = source`
 - n URL = Uniform Resource Locator
 - n Atributul `alt` definește un text alternativ pentru imagine
- ```

```

# Imagini (cont.) - Exemple

---

- n Exemplu
- n Imagine de background
- n Alinierea imaginilor în cadrul textului
- n Imagini float
- n Imagine ca hyperlink
- n Image map
- n Transformarea unei imagini in image map



# Ce urmează

---

n <head>

n <meta>

n Quiz

n XHTML

# Tehnologii web

Curs 3

# Conținuturi

---

n <head>

n <meta>

n Quiz

n XHTML

n CSS

# <head>

---

n Elementul head conține informație generală, numită și meta-informație, despre document

n <base><link><meta><title><style><script>

n <base>

n definește URL-ul de bază pentru toate link-urile de pe pagină

n Are atributul target <base target = “\_blank”>

# <head> (cont.)

---

## n <title>

- n Definește textul care apare în bara de titlu a browser-ului

## n <link>

- n Definește relația dintre două documente legate
- n `<link rel="stylesheet" type="text/css" href="theme.css" />`

# <meta>

---

- n Oferă meta-informație despre pagină
  - n Descriere,
  - n Cuvinte cheie ce pot fi folosite de motoare de căutare

```
<meta name="keywords" content="HTML,
DHTML, CSS, XML, XHTML, JavaScript,
VBScript" />
```

```
<meta name="description" content="Tutoriale
de HTML, CSS, XML, XHTML" />
```

## <meta> (cont.)

---

n <meta http-equiv="refresh" content="5" />

n **name** poate fi

n {author, description, keywords, generator, revised, altele}

n **http-equiv** poate fi

n {content-type, expires, refresh, set-cookie}

# <meta> (cont.) - Example

---

- n Descriere și cuvinte cheie în document
- n Redirectarea unui utilizator spre un alt site (în cazul schimbării adresei)



# Quiz

---

1. Ce înseamnă HTML?
  - a. Hyperlinks and Text Markup Language
  - b. Hyper Text Markup Language
  - c. Home Tool Markup Language

# Quiz

---

2. Cine face standardele WEB?
  - a. Microsoft
  - b. The World Wide Web Consortium
  - c. Netscape

# Quiz

---

3. Care este tag-ul corect pentru cel mai mare heading?
- a. `<h6>`
  - b. `<heading>`
  - c. `<h1>`
  - d. `<head>`

# Quiz

---

4. Cu ce tag se inserează o rupere de linie?
- a. `<br>`
  - b. `<break>`
  - c. `<lb>` (line break)

# Quiz

---

5. Cum se adaugă o culoare de fundal?
- a. `<background>yellow</background>`
  - b. `<body color="yellow">`
  - c. `<body bgcolor="yellow">`

# Quiz

---

6. Care este tag-ul HTML corect pentru a face un text îngroșat?
- a. `<b>`
  - b. `<bb>`
  - c. `<bld>`
  - d. `<bold>`

# Quiz

---

7. Care este tag-ul HTML corect pentru a face un text cursiv?
- a. `<ii>`
  - b. `<italics>`
  - c. `<i>`

# Quiz

---

8. Cum se creează un hyperlink?

- a. `<a name="http://www.w3.org">W3</a>`
- b. `<a url="http://www.w3.org">W3</a>`
- c. `<a>http://www.w3.org</a>`
- d. `<a href="http://www.w3.org">W3</a>`



# Quiz

---

9. Cum se poate crea un link pentru email?
- a. `<mail href="xxx@yyy">`
  - b. `<mail>xxx@yyy</mail>`
  - c. `<a href="xxx@yyy">`
  - d. `<a href="mailto:xxx@yyy">`

# Quiz

---

10. Cum se poate deschide un link într-o nouă fereastră de browser?

- a. `<a href="url" new>`
- b. `<a href="url" target="new">`
- c. `<a href="url" target="_blank">`

# Quiz

---

11. Care din tag-urile de mai jos aparțin toate de tabele?

- a. `<table><head><tfoot>`
- b. `<table><tr><td>`
- c. `<thead><body><tr>`
- d. `<table><tr><tt>`

# Quiz

---

12. Alegeți varianta corectă pentru alinierea conținutului unei celule a unui tabel la stânga.

- a. `<td valign="left">`
- b. `<td leftalign>`
- c. `<td align="left">`
- d. `<tdleft>`

# Quiz

---

13. Cum se poate crea o listă numerotată?

- a. `<list>`
- b. `<dl>`
- c. `<ol>`
- d. `<ul>`

# Quiz

---

14. Cum se poate crea o listă enumerată?

- a. `<ul>`
- b. `<ol>`
- c. `<dl>`
- d. `<list>`

# Quiz

---

15. Cum se crează o casetă de validare?

- a. `<input type="checkbox">`
- b. `<input type="check">`
- c. `<checkbox>`
- d. `<check>`

# Quiz

---

16. Cum se creează o casetă pentru introducerea de text?

- a. `<input type="textfield">`
- b. `<textinput type="text">`
- c. `<textfield>`
- d. `<input type="text">`



# Quiz

---

17. Cum se creează o listă drop-down?

- a. `<input type="list">`
- b. `<input type="dropdown">`
- c. `<select>`
- d. `<list>`

# Quiz

---

18. Cum se creează un textarea?

- a. `<input type="textbox">`
- b. `<textarea>`
- c. `<input type="textarea">`

# Quiz

---

19. Cum se inserează o imagine?

- a. `<img href="image.gif">`
- b. ``
- c. `<image src="image.gif">`
- d. `<img>image.gif</img>`

# Quiz

---

20. Cum se inserează o imagine de background?

- a. `<body background="background.gif">`
- b. `<background img="background.gif">`
- c. ``

# Ce urmează

---

n XHTML

n CSS

# XHTML

---

- n XHTML este o versiune mai strictă de HTML
- n XHTML înlocuiește HTML
- n XHTML aproape identic cu HTML 4.01
- n XHTML este HTML definit ca o aplicație XML
- n XHTML este recomandat a fi folosit de W3C  
(26 ian. 2000)

# XHTML (cont.)

---

n XHTML este o combinație între HTML și XML  
(**EX**tensible **M**arkup **L**anguage )

# De ce XHTML?

---

```
<html>
```

```
 <head>
```

```
 <title>This is bad HTML</title>
```

```
 <body>
```

```
 <h1>Bad HTML
```

```
 </body>
```



# De ce XHTML? (cont.)

---

- n XML este un limbaj de marcare în care totul trebuie să fie marcat corect, ceea ce duce la documente bine formate
- n XML a fost proiectat pentru a descrie date, HTML pentru a le prezenta
- n Codul care nu e bine format e greu de implementat pe dispozitive cu puteri limitate
- n Prin combinarea HTML cu XML obținem un limbaj util atât acum cât și în viitor

# XHTML != HTML

- n Pentru elementele XHTML trebuie folosit principiul parantezelor

Ex. `<b><i>Text bold si italic</b></i>`

- n Elementele XHTML trebuie întotdeauna închise

Ex. `<ul>`

`<li>Coffee</li>`

`<li>Tea`

`<ul>`

`<li>Black tea</li>`

`<li>Green tea</li>`

`</ul>`

`</li>`

`<li>Milk</li>`

`</ul>`

# XHTML != HTML

---

n Elementele fără conținut trebuie închise

Exemple:

Break:

```


```

Horizontal rule:

```
<hr />
```

Imagine:

```

```

# XHTML != HTML

---

- n Elementele XHTML trebuie scrise cu litere mici

`<p>Greșit`

`<p>Corect </p>`

- n Documentele XHTML trebuie să aibă un singur element root.

`<html>`

`<head> ... </head>`

`<body> ... </body>`

`</html>`

# Alte reguli de sintaxă XHTML

- n Numele atributelor trebuie să fie cu litere mici

`<table width="100%">`

- n Valorile atributelor trebuie puse între ghilimele

GREȘIT: `<table width=100%>`

- n Minimizarea este interzisă

GREȘIT:

`<input readonly>`

`<input checked>`

`<input disabled>`

`<option selected>`

`<frame noresize>`

CORECT:

`<input checked="checked" />`

`<input readonly="readonly" />`

`<input disabled="disabled" />`

`<option selected="selected" />`

`<frame noresize="noresize" />`

# Alte reguli de sintaxă XHTML (cont.)

---

n Atributul id înlocuiește atributul name

Greșit:

```

```

(doar pentru compatibilitate)

Corect:

```

```

# Reguli XHTML

---

n DOCTYPE obligatoriu

```
<!DOCTYPE tipul documentului>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
 <head>
```

```
 <title>Title goes here</title>
```

```
 </head>
```

```
 <body>
```

```
 </body>
```

```
</html>
```

# DOCTYPE

---

```
n <!DOCTYPE html PUBLIC
"-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

n Tipuri de documente

- n STRICT

- n TRANSITIONAL

- n FRAMESET



# DOCTYPE (cont.)

---

- n DTD specifică sintaxa paginii web în SGML
- n DTD este folosit de aplicații SGML, cum este HTML, pentru a specifica reguli care se aplică la marcarea documentelor de un anumit tip, inclusiv o mulțime de elemente și declarații de entități
- n XHTML este specificat într-o definiție de tip de document SGML
- n Un DTD XHTML descrie în mod precis, sintaxa și semantica permisă pentru XHTML

# DOCTYPE (cont.)

---

```
n <!DOCTYPE html PUBLIC
"-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
n <!DOCTYPE html PUBLIC
"-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
```

```
n <!DOCTYPE html PUBLIC
"-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
frameset.dtd">
```

# CSS

---

- n Ce este CSS
- n Sintaxa CSS
- n Exemple
- n CSS background
- n CSS text
- n CSS font
- n CSS border
- n CSS outline
- n CSS margin
- n CSS padding
- n CSS list
- n CSS table

# Ce este CSS

---

- n **CSS = Cascading Style Sheets**
- n Stilurile definesc modul în care sunt afișate elementele HTML
- n Stilurile sunt stocate în **foi de stiluri**
- n Au fost adăugate în HTML 4.0
- n Style Sheet extern duce la multă economie de muncă
- n Style Sheet extern stocat în fișier CSS
- n Mai multe definiții CSS se vor cascada una în alta

# CSS Demo

---

n CSS Demo

# CSS

---

- n Tag-urile HTML au fost proiectate pentru a defini conținutul unui document (paragrafe, titlu, liste, tabele etc.)
- n Modul lor de afișare era lăsat în seama browser-ului fără a folosi tag-uri pentru formatare
- n Adăugarea tag-urilor de formatare (ex. font) a îngreunat crearea de website-uri în care conținutul era separat de formatare

# CSS

---

- n W3C a creat standardul CSS o dată cu HTML 4.0
- n Majoritatea browser-elor suportă CSS
- n Stilurile definesc cum sunt prezentate elementele HTML
- n De regulă sunt puse în fișiere externe .css

# Cascadare CSS

---

- n Stilurile pot fi specificate în:
  - n Într-un element HTML
  - n În elementul head
  - n Într-un fișier extern
- n Un document HTML – mai multe fișiere CSS externe



# Ordinea de cascada

---

1. Browser (implicit, fără stil)
  2. Extern (fișier css)
  3. Intern (în cadrul <head>)
  4. Inline (în cadrul elementului)
- n Din cele 4 se va forma un stil virtual

# Sintaxa

---

selector {property: value}

Selector = element sau atribut html ce se definește

Property = atributul ce trebuie modificat

Ex.

