


A decorative graphic consisting of a thin yellow circle on the left side. A thick horizontal bar with a green-to-white gradient spans across the middle. A large black left square bracket is on the left side of the bar, and a large yellow right square bracket is on the right side of the bar.

# XHTML 2.0

*Good-bye backward compatibility,  
hello structure!*

**Marius-Gabriel Butuc**  
*Facultatea de Informatică, Iași*

A large black left bracket and a large yellow right bracket are positioned at the top of the slide. A horizontal line with a light green-to-white gradient runs across the slide, passing through the middle of the brackets.

*“If I have seen further, it is by standing on the shoulders of giants.”*

**Sir Isaac Newton**

# Cuprins

- Îmbunătățiri după XHTML 1.1
  - Secțiuni, titluri, legături, liste, imagini
- XForms
- XML Events
- XFrames

# XHTML 2.0

- *Controversat*: rupe compatibilitatea cu versiunile anterioare;
- *Un nou limbaj de marcare*, nu doar o nouă versiune;
- *Compatibil*: XHTML 2.0 poate fi parsat la fel cum un agent parsează XHTML 1.1: via parser XML și CSS compatibil XHTML 2.0 Working Draft-ului actual.

# Prezentare vs. Semantică

- Dispar elementele pur prezentaționale  
`<i>`, `<b>`, `<tt>`, `<br />` etc.
- Se păstrează doar elementele ce au semantică  
`<em>`, `<strong>`, `<line>` etc.

# Prezentare (*exemplu*)

<p>

```
int main() { <br />
```

```
write("Hello world!\n"); <br />
```

```
return 0; <br />
```

```
}
```

</p>

# Semantică (*exemplu*)

<p>

<line>int main() { </line>

<line>write("Hello  
world! \n"); </line>

<line>return 0; </line>

<line>} </line>

</p>

# Secțiuni

- Până la XHTML 2.0: `<h1>` ... `<h6>`
- XHTML 2.0: titluri generice și secțiuni;
- Avantaje:
  - Motoarele de căutare pot înțelege mai ușor importanța relativă la conținut;
  - Secțiunile se auto-conțin, putem astfel să avem chiar material introductiv deasupra titlului.

# Titluri uzuale (exemplu)

```
<html>
<head><title>Intro</title></head>
<body>
  <h1>Introduction</h1>
  <h2>Quo Vadis Web 2.0?</h2>
    <h3>The XML Web</h3>
    <p>XML Web: behind the scenes</p>
    <h3>The Semantic Web</h3>
  <h2>What is AJAX?</h2>
  <h2>What are microformats?</h2>
  <h2>What is FOAF?</h2>
</body>
</html>
```

# Secțiuni & titluri (*exemplu*)

```
<html>
<head><title>Intro</title></head>
<body>
<section>
  <h>Introduction</h>
  <section>
    <h>Quo Vadis Web 2.0?</h>
    <section>
      <h>The XML Web</h>
      <p>XML Web: behind the scenes</p>
    </section>
  </section>
</section>
...
</section>
</body>
</html>
```

# Hyperlink-uri

- Orice element poate fi transformat în hyperlink adăugându-i atributul **href**;  
`<li href="art.xml">Articole</li>`
- Similar XLink, dar incompatibile datorită diferențelor de design.
- Funcționalități XLink pot fi simulate prin această funcționalitate, liste navigaționale, elementul **link** și RDF.

# [ Liste navigaționale & meniuri ]

- Similare cu listele ordonate și neordonate;
- *Avantaj:* elementele apar doar când lista este activă (ca la meniurile ierarhice de tip *pop-up*);
- Oferă multe informații de navigare deși folosesc puțin spațiu efectiv.

# Listă navigațională (*exemplu*)

```
<nl>
  <name>Geometrie</name>
  <li href="acasa.html">Acasă</li>
  <nl>
    <name>Plan</name>
    <li href="prel.html">Preliminarii</li>
    <li href="triu.html">Triunghi</li>
    <li href="4lat.html">Patrulater</li>
    <li href="cerc.html">Cerc</li>
  </nl>
  <li href="unelte.html">Unelte</li>
</nl>
```

# Imagini: ca obiecte

- Înlocuirea elementului `img` cu elementul `object`;
- `object` permite construcții în cascadă: dacă browser-ul nu poate afișa un anumit obiect, va afișa conținutul său;
- Astfel dispăre și necesitatea atributului `alt` al elementului `img`.

# Obiecte (*exemplu*)

```
<object data="free.mpg"  
        type="application/mpeg">  
  <object data="Ana.jpg"  
        type="image/jpeg">  
    Free...<em>like a bird</em>  
  </object>  
</object>
```

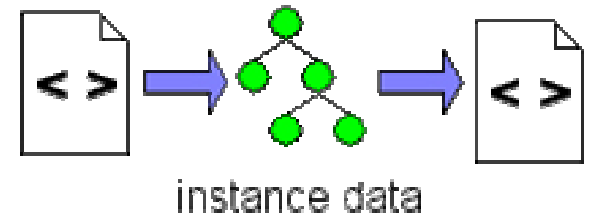
# XForms

- *Scop*: colectarea datelor;
- Independente de platformă / dispozitiv;
- Separă datele și logica de prezentare;
- Folosesc XML pentru definirea datelor;
- Au facilități precum calcule și validări;
- Reduc sau chiar elimină necesitatea de *scripting*.

# Procesarea XForms

- Se parsează