```
body {color: black}
```

```
p {font-family: "sans serif"}
```

```
p {text-align:center;color:red}
```

# Sintaxa (cont.)

---

n Pentru lizibilitate:

```
p
{
 text-align: center;
 color: black;
 font-family: arial
}
```

n Pentru grupare

```
h1,h2,h3,h4,h5,h6
{
 color: green
}
```

# Selectorul class

---

p.right

{

text-align: right

}

p.center

{

text-align: center

}

```
<p class="right">
```

Paragraf aliniat la dreapta.

```
</p>
```

```
<p class="center">
```

Paragraf centrat.

```
</p>
```

# Selectorul class (cont.)

---

```
.center
{
 text-align: center
}
```

```
<h1 class="center">
 Titlu centrat
</h1>
<p class="center">
 Paragraf centrat.
</p>
```

# Sintaxa CSS

---

```
input[type="text"] {background-color: blue}
```

Background albastru la toate elementele input care au atributul "text"

# Selectorul id

---

```
#green {color: green}
```

Stil pentru elementul cu id-ul green

```
p#para1
```

```
{
```

```
 text-align: center; color: red
```

```
}
```

Stil pentru paragraful cu id-ul para1

# Comentarii CSS

---

```
/* Acesta este un comentariu */
p {
 text-align: center;
 /* Acesta este un alt comentariu */
 color: black; font-family: arial
}
```



# CSS extern - sintaxa

---

```
<head>
```

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

```
</head>
```

# CSS intern - sintaxa

---

```
<head>
<style type="text/css">
 hr {color: sienna}
 p {margin-left: 20px}
 body {background-image:
 url("images/back40.gif")}
</style>
</head>
```

# CSS inline - sintaxa

---

```
<p style="color: sienna; margin-left: 20px">
```

Acesta este un paragraf

```
</p>
```

Nu e indicat a fi folosit - se amestecă  
prezentarea cu formatarea

# Foi de stiluri multiple

---

CSS extern

```
h3
{
 color: red;
 text-align: left;
 font-size: 8pt
}
```

CSS intern

```
h3
{
 text-align: right;
 font-size: 20pt
}
```

Rezultatul:

```
color: red;
text-align: right;
font-size: 20pt
```

# Tehnologii web

Curs 4 - CSS

# Conținuturi

---

- n CSS background
- n CSS text
- n CSS font
- n CSS border
- n CSS margin
- n CSS padding

# CSS background

---

- n Proprietatea **background** definește efectul din fundal al unui element
- n Permite controlul: culorii, al unei imagini de fundal, repetarea unei imagini pe verticală și/sau orizontală și poziționarea unei imagini pe pagină

# CSS background - exemple

---

- n Stabilirea culorii de fundal pentru un element
- n Stabilirea unei imagini ca fundal
- n Repetarea unei imagini în fundal
- n Repetarea unei imagini în fundal vertical
- n Repetarea unei imagini în fundal orizontal
- n Afișarea unei imagini de fundal fără repetare
- n Poziționarea imaginii de fundal cu %
- n Poziționarea imaginii de fundal cu pixeli
- n Imagine de fundal fixată



# CSS background - proprietăți

---

- n background – proprietate prescurtată pentru a seta toate proprietățile de background într-o singură declarație
- n Valori:
  - n *background-color*
  - n *background-image*
  - n *background-repeat*
  - n *background-attachment*
  - n *background-position*

# CSS background - proprietăți

---

n background-attachment – stabilește dacă imaginea de fundal este fixată sau se mișcă cu restul paginii la scroll

n Valori

n scroll

n fixed

# CSS background - proprietăți

---

n background-color – stabilește culoarea de fundal a unui element

n Valori

n *color-rgb*

n *color-hex*

n *color-name*

n transparent

# CSS background - proprietăți

---

- n background-image – stabilește o imagine ca fundal pentru un element
- n Valori
  - n url(*URL*)
  - n none

# CSS background - proprietăți

---

- n background-position – stabilește poziția de start pentru imaginea din background
- n Valori
  - n top left
  - n top center
  - n top right
  - n center left
  - n center center
  - n center right
  - n bottom left
  - n bottom center
  - n bottom right
  - n *x% y%*
  - n *xpos ypos*

# CSS background - proprietăți

---

n background-repeat – stabilește cum se va repeta o imagine în fundal

n Valori

- n repeat
- n repeat-x
- n repeat-y
- n no-repeat

# CSS text

---

- n Proprietatea **text** definește modul în care este prezentat textul
- n Se poate
  - n Modifica culoarea,
  - n Schimba (mări/micșora) spațiul dintre litere
  - n Aliniere, decorare, indentare etc.

# CSS text - exemple

---

- n Stabilirea culorii textului
- n Stabilirea culorii de fundal pentru o parte din text
- n Mărirea/micșorarea spațiului dintre caractere
- n Stabilirea într-un paragraf a spațiului dintre linii
- n Alinierea textului
- n Decorarea textului
- n Indentarea textului
- n Capitalizarea literelor
- n Mărirea spațiului dintre cuvinte
- n Dezactivarea ruperii liniilor



# CSS text - proprietăți

---

n color {*color*}

n direction {ltr, rtl}

n line-height {normal, *number*, *length*, %}

n letter-spacing {normal, *length*}

n text-align {left, right, center, justify}

n text-decoration {none, underline, overline, line-through, blink}

# CSS text – proprietăți (cont.)

---

n text-indent {*length*, %}

n text-shadow {none, *color*, *length*}

n text-transform {none, capitalize, uppercase, lowercase}

n white-space {normal, pre, nowrap}

n word-spacing {normal, *length* }

# CSS font

---

- n Definesc font-ul folosit
- n Se poate modifica familia font-ului, grosimea, dimensiunea, stilul

# CSS font - exemple

---

- n Stabilirea font-ului unui text
- n Stabilirea font-ului cu valoarea “caption”
- n Stabilirea dimensiunii textului
- n Stabilirea dimensiunii textului cu font-size-adjust
- n Stabilirea stilului unui font
- n Stabilirea tipului font-ului (variant)
- n Stabilirea grosimii font-ului
- n Toate proprietățile font-ului într-o singură proprietate prescurtată

# CSS font - proprietăți

- n **font** pentru stabilire tuturor proprietăților
- n **font-family** {*family-name*, *generic-family*}
- n **font-size** {xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, *length*, % }
- n **font-size-adjust** {none, *number* }
- n **font-stretch** {normal, wider, narrower, ultra-condensed, extra-condensed, condensed, semi-condensed, semi-expanded, expanded, extra-expanded, ultra-expanded }
- n **font-style** {normal, italic, oblique }
- n **font-variant** {normal, small-caps}
- n **font-weight** {normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900}

# CSS border

---

- n Definește chenarul din jurul unui element
- n Permite stabilirea stilului și culorii chenarului unui element
- n De regulă în HTML se folosesc tabele pentru a pune chenare în jurul unor elemente.
- n CSS permite stabilirea de chenare cu diferite efecte pentru orice element

# CSS border - exemple

---

- n Toate proprietățile border într-o singura declarație
- n Chenare diferite pentru fiecare latură
- n Toate proprietățile border top într-o singură declarație
- n Proprietate prescurtată pentru stabilirea proprietăților părții de jos a chenarului
- n Proprietate prescurtată pentru stabilirea proprietăților părții din stânga a chenarului
- n Proprietate prescurtată pentru stabilirea proprietăților părții din dreapta a chenarului

# CSS border – exemple (cont.)

---

- n Stabilirea stilului celor 4 părți ale chenarului
- n Stilul părții de sus a chenarului
- n Stilul părții de jos a chenarului
- n Stilul părții din stânga a chenarului
- n Stilul părții din dreapta a chenarului
- n Toate grosimile celor 4 părți ale chenarului
- n Grosimea părții de sus a chenarului
- n Grosimea părții de jos a chenarului
- n Grosimea părții din stânga a chenarului
- n Grosimea părții din dreapta a chenarului



# CSS border – exemple (cont.)

---

- n Culoarea celor 4 părți ale chenarului
- n Culoarea părții de sus a chenarului
- n Culoarea părții de jos a chenarului
- n Culoarea părții din stânga a chenarului
- n Culoarea părții din dreapta a chenarului

# CSS border - proprietăți

- n **border** prescurtare pentru *border-width*, *border-style*, *border-color*
- n **border-bottom**
- n **border-bottom-color**
- n **border-bottom-style**
- n **border-bottom-width** {thin,medium, thick, *length*}
- n
- n **border-color**
- n
- n **border-left**
- n **border-left-color**
- n **border-left-style**
- n **border-left-width** {thin,medium, thick, *length*}

# CSS border - proprietăți

---

- n **border-right**
- n **border-right-color**
- n **border-right-style**
- n **border-right-width** {thin,medium, thick, *length*}
  
- n **border-style** {none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset }
- n
- n **border-top**
- n **border-top-color**
- n **border-top-style**
- n **border-top-width** {thin,medium, thick, *length*}
- n **border-width** {thin,medium, thick, *length*}

# CSS margin

---

- n Proprietatea **margin** definește spațiul dintre elemente
- n Se pot folosi valori negative pentru a suprapune conținut
- n Marginea de sus, jos, stânga, dreapta pot fi schimbate independent
- n Marginea implicită a elementului body este 8px (în IE și Netscape)

# CSS margin - exemple

---

- n Toate proprietățile margin într-o singură declarație
- n Stabilirea marginii de sus a textului folosind o valoare în cm
- n Stabilirea marginii de sus a textului folosind o valoare în %
- n Stabilirea marginii de jos a textului folosind o valoare în cm
- n Stabilirea marginii de jos a textului folosind o valoare în %

# CSS margin – exemple (cont.)

---

- n Stabilirea marginii din stânga textului folosind o valoare în cm
- n Stabilirea marginii din stânga textului folosind o valoare în %
- n Stabilirea marginii din dreapta textului folosind o valoare în cm
- n Stabilirea marginii din dreapta textului folosind o valoare în %

# CSS margin - proprietăți

---

- n **margin** {*margin-top, margin-right, margin-bottom, margin-left* }
- n **margin-bottom** {*auto, length, %*}
- n **margin-left** {*auto, length, %*}
- n **margin-right** {*auto, length, %*}
- n **margin-top** {*auto, length, %*}

# CSS padding

---

- n Proprietatea **padding** definește spațiul dintre chenarul unui element și conținutul său
- n Nu sunt permise valori negative
- n Cele 4 părți pot fi definite separat



# CSS padding - exemple

---

- n Toate proprietățile padding într-o singură declarație
- n Padding sus cu cm
- n Padding sus cu %
- n Padding jos cu cm
- n Padding jos cu %
- n Padding stânga cu cm
- n Padding stânga cu %
- n Padding dreapta cu cm
- n Padding dreapta cu %

# CSS padding - proprietăți

---

- n **padding** {*padding-top, padding-right, padding-bottom, padding-left*}
- n **padding-bottom** {*length, %*}
- n **padding-left** {*length, %*}
- n **padding-right** {*length, %*}
- n **padding-top** {*length, %*}

# Tehnologii web

Curs 5 – CSS (cont.)

# Conținuturi

---

- n CSS list
- n CSS table
- n CSS dimensiuni
- n CSS clasificare
- n CSS poziționare
- n CSS pseudo-clase
- n CSS pseudo-elemente
- n CSS galerie de imagini
- n CSS image opacity
- n CSS media
- n QUIZ

# CSS list

---

- n Proprietățile **list** permit plasarea marcatorilor de liste, schimbarea marcatorului sau folosirea unei imagini ca marcator

# CSS list - exemple

---

- n Diferiți marcatori pentru elementele unei liste enumerate
- n Diferiți marcatori pentru elementele unei liste numerotate
- n Toți marcatorii pentru elemente de listă
- n Imagine ca marcator pentru element de listă
- n Plasarea marcatorului
- n Toate proprietățile pentru liste într-o singură declarație

# CSS list - proprietăți

---

- n **list-style** {*list-style-type*, *list-style-position*, *list-style-image*}
- n **list-style-image** {none, *url*}
- n **list-style-position** {inside, outside}
- n **list-style-type** {none, disc, circle, square, decimal, decimal-leading-zero, lower-roman, upper-roman, lower-alpha, upper-alpha, lower-greek, lower-latin, upper-latin, hebrew, armenian, georgian, cjk-ideographic, hiragana, katakana, hiragana-iroha, katakana-iroha}
- n **marker-offset** {auto, *length* }

# CSS table

---

- n Proprietatea **table** permite stabilirea layout-ului unui tabel



# CSS table - exemple

---

- n Stabilirea layout-ului unui tabel
- n Afişarea celulelor goale dintr-un tabel (MAC IE)
- n Chenar simplu sau detaşat ca în HTML
- n Spaţiul dintre celulele tabelului (MAC IE)
- n Poziţia etichetei tabelului (MAC IE)

# CSS table - proprietăți

---

- n **border-collapse** {collapse, separate}
- n **border-spacing** [fără IE Win] {*length length*}
- n **caption-side** [fără IE Win] {top, bottom, left, right}
- n **empty-cells** [fără IE Win] {show, hide}
- n **table-layout** {auto, fixed}

# CSS dimensiuni

---

- n Proprietățile pentru dimensiuni permit controlul lungimii și lățimii elementelor
- n Permite mărirea spațiului dintre două linii

# CSS dimensiuni - exemple

---

- n Stabilirea înălțimii unui imagini în px
- n Stabilirea înălțimii unui imagini în %
- n Stabilirea lățimii unui element în px
- n Stabilirea lățimii unui element în %
- n **FĂRĂ IE**
  - n Stabilirea înălțimii maxime a unui element
  - n Stabilirea lățimii maxime a unui element în px
  - n Stabilirea lățimii maxime a unui element în %
  - n Stabilirea înălțime minimă a unui element
  - n Stabilirea lățime minimă a unui element în px
  - n Stabilirea lățime minime a unui element în %

# CSS dimensiuni - exemple

---

- n Spațiul dintre linii – %
- n Spațiul dintre linii – px
- n Spațiul dintre linii – număr

# CSS dimensiune - proprietăți

---

n **height** {auto, *length*, %}

n **line-height** {normal, *number*, *length*, %}

n **max-height** {none, *length*, %}

n **max-width** {none, *length*, %}

n **min-height** {*length*, %}

n **min-width** {*length*, %}

n **width** {auto, %, *length*}

# CSS clasificare

---

- n Proprietățile de clasificare permit stabilirea modului și a locului în care sunt afișate elementele
- n Se poate stabili unde va apărea o imagine în cadrul altui element
- n Poziționarea unui element relativ la poziția sa normală
- n Poziționarea unui element la o valoare absolută
- n Controlarea vizibilității unui element

# CSS clasificare - exemple

---

- n Afișarea inline a unui element
- n Afișarea unui element ca bloc
- n O utilizare simplă a proprietății float
- n O imagine cu chenar și margini care se plasează la dreapta paragrafului
- n O imagine cu etichetă care se plasează la dreapta paragrafului
- n Prima literă a paragrafului plasată la dreapta
- n Crearea unui meniu orizontal



# CSS clasificare – exemple (cont.)

---

- n Crearea unui homepage fără tabele
- n Poziționarea relativă a unui element
- n Poziționarea absolută a unui element
- n Element invizibil
- n Ascunderea unui tabel
- n Schimbarea cursorului
- n Ștergerea părților unui element float

# CSS clasificare - proprietăți

---

- n **clear** {left, right, both, none} stabilește părțile unui element unde nu sunt permise alte elemente float
- n **cursor** {*url*, auto, crosshair, default, pointer, move, e-resize, ne-resize, nw-resize, n-resize, se-resize, sw-resize, s-resize, w-resize, text, wait, help}
- n **display** {none, inline, block, list-item, run-in, compact, marker, table, inline-table, table-row-group, table-header-group, table-footer-group, table-row, table-column-group, table-column, table-cell, table-caption}

# CSS clasificare - proprietăți

---

n **float** {left, right, none} stabilește unde va apărea o imagine sau un text în cadrul altui element

n **position** {static, relative, absolute, fixed}

n **visibility** {visible, hidden, collapse}

# CSS poziționare

---

- n Proprietăți ce permit poziționarea elementelor
- n Permit specificarea poziția stânga, dreapta, sus, jos a unui element
- n Permite specificarea formei unui element
- n Plasarea unui element în spatele altuia
- n Stabilirea a ceea ce trebuie să se întâmple când un element nu încapă în zona specificată

# CSS poziționare - exemple

---

- n Position:relative
- n Position:absolute
- n Position:fixed
- n Stabilirea formei unui element
- n Stabilire proprietății overflow – cu scroll
- n Stabilire proprietății overflow – cu ascundere
- n Overflow tratat automat de browser
- n Alinierea pe verticală a unei imagini
- n Z-index (1)
- n Z-index (2)

# CSS poziționare – exemple (cont.)

---

- n Stabilirea părții de sus a unei imagini – px
- n Stabilirea părții de sus a unei imagini - %
- n Stabilirea părții de jos a unei imagini – px
- n Stabilirea părții de jos a unei imagini - %
- n Stabilirea părții din stânga a unei imagini – px
- n Stabilirea părții din stânga a unei imagini - %
- n Stabilirea părții din dreapta a unei imagini – px
- n Stabilirea părții din dreapta a unei imagini - %

# CSS poziționare - proprietăți

---

- n bottom {auto, %, *length*}
- n clip {*shape*, auto}
- n left {auto, %, *length*}
- n overflow {visible, hidden, scroll, auto}
- n position {static, relative, absolute, fixed}
- n right {auto, %, *length*}
- n top {auto, %, *length*}
- n vertical-align {baseline, sub, super, top, text-top, middle, bottom, text-bottom, *length*, %}
- n z-index {auto, *number*}

# CSS pseudo-clase

---

- n Folosite pentru a adăuga efecte anumitor selectori
- n Sintaxa:
  - n selector:pseudo-clase {property: value}
  - n SAU
  - n selector.class:pseudo-clase {property: value}



# CSS pseudo-clase - exemple

---

- n Hyperlink
- n Hyperlink 2
- n Hyperlink: cu :focus (fără IE)
- n :first-child (IE numai dacă este declarat <!DOCTYPE>)
- n :lang (fără IE)

# CSS pseudo-class – proprietăți

---

n :active

n :focus

n :hover

n :link

n :visited

n :first-child

n :lang

# CSS pseudo-elemente

---

- n Pseudo-elementele sunt folosite pentru a adăuga efecte speciale unor selectori
- n Sintaxa:
  - n `selector:pseudo-element {property: value}`
  - n `selector.class:pseudo-element {property: value}`

# CSS pseudo-elemente - exemple

---

- n Prima literă cu efect special
- n Prima linie cu efect special
- n Prima literă și prima linie cu efect special
- n :before pentru a adăuga conținut în fața unui element
- n :after pentru a adăuga conținut după un element

# CSS pseudo-elemente - proprietăți

---

- n **:first-line** poate fi folosit doar cu elemente block
  - n font
  - n color
  - n background
  - n word-spacing
  - n letter-spacing
  - n text-decoration
  - n vertical-align
  - n text-transform
  - n line-height
  - n clear

# CSS pseudo-elemente - proprietăți

---

- n **:first-letter** adaugă un stil special primei litere din text
  - n font
  - n color
  - n background
  - n margin
  - n padding
  - n border
  - n text-decoration
  - n vertical-align (doar dacă 'float' este 'none')
  - n text-transform
  - n line-height
  - n float
  - n clear

# CSS pseudo-elemente - proprietăți

---

n :before folosit pentru a adăuga conținut în fața unui element

n Ex.

```
h1:before
{
 content: url(beep.wav)
}
```

# CSS pseudo-elemente - proprietăți

---

n :after folosit pentru a adăuga conținut după un element

n Ex.

```
h1:after
{
 content: url(beep.wav)
}
```



# CSS pseudo-elemente și clase

---

Exemplu:

```
p.articol: first-letter {color:#ff0000}
```

```
<p class="articol">Un paragraf din articol</p>
```

# CSS galerie de imagini

---

n Galerie de imagini creată cu CSS

# CSS image opacity

---

- n Crearea de imagini transparente cu efect la mouseover
- n Un careu transparent cu text pe o imagine de fundal

# CSS2 Media

---

- n @media permite specificarea modului în care vor fi prezentate documentele în medii diferite
- n Documentul poate fi prezentat diferit pe ecran, hârtie etc.

# CSS2 Media - exemplu

```
<html> <head>
<style>
@media screen {
 p.test {
 font-family:verdana,sans-serif; font-size:14px
 }
}
@media print {
 p.test {font-family:times,serif; font-size:10px}
}
@media screen, print {
 p.test {font-weight:bold}
}
</style> </head>
<body>....</body> </html>
```

# CSS2 Media - tipuri

---

n all

n aural

n braille

n embossed

n handheld

n print

n projection

n screen

n tty

n tv

# CSS Quiz

---

1. Ce reprezintă CSS?
  - a. Cascading Style Sheets
  - b. Creative Style Sheets
  - c. Computer Style Sheets
  - d. Colorful Style Sheets

# CSS Quiz

---

2. Cum ne putem referi în HTML la un CSS extern?

- a. `<link rel="stylesheet" type="text/css" href="mystyle.css">`
- b. `<stylesheet>mystyle.css</stylesheet>`
- c. `<style src="mystyle.css">`



# CSS Quiz

---

3. Unde trebuie pusă referința la un CSS extern în cadrul unei pagini HTML?
  - a. În secțiunea <head>
  - b. La sfârșitul documentului
  - c. La începutul documentului
  - d. În secțiunea <body>

# CSS Quiz

---

4. Care element HTML este folosit pentru a defini un style sheet intern?
- a. `<CSS>`
  - b. `<style>`
  - c. `<script>`

# CSS Quiz

---

5. Care atribut HTML este folosit pentru a defini un style sheet inline?
- a. font
  - b. styles
  - c. style
  - d. class

# CSS Quiz

---

6. Care este sintaxă corectă CSS?
- a. `body:color=black`
  - b. `body {color: black}`
  - c. `{body;color:black}`
  - d. `{body:color=black(body)}`

# CSS Quiz

---

7. Cum se introduc comentarii în fișiere CSS?
- a. // acesta e un comentariu //
  - b. ‘ acesta e un comentariu
  - c. // acesta e un comentariu
  - d. /\* acesta e un comentariu \*/

# CSS Quiz

---

8. Care proprietate se folosește pentru a schimba culoarea de fundal?
- a. color:
  - b. background-color:
  - c. bgcolor:

# CSS Quiz

---

9. Cum se adaugă o culoare de fundal pentru toate elementele h1?
- a. `h1.all {background-color:#FFFFFF}`
  - b. `all.h1 {background-color:#FFFFFF}`
  - c. `h1 {background-color:#FFFFFF}`

# CSS Quiz

---

10. Cum se schimbă culoarea textului unui element?

- a. `text-color=`
- b. `text-color:`
- c. `fgcolor:`
- d. `color:`



# CSS Quiz

---

11. Care proprietate CSS stabilește dimensiunea textului?

- a. text-size
- b. text-style
- c. font-style
- d. font-size

# CSS Quiz

---

12. Care este sintaxa CSS corectă pentru ca toate elementele p să aibă textul bold?

- a. `p {font-weight:bold}`
- b. `<p style="text-size:bold">`
- c. `p {text-size:bold}`
- d. `<p style="font-size:bold">`

# CSS Quiz

---

13. Cum se afișează hyperlink-uri fără linie de subliniere?

- a. `a {text-decoration:none}`
- b. `a {decoration:no underline}`
- c. `a {text-decoration:no underline}`
- d. `a {underline:none}`

# CSS Quiz

---

14. Cum se face ca fiecare cuvânt dintr-un text să înceapă cu literă mare?
- a. `text-transform:capitalize`
  - b. Nu se poate face cu CSS
  - c. `text-transform:uppercase`

# CSS Quiz

---

15. Cum se schimbă font-ul unui element?

- a. font-family:
- b. font=
- c. f:

# CSS Quiz

---

16. Cum se face textul bold?

- a. `font-weight:bold`
- b. `font:b`
- c. `style:bold`

# CSS Quiz

---

17. Cum se afișează un chenar de tipul

top border = 10 pixel

bottom border = 5 pixel

left border = 20 pixel

right border = 1 pixel?

- a. border-width:10px 5px 20px 1px
- b. border-width:10px 20px 5px 1px
- c. border-width:10px 1px 5px 20px
- d. border-width:5px 20px 10px 1px

# CSS Quiz

---

18. Cum se modifică marginea din stânga a unui element?

- a. `text-indent:`
- b. `indent:`
- c. `margin:`
- d. `margin-left:`



# CSS Quiz

---

19. Pentru a defini spațiul dintre chenarul unui element și conținut se folosește proprietatea padding. Se pot folosi pentru această proprietate valori negative?
- a. Da
  - b. Nu

# CSS Quiz

---

20. Cum se stabilesc marcatorii unei liste la pătrate?

- a. type: square
- b. list-style-type: square
- c. type: 2
- d. list-type: square

# Tehnologii web

Curs 6 – JavaScript

# Conținuturi

---

- n Introducere
- n Variabile
- n If... Else
- n Switch
- n Operatori
- n Casete popup
- n Funcții
- n For
- n While
- n Break
- n For...in
- n Evenimente
- n Try...catch
- n Throw
- n Onerror
- n Caractere speciale

# Introducere

---

- n JavaScript este limbajul script folosit cel mai frecvent pe web
- n Folosit pentru
  - n A îmbunătăți designul
  - n A valida formulare
  - n Detecta browserul
  - n Crearea de cookie
  - n etc.

# Ce este JavaScript

---

- n JavaScript a fost proiectat pentru a adăuga interactivitate în paginile HTML
- n Un limbaj script este un limbaj de programare
- n Un script constă din linii de cod executate una după alta
- n Este inclus direct în paginile HTML
- n JavaScript este un limbaj interpretat (programele sunt executate fără a fi compilate în prealabil)
- n Nu e nevoie de licență pentru a folosi JavaScript (poate fi folosit de oricine)

# Java vs. JavaScript

---

- n Sunt total diferite (nu au nici o legătură)
- n JavaScript dezvoltat de Netscape
- n Java dezvoltat de Sun Microsystems – limbaj mult mai puternic din aceeași categorie cu C++, C#

# Ce se poate face cu JavaScript?

---

- n **JavaScript oferă designerilor HTML o unealtă de programare** – autorii HTML nu sunt de regulă programatori, dar JavaScript este un limbaj script cu o sintaxă foarte simplă. Aproape oricine poate adăuga secvențe de cod mici în paginile HTML
- n **JavaScript poate pune text dinamic în paginile HTML** – O instrucțiune JavaScript de tipul:  
`document.write("<h1>" + name + "</h1>")` poate scrie valoarea unei variabile într-o pagină HTML
- n **JavaScript poate reacționa la evenimente** – un program JavaScript se poate executa când se întâmplă ceva: pagina s-a încărcat complet, se dă click pe un element HTML



# Ce se poate face cu JavaScript?

---

- n **JavaScript poate citi și scrie conținutul elementelor HTML**
- n **JavaScript poate fi folosit pentru a valida date** – Un program JavaScript poate valida datele dintr-un formular înainte de a fi trimise la server. Aceasta scutește serverul de o procesare suplimentară
- n **JavaScript poate detecta browserul folosit de “internaut”** – cu scopul de a încărca o pagină creată special pentru acel browser.
- n **JavaScript poate fi folosit pentru a crea cookies** – poate stoca și citi informații de pe calculatorul “internautului”

# JavaScript – cum se face

---

- n Elementul HTML <script> este folosit pentru a insera cod JavaScript într-o pagină web
- n Exemple:
  - n Scrierea de text pe o pagină
  - n Formatarea textului pe pagină cu elemente HTML

# JavaScript în HTML

---

```
<html>
```

```
<body>
```

```
<script type="text/javascript">
 document.write("Hello World!")
```

```
</script>
```

```
</body>
```

```
</html>
```

# Browsere care nu înțeleg JavaScript

---

```
<script type="text/javascript">
<!--
document.write("Hello World!")
//-->
</script>
```

# Unde se pune codul?

---

- n În body - [exemplu](#)
  - n Se execută când pagina se încarcă
- n În head - [exemplu](#)
  - n Se execută când este apelat
- n Extern – [exemplu](#) pentru ca scriptul să fie folosit în mai multe pagini
  - n Fișier extern cu extensia .js
- n Se pot plasa oricâte scripturi într-o pagină web

# Exemplu – head, body

---

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

 </script>
</head>
<body>
 <script type="text/javascript">

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

# Exemplu - extern

---

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

# Variabile

---

- n Variabilele sunt folosite pentru a stoca date
- n Reprezintă un container pentru informația ce vrem să o stocăm
- n Valoarea unei variabile se poate schimba în timpul execuției unui script
- n Numele de variabile sunt case sensitive (strname != STRNAME)



# Variabile

---

n Declarație:

n var strname = *valoare*

n strname = *valoare*

n Exemplu

n Vizibilitate:

n Local (declarate în funcție)

n Global (declarate în afara funcțiilor – vizibile după declarare în toate funcțiile)

# if ... else

---

n Exemplu 1

n Exemplu 2

n Exemplu 3

n Exemplu 4

n Sintaxa:

```
if (condiție)
```

```
{
```

```
 Cod care se execută când condiția este true
```

```
}
```

# Exemplu if

---

```
<script type="text/javascript">
// Scribe "Buna dimineata" daca ora este mai
// mica decât 10
var d=new Date()
var time=d.getHours()
if (time<10)
{
 document.write("Buna dimineata")
}
</script>
```

# Exemplu if

---

```
<script type="text/javascript">
//Scrie ora mesei dacă ora este 13
var d=new Date()
var time=d.getHours()
if (time==13)
{
 document.write("Ora mesei")
}
</script>
```

# Exemplu if...else

```
<script type="text/javascript">
//dacă ora este înainte 10 se va afișa "Buna dimineata"
//altfel "Buna ziua".
var d = new Date()
var time = d.getHours()
if (time < 10)
{
 document.write("Buna dimineata")
}
else
{
 document.write("Buna ziua")
}
</script>
```

# switch

n Exemplu

n Sintaxa:

```
switch(n)
```

```
{
```

```
case 1:
```

```
 Se execută blocul de cod 1
```

```
 break
```

```
case 2:
```

```
 Se execută blocul de cod 2
```

```
 break
```

```
default:
```

```
 cod care se execută dacă n este diferit de 1 și 2
```

```
}
```

# Operatori

## n Aritmetici:

n +, -, \*, /, %, ++, --

## n Atribuire:

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

## n Relaționali:

n == (compară valoarea)

n === (compară valoarea și tipul)

n x=5

n y= "5"

n x==y (true)

n x===y (false)

n <, >, <=, >=

# Operatori

---

n Logici:

n &&, ||, !

n Operatori pe șiruri:

n txt1="What a very"

n txt2="nice day!"

n txt3=txt1 + " " + txt2

n Condițional:

n numevariabila=(condiție)?valoare1:valoare2



# Casetă pop-up

---

- n Alert box (OK)
- n Alert box cu rupere de linii
- n Casetă de confirmare (OK, Cancel)
- n Casetă prompt (permite introducerea de text)

# Funcții

---

- n Funcția este o secvență reutilizabilă de cod care este executată când are loc un eveniment sau când este apelată
- n Pentru ca scriptul să nu fie executat când se încarcă pagina se poate pune într-o funcție
- n Funcția poate fi apelată de oriunde din pagină sau din alte pagini dacă este inclusă într-un fișier .js
- n E indicat ca funcțiile să fie puse în secțiunea head pt. a fi încărcate

# Funcții - exemplu

---

```
<html> <head>
<script type="text/javascript">
function displaymessage()
{
alert("Hello World!")
}
</script>
</head>
<body>
<form>
<input type="button" value="Click!" onclick="displaymessage()" >
</form>
</body> </html>
```

# Funcții - sintaxa

---

Cu parametri

```
function numelefuncției (var1, var2, ..., varX)
{
 cod
}
```

Fără parametri

```
function numelefuncției ()
{
 cod
}
```

# Funcții - return

---

```
function prod(a,b)
{
 x=a*b
 return x
}
```

Apel:

```
product=prod(2,3)
```

# Funcții exemple

---

- n Cum se apelează o funcție
- n Transmiterea de parametri funcțiilor
- n Transmiterea de parametri funcțiilor 2
- n Întoarcerea unei valori dintr-o funcție
- n Funcție cu argumente care întoarce o valoare

# Bucula for

---

n Buclele sunt folosite pentru a executa aceeași secvență de cod de mai multe ori

n Sintaxa:

```
for(var=start;var<=end;var=var+increment)
{
 cod
}
```

Exemplu

# Bucla for - exemplu

---

```
<html> <body>
<script type="text/javascript">
var i=0
for (i=0;i<=10;i++)
{
 document.write("The number is " + i)
 document.write("
")
}
</script>
</body> </html>
```



# Bucula while

n Se folosește atunci când vrem să executăm o secvență de cod atâta timp cât o condiție este adevărată (se poate ca să nu se execute niciodată)

n Sintaxa:

```
while (var<=endvalue)
{
 cod
}
```

Exemplu

# Bucla while - exemplu

---

```
<html> <body>
<script type="text/javascript">
var i=0
while (i<=10)
{
 document.write("The number is " + i)
 document.write("
")
 i=i+1
}
</script>
</body> </html>
```

# Bucula do...while

n Se folosește atunci când vrem să executăm o secvență de cod atâta timp cât o condiție este adevărată (se execută cel puțin o dată indiferent de valoarea condiției la început)

n Sintaxa:

```
do
{
 cod
} while (var<=endvalue)
```

Exemplu

# Bucla do...while - exemplu

---

```
<html> <body>
<script type="text/javascript">
var i=0
do
{
 document.write("The number is " + i)
 document.write("
")
 i=i+1
} while (i<=5)
</script>
</body> </html>
```

# break și continue

---

- n **break** oprește execuția unei bucle și predă controlul execuției instrucțiunii de după buclă
- n **continue** oprește execuția buclei curente și trece la reevalua condiției

# break - exemplu

---

```
<html> <body>
<script type="text/javascript">
var i=0
for (i=0;i<=10;i++)
{
 if (i==3)
 {
 break
 }
 document.write("The number is " + i)
 document.write("
")
}
</script> </body> </html>
```

# continue - exemplu

---

```
<html> <body>
<script type="text/javascript">
var i=0
for (i=0;i<=10;i++)
{
 if (i==3)
 {
 continue
 }
 document.write("The number is " + i)
 document.write("
")
}
</script>
</body> </html>
```

# for..in

---

- n Folosită pentru a itera peste elementele unui tablou sau peste proprietățile unui obiect
- n Sintaxa:

```
for (variabila in obiect)
{
 cod
}
```



# for...in - exemplu

---

```
<html> <body>
<script type="text/javascript">
var x
var mycars = new Array()
mycars[0] = "Saab"
mycars[1] = "Volvo"
mycars[2] = "BMW"
for (x in mycars){
 document.write(mycars[x] + "
")
}
</script>
</body> </html>
```

# Evenimente

---

- n Evenimentele sunt acțiuni ce pot fi detectate de JavaScript
- n Folosind JavaScript avem abilitatea de crea pagini web dinamice
- n Fiecare element al unei pagini web are anumite evenimente ce pot lansa anumite funcții
- n Exemplu: evenimentul **onClick** al unui buton

# Exemple de evenimente

---

- n Mouse click
- n Încărcarea unei pagini web sau a unei imagini
- n Deplasarea mouse-ului deasupra unui element al paginii web
- n Selectarea unei casete pentru introducerea de text într-un formular HTML
- n Transmiterea unui formular (Submit)
- n Apăsarea unei taste

# onload și onUnload

---

- n Evenimente lansate când se intră pe o pagină sau când este părăsită
- n Evenimentul onload este folosit pentru a verifica ce tip de browser este folosit, versiunea acestuia pentru a putea încărca pagina corespunzătoare
- n Folosite și pentru a seta cookie

# onFocus, onBlur și onChange

---

n Evenimente folosite în combinație cu validarea câmpurilor formularelor

n Exemplu:

```
<input type="text" size="30" id="email"
 onchange="checkEmail()">;
```

# onSubmit

---

- n Folosit pentru a valida toate câmpurile unui formular înainte de a fi trimis
- n Exemplu:

```
<form method="post" action="xxx.htm"
 onsubmit="return checkForm()">
```

Dacă funcția *checkForm* întoarce *false* atunci transmiterea datelor este anulată.

Datele sunt transmise doar dacă funcția întoarce *true*.

# onMouseOver și onMouseOut

---

n Folosite adesea pentru a crea butoane  
“animate”

n Exemplu:

```
<a href="http://ex.org"
onmouseover="alert('Hello');return false">


```

# try...catch

---

- n Blocul try...catch permite testarea erorilor din din cod
- n În script-uri pot să apară erori pe care browserul le prezintă
- n Sunt utile pentru dezvoltator dar nu și pentru cel ce navighează
- n Mecanismul permite separarea codului ce reprezintă logica aplicației de codul de tratare a erorilor ce pot să apară



# try...catch

---

n Sintaxa:

```
try
{
 //Aici este codul
}
catch(err) { //Tratarea erorii }
```

# try...catch – exemplu eronat

---

```
<html> <head>
<script type="text/javascript">
function message()
{
 adddalert("Welcome guest!") /*eroare*/
}
</script>
</head>
<body>
<input type="button" value="View message"
 onclick="message()" />
</body> </html>
```

# try...catch – exemplu corectat

```
<html> <head>
<script type="text/javascript">
var txt=""
function message() {
 try {
 addlert("Welcome guest!") /*numele functiei gresit*/
 } catch(err) {
 txt="There was an error on this page.\n\n"
 txt+="Error description: " + err.description + "\n\n"
 txt+="Click OK to continue.\n\n"
 alert(txt)
 }
}
</script> </head>
<body> <input type="button" value="View message"
 onclick="message()" />
</body> </html>
```

# try...catch – exemple

---

n try... catch 1

n try... catch 2

# throw

---

- n Permite crearea unei excepții
- n Împreună cu try...catch permite controlul execuției scriptului și generarea unor mesaje de eroare adecvate
- n Sintaxa:
  - n `throw(excepție)`
- n Excepție: string, întreg, boolean, obiect

# throw - exemplu

```
<html> <body> <script type="text/javascript">
var x=prompt("Enter a number between 0 and 10:", "")
try {
 if(x>10)
 throw "Err1"
 else if(x<0)
 throw "Err2"
} catch(er) {
 if(er=="Err1")
 alert("Error! The value is too high")
 if(er == "Err2")
 alert("Error! The value is too low")
}
</script> </body> </html>
```

# Evenimentul onerror

---

n Soluție mai veche pentru tratarea erorilor

n Exemplu

n Evenimentul este lansat când are loc o eroare în script

n Sintaxa:

```
onerror=handleErr
```

```
function handleErr(msg, url, l)
```

```
{
```

```
 //Tratarea erorii și return true (browser-ul nu afișează
```

```
 // mesajul standard de eroare) sau false
```

```
}
```

# Carattere speciale

---

- n \' apostrof
- n \" ghilimele
- n \& ampersand
- n \\ backslash
- n \n new line
- n \r carriage return
- n \t tab
- n \b backspace
- n \f form feed



# Tehnologii web

Curs 7 – JavaScript

# Conținuturi

---

- n Obiecte JavaScript
- n Obiectul String
- n Obiectul Date
- n Tablouri
- n Obiectul Boolean
- n Obiectul Math
- n HTML DOM
- n Detectarea browserului
- n Cookies
- n Validarea formularelor

# Obiecte JavaScript

---

n JavaScript este un limbaj OOP

n Obiect = proprietăți + metode

n Exemplu:

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

```
 var txt="Hello World!"
```

```
 document.write(txt.length) /*Proprietate*/
```

```
</script>
```

# Obiecte JavaScript

---

n Metodele sunt acțiunile ce pot fi efectuate cu ajutorul obiectelor

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

```
 var str="Hello world!"
```

```
 document.write(str.toUpperCase()) /*metodă*/
```

```
</script>
```

# Obiectul String

---

- n Folosit pentru a lucra cu șiruri de caractere
- n Exemple:
  - n Lungimea unui șir
  - n Stilizarea șirurilor de caractere
  - n indexOf()
  - n match()
  - n replace()

# Obiectul Date

---

n Folosit pentru a lucra cu ora și data

```
var d=new Date()
```

```
d.setFullYear(2010)
```

```
d.setDate(d.getDate()+5) /*5 zile*/
```

```
/*ajustarea lunii și a anului se face automat*/
```

# Obiectul Date

---

Comparare:

```
var d=new Date()
d.setFullYear(2010,0,14)
var today = new Date()
if (d>today)
 alert("Azi este inainte de 14 Ianuarie 2010")
else
 alert(" Azi este dupa 14 Ianuarie 2010 ")
```

# Obiectul Date - exemplu

---

- n Date()
- n getTime()
- n setFullYear()
- n toUTCString()
- n getDay()
- n Ceas



# Tablouri

---

n Obiectul Array este folosit pentru a lucra cu tablouri

n Sintaxa:

n `var ar = new Array() /*sau Array(3)*/`

n `ar[0] = "Saab";`

n `ar[1] = "Fiat";`

n `ar[2] = "Volvo";`

SAU

n `var ar = new Array("Saab", "Fiat", "Volvo")`

# Tablouri accesare-modificare

---

n document.write(ar[0])

n ar[0] = "Opel"

n Exemple

n Crearea unui tablou

n Iterarea peste elementele unui tablou

n concat()

n join()

n sort() literal

n sort() numeric

# Obiectul Boolean

---

- n Folosit pentru a converti valori non-booleene în valori Boolean (true sau false)
- n Este o învelitoare pentru o valoare Boolean
- n Sintaxa (valori false):
  - n `var b=new Boolean()`
  - n `var b=new Boolean()`
  - n `var b=new Boolean(0)`
  - n `var b=new Boolean(null)`
  - n `var b=new Boolean("")`
  - n `var b=new Boolean(false)`
  - n `var b=new Boolean(NaN)`

# Obiectul Boolean

---

n Valori true

n var b=new Boolean(true)

n var b=new Boolean("true")

n var b=new Boolean("false")

n var b=new Boolean("Richard")

# Obiectul Math

---

- n Permite efectuarea operațiilor matematice
- n Valori:
  - n Math.E
  - n Math.PI
  - n Math.SQRT2
  - n Math.SQRT1\_2
  - n Math.LN2
  - n Math.LN10
  - n Math.LOG2E
  - n Math.LOG10E

# Objectul Math

---

## n Metode:

n sin, cos, tan, abs, acos, asin, atan, atan2, ceil, exp, floor, log, min, max, pow, random, sqrt

## n Exemple:

n `document.write(Math.round(4.7))`

n `document.write(Math.random()) /*[0,1)*/`

n `document.write(Math.floor(Math.random()*11))`

# Obiectele HTML DOM

---

- n JavaScript permite accesare și manipularea tuturor obiectelor HTML DOM
- n DOM = Document Object Model
- n HTML DOM = standard W3C
- n Definește o mulțime standard de obiecte pentru HTML, și o modalitate standard de accesare și manipulare a documentelor HTML

# Obiectele HTML DOM

---

- n Toate elementele HTML împreună cu textul și atributele lor pot fi accesate prin DOM
- n Conținutul poate fi modificat sau șters
- n Se pot crea noi elemente
- n HTML DOM este independent de platformă și limbaj; poate fi folosit de orice limbaj: Java, JavaScript, VBScript



# Obiectele HTML DOM

---

## n Window

- n Obiectul de la nivelul cel mai de sus din ierarhie.
- n Reprezintă o fereastră a browserului.
- n Creat automat cu fiecare instanță a elementului `<body>` sau `<frameset>`

## n Proprietăți:

- n closed, defaultStatus, document, history, length, location, name, opener, outerheight, outerwidth, pageXOffset, pageYOffset, parent, personalbar, scrollbars, self, status, statusbar, toolbar, top

# Obiectele HTML DOM

---

## n Metode:

- n alert(), blur(), clearInterval(), clearTimeout(), close(), confirm(), createPopup(), focus(), moveBy(), moveTo(), open(), print(), prompt(), resizeBy(), resizeTo(), scrollBy(), scrollTo(), setInterval(), setTimeout()

# Obiectele HTML DOM

---

- n Navigator = conține informații despre browserul client
- n Screen = conține informații despre displayul clientului
- n History = conține URL-urile vizitate în fereastra browser-ului
- n Location = conține informații despre URL-ul curent

# Detectarea browserului

---

- n Detectarea browserului și a versiunii
- n Mai multe detalii despre browser
- n Toate detaliile despre browser
- n Informație pentru utilizator

# Cookies

---

- n Exemplu
- n Cookie = variabilă stocată pe calculatorul utilizatorului; de fiecare dată când este cerută o pagină va fi trimis și cookie-ul.
- n Cu JS putem crea și obține valoarea unui cookie
- n Exemple: nume, parolă, data

# Cookies

---

## n Creare și stocare

```
function setCookie(c_name,value,expiredays) {
 var exdate=new Date()
 exdate.setDate(exdate.getDate()+expiredays)
 document.cookie=c_name+ "="
 +escape(value)+ ((expiredays==null) ? "" :
 ";expires="+exdate.toGMTString())
}
```

# Cookies

n Obținere cookie

```
function getCookie(c_name) {
 if (document.cookie.length>0) {
 c_start=document.cookie.indexOf(c_name + "=")
 if (c_start!=-1) {
 c_start=c_start + c_name.length+1
 c_end=document.cookie.indexOf(";",c_start)
 if (c_end==-1) c_end=document.cookie.length
 return unescape(document.cookie.substring(c_start,c_end))
 }
 }
 return ""
}
```

# Cookies

n Verificare cookie

```
function checkCookie() {
 username=getCookie('username')
 if (username!=null && username!=""){
 alert('Welcome again '+username+'!')
 } else {
 username=prompt('Please enter your name:',"")
 if (username!=null && username!="") {
 setCookie('username',username,365) }
 }
 }
}
```



# Validarea formularelor

---

- n JS poate fi folosit pentru a valida datele introduse în formulare HTML, înainte de a fi trimise la server
  - n Au fost lăsate câmpuri goale
  - n A fost introdusă o adresă de email validă
  - n A fost introdusă o dată calendaristică corectă
  - n A fost introdus text într-un câmp numeric

# Validarea formularelor

n Câmpuri obligatorii: Exemplu

```
function validate_required(field,alerttxt)
{
 with (field) {
 if (value==null || value=="") {
 alert(alerttxt); return false;
 } else { return true; }
 }
}
```

# Validarea formularelor

n Validare email Exemplu

```
function validate_email(field,alerttxt)
{
 with (field) {
 apos=value.indexOf("@");
 dotpos=value.lastIndexOf(".");
 if (apos<1 || dotpos-apos<2){
 alert(alerttxt);
 return false;
 } else {return true;}
 }
}
```

# Tehnologii web

Curs 8 – JavaScript

# Conținuturi

---

- n Imagini animate
- n Image map JS
- n Evenimente legate de timp
- n Crearea obiectelor
- n Quiz

# Imagini animate

---

- n JS permite crearea de imagini animate
- n Codul JS schimbă imaginile când au loc anumite evenimente
- n Exemplu:

```



```

# Imagini animate

---

n Codul JS

```
<script type="text/javascript">
function mouseOver()
{
 document.b1.src = "b_blue.gif";
}
function mouseOut()
{
 document.b1.src = "b_pink.gif";
}
</script>
```

# Image map JS

---

- n [Exemplu de image map simplu HTML](#)
- n [Exemplu de image map cu JS adăugat](#)



# Evenimente legate de timp

---

- n Putem face ca o funcție să se execute după o anumită perioadă de timp
- n Exemple:
  - n Simplu
  - n La fel de simplu
  - n Evenimente de timp în buclă infinită
  - n Evenimente de timp în buclă infinită cu Stop
  - n Un ceas creat cu evenimente de timp

# Evenimente legate de timp

---

- n setTimeout() – se execută codul ulterior
- n clearTimeout() – anulează setTimeout()
- n Metode ale obiectului HTML DOM window
- n Sintaxa:
  - n var t=setTimeout("instr. JS",milisecunde);
  - n clearTimeout(setTimeout\_variable)

# Evenimente legate de timp - exemplu

---

```
<html> <head>
<script type="text/javascript">
function timedMsg()
{
 var t=setTimeout("alert('5 seconds!')",5000);
}
</script> </head>
<body> <form>
<input type="button" value="Afiseaza alertbox!"
 onClick="timedMsg()">
</form> </body> </html>
```

# Crearea obiectelor

---

- n În JS putem crea propriile noastre obiecte
- n Obiectul = un tip special care conține proprietăți și metode
- n Un obiect de tip Persoana
  - n Proprietăți: vârsta, înălțime etc.
  - n Metode: sleep(), work(), eat() etc.
- n Sintaxa:
  - n objName.propName
  - n objName.methodName()

# Crearea obiectelor

---

- n Sintaxa pentru crearea obiectelor:
  - n `personObj=new Object();`
  - n `personObj.firstname="John";`
  - n `personObj.lastname="Doe";`
  - n `personObj.age=50;`
  - n `personObj.eyecolor="blue";`
  - n `personObj.eat=eat; // crearea unei funcții`

# Crearea obiectelor

---

n Crearea unui șablon (clasă) pentru obiecte:

```
function person(prenume, nume, varsta, culoare)
{
 this.prenume = prenume;
 this.nume = nume;
 this.varsta = varsta;
 this.culoare = culoare;
}
Obj1 = new person("John", "Doe", 50, "blue");
Obj2 = new person("Sally", "Rally", 48, "green");
```

# Crearea obiectelor

## Adăugarea unei funcții

```
function person(prenume, nume, varsta, culoare)
{
 this.prenume = prenume;
 this.nume = nume;
 this.varsta = varsta;
 this.culoare = culoare;
 this.nume_nou = schimba_nume;
}
function schimba_nume(nume_nou)
{
 this.nume=nume_nou;
}
Obj1.nume_nou("Mary");
```

# Crearea obiectelor

---

- n Exemplu 1 (crearea directă a unui obiect)
- n Exemplu 2 (crearea unui șablon)



# Quiz

---

1. În interiorul cărui element HTML se pune codul JS?
  - a. `<js>`
  - b. `<script>`
  - c. `<scripting>`
  - d. `<javascript>`

# Quiz

---

2. Care este sintaxa corectă JS pentru a scrie "Hello world!"
- a. `document.write("Hello World")`
  - b. `response.write("Hello World")`
  - c. `("Hello World")`
  - d. `"Hello World"`

# Quiz

---

3. Care este locul în care trebuie plasate scripturile JS?
  - a. Secțiunea `<body>`
  - b. Secțiunea `<head>` și secțiunea `<body>`
  - c. Secțiunea `<head>`

# Quiz

---

4. Care este sintaxa corectă pentru a ne referi la un script extern numit "xxx.js"?
- a. `<script src="xxx.js">`
  - b. `<script href="xxx.js">`
  - c. `<script name="xxx.js">`

# Quiz

---

5. Un script extern trebuie să conțină elementul `<script>`
- a. Adevărat
  - b. Fals

# Quiz

---

6. Cum se scrie "Hello world!" într-o casetă alert?
- a. `msgBox("Hello World")`
  - b. `alertBox("Hello World")`
  - c. `alertBox="Hello World"`
  - d. `alert("Hello World")`

# Quiz

---

7. Cum se creează o funcție?
- a. `function myFunction()`
  - b. `function:myFunction()`
  - c. `function=myFunction()`

# Quiz

---

8. Cum se apelează o funcție cu numele myFunction?
- a. call myFunction()
  - b. call function myFunction
  - c. myFunction()



# Quiz

---

9. Cum se scrie o expresie condițională pentru executarea unei instrucțiuni doar dacă  $i$  este egal cu 5?
- a. `if i==5 then`
  - b. `if i=5`
  - c. `if i=5 then`
  - d. `if (i==5)`

# Quiz

---

10. Cum se scrie o expresie condițională pentru executarea unor instrucțiuni doar atunci când  $i$  nu este egal cu 5?
- a. `if (i <> 5)`
  - b. `if =! 5 then`
  - c. `if <>5`
  - d. `if (i != 5)`

# Quiz

---

11. Câte tipuri de instrucțiuni repetitive există în JavaScript?
- a. Patru. "for", "while", "do...while", "loop...until" loop
  - b. Una. "for"
  - c. Două. "for", "while"

# Quiz

---

12. Cum începe o buclă for?

- a. `for (i = 0; i <= 5; i++)`
- b. `for i = 1 to 5`
- c. `for (i = 0; i <= 5)`
- d. `for (i <= 5; i++)`

# Quiz

---

13. Cum se adaugă un comentariu în JS?

- a. 'Comentariu
- b. <!-- Comentariu -->
- c. // Comentariu

# Quiz

---

14. Care este sintaxa corectă JS pentru a adăuga un comentariu pe mai multe linii?
- a. `/*Comentariu pe  
mai multe linii*/`
  - b. `// Comentariu pe  
mai multe linii //`
  - c. `<!-- Comentariu pe  
mai multe linii -->`

# Quiz

---

15. Cum se creează tablourile JS?

- a. `var txt = new Array("tim", "kim", "jim")`
- b. `var txt = new Array(1:"tim", 2:"kim", 3:"jim")`
- c. `var txt = new Array="tim", "kim", "jim"`
- d. `var txt = new  
Array:1=("tim")2=("kim")3=("jim")`

# Quiz

---

16. Cum se rotunjește numărul 7.25 la cel mai apropiat întreg?
- a. `Math.round(7.25)`
  - b. `round(7.25)`
  - c. `Math.rnd(7.25)`
  - d. `rnd(7.25)`



# Quiz

---

17. Cum se poate afla cel mai mare număr dintre 2 și 4?

- a. `ceil(2,4)`
- b. `Math.ceil(2,4)`
- c. `Math.max(2,4)`
- d. `top(2,4)`

# Quiz

---

18. Care este sintaxa JS pentru a deschide o fereastră nouă cu numele “window2”?
- a. `new("http://ex.com", "window2")`
  - b. `new.window("http://ex.com", "window2")`
  - c. `window.open("http://ex.com", "window2")`
  - d. `open.new("http://ex.com", "window2")`

# Quiz

---

19. Cum se scrie un mesaj în bara de stare a browser-ului?

- a. `status("mesaj")`
- b. `statusbar = "mesaj"`
- c. `window.status = "mesaj"`
- d. `window.status("mesaj")`

# Quiz

---

20. Cum se poate afla numele browser-ului client?

- a. `browser.name`
- b. `navigator.appName`
- c. `client.navName`

# Tehnologii web

Curs 9 – XML

# Conținuturi

---

- n Introducere
- n Cum se folosește XML
- n Sintaxa
- n Elemente
- n Atribute
- n Validare
- n Browsere
- n Vizualizare
- n CSS
- n XSL
- n Insule de date
- n XML Parser
- n XML Namespace
- n XML CDATA
- n Tehnologii XML
- n Quiz

# Introducere

---

- n XML (EXtensible Markup Language)
  - n A fost proiectat pentru a descrie datele și pentru a se concentra asupra ce reprezintă datele
- n HTML
  - n a fost proiectat pentru a afișa datele și pentru a se concentra asupra modului în care arată datele

# Introducere

---

## n XML

- n Limbaj de marcare, similar cu HTML
- n Proiectat pentru a descrie datele
- n Elementele XML nu sunt predefinite. Oricine poate să-și definească propriile elemente
- n Folosește un DTD (Document Type Definition) sau XML Schema pentru a descrie datele
- n Este o recomandare W3C (feb. 1998)



# Introducere

---

- n Diferențe XML – HTML
  - n XML proiectat pentru a transporta datele
  - n XML nu este un înlocuitor pentru HTML
  - n XML și HTML au fost proiectate cu scopuri diferite
    - n XML – descrie informația
    - n HTML – afișează informația
  - n XML a fost creat pentru structura, stoca și transmite informația

# Introducere

---

n Exemplu:

```
<note>
```

```
 <to>Tove</to>
```

```
 <from>Jani</from>
```

```
 <heading>Reminder</heading>
```

```
 <body>
```

```
 Don't forget me this weekend!
```

```
 </body>
```

```
</note>
```

*Informație învelită în elemente XML*

# Introducere

---

- n Pentru ca “nota” să fie utilă trebuie scrisă o secvență de cod pentru a o afișa, transmite sau recepționa
- n XML este liber și extensibil
- n Elementele XML nu sunt predefinite
- n Elementele folosite în HTML și fixate iar structura este fixată

# Introducere

---

- n În XML elementele pot fi inventate, de asemenea și structura documentului.
- n XML este o unealtă cross-platformă, independentă de software și hardware pentru transmiterea informației

# Cum se folosește?

---

- n XML poate separa datele de prezentare
- n Datele pot fi stocate în afara documentului HTML
- n Modificarea datelor nu necesită modificarea documentului HTML
- n Astfel, ne putem concentra pe partea de prezentare când creăm documentul HTML

# Cum se folosește?

---

- n XML permite transferul de date între sisteme incompatibile prin Internet
- n Convertirea datelor în XML poate reduce semnificativ efortul necesar pentru transmiterea datelor între sisteme diferite.
- n Datele XML sunt stocate în fișiere text simple, astfel datele pot fi partajate cu ușurință
- n Cu XML datele sunt disponibile pentru mai mulți utilizatori

# Cum se folosește?

---

- n XML poate fi folosit pentru a crea noi limbaje
- n XML stă la baza WAP și WML
- n Toate aplicațiile vor salva datele în format XML (vezi docx)

# Sintaxa

---

n Reguli simple și foarte stricte

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<note>
```

```
 <to>Tove</to>
```

```
 <from>Jani</from>
```

```
 <heading>Reminder</heading>
```

```
 <body>Don't forget me this weekend!</body>
```

```
</note>
```



# Sintaxa

---

- n Prima linie definește versiunea XML și encoding-ul folosit pentru caractere.
- n Următoarea linie descrie elementul **root**
- n Următoarele 4 linii descriu elementele copil ale lui root (to, from, heading și body)

# Sintaxa

---

- n Toate elementele XML trebuie să aibă un element de închidere
- n Elementele sunt case-sensitiive
- n Trebuie respectat principiul parantezelor
- n Documentele XML trebuie să aibă un singur element root
- n Orice element poate avea subelemente (elemente copil)

# Sintaxa

---

<root>

  <child>

    <subchild>.....</subchild>

  </child>

</root>

# Sintaxa

- n Valorile atributelor elementelor trebuie să fie delimitate de ghilimele

```
<?xml version="1.0" encoding="ISO-8859-1"?>
 <note date=12/11/2002>
 <to>Tove</to>
 <from>Jani</from>
</note>
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
 <note date="12/11/2002">
 <to>Tove</to>
 <from>Jani</from>
</note>
```

# Sintaxa

---

- n Spațiile albe sunt păstrate
- n Comentarii XML
  - n `<!-- This is a comment -->`

# Elemente

n Elementele XML sunt extensibile

n V1

```
<note>
 <to>Tove</to>
 <from>Jani</from>
 <body>Don't forget me this weekend!</body>
</note>
```

n V2

```
<note>
 <date>2002-08-01</date>
 <to>Tove</to>
 <from>Jani</from> <heading>Reminder</heading>
 <body>Don't forget me this weekend!</body>
</note>
```

# Elemente

---

- n Un program care prelucrează V1 poate prelucra și V2
- n Documentele XML pot fi extinse pentru a purta mai multă informație fără a afecta programele care prelucrează doar varianta inițială. Noua variantă de document XML poate fi prelucrată în continuare de acele programe

# Elemente

---

- n Elementele XML au relații
- n Relațiile sunt de tip părinte-copil
- n **Exemplu:**

## **XML**

### Introducere în XML

- n Ce este HTML
- n Ce este XML

### Sintaxa XML

- n Elementele trebuie să aibă element de închidere
- n Elementele trebuie să respecte principiul parantezelor



# Elemente

---

```
<book>
```

```
<title>XML</title>
```

```
<prod id="33-657" media="paper"></prod>
```

```
<chapter>Introducere în XML
```

```
 <para>Ce este HTML</para>
```

```
 <para>Ce este XML</para>
```

```
</chapter>
```

```
<chapter>Sintaxa XML
```

```
 <para>Elementele trebuie să aibă element de închidere
 </para>
```

```
 <para>Elementele trebuie să respecte principiul
 parantezelor </para>
```

```
</chapter> </book>
```

# Elemente

---

- n Elementele au conținut
  - n Alte elemente (book)
  - n Mixt (chapter)
  - n Simplu (para, title)
  - n Nimic (prod)
- n Elementele pot avea și atribuite = perechi (nume, valoare)

# Atribute

---

- n Sunt folosite pentru a oferi informație suplimentară despre elemente
- n De ex. `<IMG SRC="computer.gif">`
- n Oferă informație care nu face parte din datele efective dar e utilă pentru soft-ul care gestionează elementul:
  - n `<file type="gif">computer.gif</file>`

# Atribute

---

n Atributele trebuie delimitate de

n ‘

n ”

n <gangster name='George "Shotgun" Ziegler'>

n <gangster name="George 'Shotgun' Ziegler">

# Attribute vs. elemente

---

```
<person sex="female">
 <firstname>Anna</firstname>
 <lastname>Smith</lastname>
</person>
```

SAU

```
<person>
 <sex>female</sex>
 <firstname>Anna</firstname>
 <lastname>Smith</lastname>
</person>
```

# Attribute vs. element

---

```
<note date="12/11/2002">
```

```
 <to>Tove</to>
```

```
 <from>Jani</from>
```

```
 <heading>Reminder</heading>
```

```
 <body>Don't forget me this weekend!</body>
```

```
</note>
```

# Attribute vs. element

---

<note>

<date>12/11/2002</date>

<to>Tove</to>

<from>Jani</from>

<heading>Reminder</heading>

<body>Don't forget me this weekend!</body>

</note>

# Attribute vs. element

---

```
<note>
 <date>
 <day>12</day>
 <month>11</month>
 <year>2002</year>
 </date>
 <to>Tove</to>
 <from>Jani</from>
 <heading>Reminder</heading>
 <body>Don't forget me this weekend!</body>
</note>
```



# Attribute vs. elemente

---

Un exemplu negativ:

```
<note day="12" month="11" year="2002"
 to="Tove" from="Jani" heading="Reminder"
 body="Don't forget me this weekend!">
</note>
```

# Atribute

---

- n Dezavantaje
  - n Nu pot conține mai multe valori
  - n Nu pot fi extinse cu ușurință
  - n Nu pot descrie structuri
  - n Sunt mai dificil de gestionat de programe
- n E indicat a se folosi elemente pentru a descrie date
- n Atributele trebuie folosite atunci când trebuie oferită informație care nu e relevantă pentru acele date (pentru metadata)

# Attribute

---

```
<messages>
 <note id="p501">
 <to>Tove</to>
 <from>Jani</from>
 <heading>Reminder</heading>
 <body>Don't forget me this weekend!</body>
 </note>
 <note id="p502">
 <to>Jani</to>
 <from>Tove</from>
 <heading>Re: Reminder</heading>
 <body>I will not!</body> </note>
</messages>
```

# Validare XML

---

- n Document XML **bine format** = respectă regulile de sintaxă
  - n Au un singur element root
  - n Elementele au tag de închidere
  - n Se face distincția între literele mari și mici
  - n Se respectă principiul parantezelor
  - n Atributele sunt delimitate de “ sau ‘

# Validare XML

---

n Document XML **valid** =

- n respectă un **DTD** (Document Type Definition)
- n Respectă o **XML Schema** (alternativă XML la DTD)

# Vizualizarea fișierelor XML

---

- n În browser
- n În editoare XML
  - n Note
  - n Note cu eroare
  - n CD catalog
  - n Catalog de plante
  - n Meniu
- n Dacă vrem ca documentele XML să fie afișate ca pagini web trebuie adăugată informație de afișare

# Vizualizarea fișierelor XML

---

- n Documentele XML nu conțin informație în legătură cu modul în care trebuie afișate datele
- n Elementele sunt inventate așa că browser-ul nu știe cum să le interpreteze
- n Câteva soluții:
  - n CSS, XSL, JavaScript, XML Data Islands

# Afişare XML cu CSS (nu e recomandat)

---

n XML

n CSS

n XML+CSS

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/css"
href="cd_catalog.css"?>
<CATALOG> <CD>
```



# Afișare XML cu XSL (recomandat)

---

- n XSL (eXtensible Stylesheet Language)
- n Mult mai sofisticat decât CSS
- n XSL transformă XML în HTML
  - n XML
  - n XSL
  - n XML + XSL

# XML Data Island (doar IE)

---

n Date XML incluse în fișierul HTML

```
<html>
```

```
 <body>
```

```
 <xml id="note" src="note.xml"></xml>
```

```
 </body>
```

```
</html>
```

# XML Data Island (doar IE)

```
<html> <body>
<xml id="cdcat" src="cd_catalog.xml"></xml>
 <table border="1" datasrc="#cdcat">
 <tr>
 <td></td>
 <td></td>
 </tr>
 </table>
</body> </html>
```

Rezultatul

# Exemplu

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<nitf>
 <head>
 <title>Colombia Earthquake</title>
 </head>
 <body>
 <headline>
 <hl1>143 Dead in Colombia Earthquake</hl1>
 </headline>
 <byline>
 <bytag>By Jared Kotler, Associated Press Writer</bytag>
 </byline>
 <dateline>
 <location>Bogota, Colombia</location>
 <date>Monday January 25 1999 7:28 ET</date>
 </dateline>
 </body>
</nitf>
```

# XML Parser

---

- n Exemplu cross-browser care încarcă un document XML și îl afișează
- n Exemplu cross-browser care încarcă un string XML și îl afișează
- n Pentru gestionarea unui document XML e nevoie de un parser
- n Există diferențe între IE și Mozilla
- n O dată încărcat în memorie documentul XML poate fi gestionat folosind XML DOM

# XML Parser - IE

---

n Instanțierea parser-ului XML Microsoft:

```
var xmlDoc=new ActiveXObject("Microsoft.XMLDOM");
```

n Încărcarea unui document XML

```
xmlDoc.async="false";
```

```
xmlDoc.load("note.xml");
```

# XML Parser - Mozilla

---

```
var xmlDoc =
 document.implementation.createDocument
 ("ns", "root", null);
```

```
var xmlDoc =
 document.implementation.createDocument
 ("", "", null);
xmlDoc.load("note.xml");
```

# XML Namespace

- n Reprezintă o metodă de a evita conflictele între numele elementelor XML:

```
<table>
```

```
 <tr> <td>Apples</td> <td>Bananas</td> </tr>
```

```
</table>
```

Și

```
<table>
```

```
 <name>African Coffee Table</name>
```

```
 <width>80</width>
```

```
 <length>120</length>
```

```
</table>
```



# XML Namespace

---

- n Dacă cele două elemente se pun în același fișier atunci apare un conflict de nume.
- n Două elemente cu același nume cu conținut diferit
- n O soluție:

```
<h:table>
 <h:tr>
 <h:td>Apples</h:td>
 <h:td>Bananas</h:td> </h:tr>
</h:table>
```

# XML Namespace

---

```
<f:table>
```

```
 <f:name>African Coffee Table</f:name>
```

```
 <f:width>80</f:width>
```

```
 <f:length>120</f:length>
```

```
</f:table>
```

Prin utilizarea unui prefix avem două tipuri de table.

# XML Namespace

---

```
<h:table xmlns:h="http://www.w3.org/TR/html4/">
 <h:tr>
 <h:td>Apples</h:td>
 <h:td>Bananas</h:td>
 </h:tr>
</h:table>
```

```
<f:table xmlns:f="http://www.w3schools.com/furniture">
 <f:name>African Coffee Table</f:name>
 <f:width>80</f:width>
 <f:length>120</f:length>
</f:table>
```

# XML Namespace

---

- n Atributul xmlns face ca prefixul să aibă un nume calificat asociat cu un namespace
- n Se plasează în tag-ul de start al unui element
- n Sintaxa:
  - n `xmlns:namespace-prefix="namespaceURI"`
- n Toate elementele copil cu același prefix sunt asociate cu același namespace
- n Rolul adresei este de a face numele namespace-ului unic

# XML Namespace

---

- n Uniform Resource Identifier (URI) = șir de caractere care identifică o resursă Internet
- n Cel mai obișnuit URI este Uniform Resource Locator (URL) = identifică adresa unui domeniu Internet
- n Namespace implicit = nu mai e nevoie de a folosi prefixe
  - n Sintaxa: `xmlns="namespaceURI"`

# XML Namespace

---

```
<table xmlns="http://www.w3.org/TR/html4/">
 <tr>
 <td>Apples</td>
 <td>Bananas</td>
 </tr>
</table>
```

```
<table xmlns="http://www.w3schools.com/furniture">
 <name>African Coffee Table</name>
 <width>80</width>
 <length>120</length>
</table>
```

# XML Namespace

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html> <body> <h2>My CD Collection</h2>
 <table border="1">
 <tr> <th align="left">Title</th> <th align="left">Artist</th> </tr>
 <xsl:for-each select="catalog/cd"> <tr>
 <td><xsl:value-of select="title"/></td>
 <td><xsl:value-of select="artist"/></td>
 </tr> </xsl:for-each>
 </table> </body> </html>
</xsl:template>
</xsl:stylesheet>
```

# XML CDATA

---

- n Textul din secțiuni CDATA va fi ignorat de parser-ul XML
- n Caracterele ilegale trebuie introduse ca entități
- n Nu se poate scrie:
  - n `<message>if salary < 1000 then</message>`
- n Corect este:
  - n `<message>if salary &lt; 1000 then</message>`



# XML CDATA

---

|        |   |           |
|--------|---|-----------|
| &lt;   | < | mai mic   |
| &gt;   | > | mai mare  |
| &amp;  | & | ampersand |
| &apos; | ' | apostrof  |
| &quot; | “ | ghilimele |

# XML CDATA

---

```
<script>
<![CDATA[
function matchwo(a,b)
{
 if (a < b && a < 0) then {
 return 1;
 } else {
 return 0;
 }
}
]]>
</script>
```

# XML Encoding

n Pentru a introduce diacritice (ă, î, â, ș, ț) într-un document XML acesta trebuie salvat ca fișier Unicode și precizat encoding-ul folosit la salvare

```
<?xml version="1.0"
 encoding="windows-1252"?>
```

```
<?xml version="1.0"
 encoding="ISO-8859-1"?>
```

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<?xml version="1.0" encoding="UTF-16"?>
```

# XML Aplicație

---

- n Exemplu
- n Datele dintr-un document XML pot fi aduse într-o pagină web printr-un data island (doar IE)
- n Câmpurile pot fi legate de elemente HTML care au atributul **datasrc** și **datafld** (table, span, div)

# Obiectul XMLHttpRequest

---

- n Cu un request HTTP o pagină poate cere și primi un răspuns de la server fără a reîncărca pagina
- n Se rămâne pe aceeași pagină în timp ce script-urile transferă date în fundal
- n Se poate schimba pagina după ce a fost încărcată
- n Exemplu: [Google Suggest](#)

# Obiectul XMLHttpRequest

---

n Este un obiect JavaScript

n Creare:

n Mozilla, Firefox, Safari, Opera, and Netscape

```
var xmlhttp=new XMLHttpRequest()
```

n IE

```
var xmlhttp =
```

```
new ActiveXObject("Microsoft.XMLHTTP")
```

n [Exemplu](#), [Exemplu2](#),

# Tehnologii XML

---

- n **XHTML (Extensible HTML)** versiune de HTML mai strictă.
- n **XML DOM (XML Document Object Model)** definește un mod standard pentru accesarea și manipularea documentelor XML.
- n **XSL (Extensible Style Sheet Language)** - XSL e format din 3 părți:
  - n **XSLT (XSL Transformation)** – limbaj pentru transformarea documentelor XML în alte formate XML de exemplu XHTML
  - n XPath – limbaj pentru navigarea în documentele XML
  - n **XSL-FO (Extensible Style Sheet Language Formatting Objects)** – limbaj pentru formatarea documentelor XML pentru a fi afișate pe ecran, hârtie etc.

# Tehnologii XML

---

- n **XLink (XML Linking Language)** limbaj pentru crearea de hyperlink-uri în documente XML
- n **XPointer (XML Pointer Language)** permite hyperlink-urilor XLink să indice la anumite părți din documentul XML
- n **DTD (Document Type Definition)** folosit pentru a defini elementele permise într-un document XML



# Tehnologii XML

---

- n **XSD (XML Schema)** alternativă bazată pe XML pentru DTD
- n **XForms (XML Forms)** folosește XML pentru a defini date formular
- n **XQuery (XML Query Language)** proiectat pentru a interoga date XML
- n **SOAP (Simple Object Access Protocol)** este un protocol bazat pe XML care permite aplicațiilor să transfere date prin HTTP

# Tehnologii XML

---

- n **WSDL (Web Services Description Language)** limbaj bazat pe XML pentru a descrie serviciile web
- n **RDF (Resource Description Framework)** limbaj bazat pe XML pentru descrierea resurselor web
- n **RSS (Really Simple Syndication)** este un format pentru sindicalizarea știrilor și a conținutului website-urilor ce conțin știri

# Tehnologii XML

---

- n **WAP (Wireless Application Protocol)** proiectat pentru a prezenta conținutul web unor clienți mobili, de ex. telefoane mobile
- n **SMIL (Synchronized Multimedia Integration Language)** limbaj pentru descrierea prezentărilor audiovizuale
- n **SVG (Scalable Vector Graphics)** permite definirea graficelor în format XML

# Quiz

---

1. Ce reprezintă XML?
  - a. Example Markup Language
  - b. eXtensible Markup Language
  - c. X-Markup Language
  - d. eXtra Modern Link

# Quiz

---

2. Cum se pot descrie datele XML?
  - a. XML folosește DTD pentru a descrie datele
  - b. XML folosește XSL pentru a descrie datele
  - c. XML folosește un nod de descriere pentru a descrie datele

# Quiz

---

3. XML a fost creat pentru a înlocui HTML?
  - a. Adevarat
  - b. Fals

# Quiz

---

4. Care e sintaxa corectă pentru definirea versiunii XML?
- a. `<xml version="1.0" />`
  - b. `<?xml version="1.0"?>`
  - c. `<?xml version="1.0" />`

# Quiz

---

5. Ce înseamnă DTD?
- a. Document Type Definition
  - b. Dynamic Type Definition
  - c. Direct Type Definition
  - d. Do The Dance



# Quiz

---

6. Secvența de mai jos este “bine formată”?

```
<?xml version="1.0"?>
```

```
<note>
```

```
 <to>Tove</to>
```

```
 <from>Jani</from>
```

```
 <heading>Reminder</heading>
```

```
 <body>Don't forget me this weekend!</body>
```

```
</note>
```

a. Da

b. Nu

# Quiz

---

7. Secvența de mai jos este “bine formată”?

```
<?xml version="1.0"?>
```

```
<to>Tove</to>
```

```
<from>Jani</from>
```

```
<heading>Reminder</heading>
```

```
<body>Don't forget me this weekend!</body>
```

- a. Da
- b. Nu

# Quiz

---

8. Care afirmație este adevărată?
- a. Toate numele elementelor XML trebuie scrise cu litere mici
  - b. Toate afirmațiile sunt adevărate
  - c. Toate documentele XML trebuie să aibă DTD
  - d. Toate elementele XML trebuie să fie închise corespunzător

# Quiz

---

9. Care afirmație este adevărată?
- a. Elementele XML trebuie să fie imbricate (încuibate) corespunzător
  - b. Tag-urile XML sunt case-sensitive
  - c. Documentele XML trebuie să aibă un element rădăcină
  - d. Toate afirmațiile sunt adevărate

# Quiz

---

10. XML păstrează spațiile albe?

- a. Da
- b. Nu

# Quiz

---

11. Este acesta un document “bine format”?

```
<?xml version="1.0"?>
```

```
<note>
```

```
 <to age="29">Tove</to>
```

```
 <from>Jani</from>
```

```
</note>
```

- a. Da
- b. Nu

# Quiz

---

12. Este acesta un document “bine format”?

```
<?xml version="1.0"?>
```

```
<note>
```

```
 <to age=29>Tove</to>
```

```
 <from>Jani</from>
```

```
</note>
```

- a. Da
- b. Nu

# Quiz

---

13. Elementele XML nu pot fi goale?

- a. Adevărat
- b. Fals



# Quiz

---

14. Care nu e nume corect pentru un element XML?

- a. Toate trei
- b. <h1>
- c. <1dollar>
- d. <Note>

# Quiz

---

15. Care nu e nume corect pentru un element XML?

- a. <age>
- b. <NAME>
- c. <first name>
- d. Toate trei

# Quiz

---

15. Care nu e nume corect pentru un element XML?

- a. <age>
- b. <NAME>
- c. <first name>
- d. Toate trei

# Quiz

---

16. Care nu e nume corect pentru un element XML?

- a. <7eleven>
- b. Toate trei
- c. <phone number>
- d. <xmldocument>

# Quiz

---

17. Atributele XML trebuie întotdeauna delimitate de ghilimele

- a. Da
- b. Nu

# Quiz

---

18. Ce înseamnă XSL?

- a. eXpandable Style Language
- b. eXtensible Style Listing
- c. eXtra Style Language
- d. eXtensible Stylesheet Language

# Quiz

---

19. Care e varianta corectă de a te referi la un stylesheet cu numele “mystyle.xsl”?

- a. `<?xml-stylesheet type="text/xsl" href="mystyle.xsl" ?>`
- b. `<stylesheet type="text/xsl" href="mystyle.xsl" />`
- c. `<link type="text/xsl" href="mystyle.xsl" />`

# Quiz

---

20. Pentru ca parser-ul XML să ignore procesarea unei părți dintr-un document XML, trebuie folosită sintaxa:

- a. `<CDATA> Text ignorat </CDATA>`
- b. `<![CDATA[ Text ignorat ]]>`
- c. `<PCDATA> Text ignorat </PCDATA>`
- d. `<xml:CDATA[ Text ignorat ]>`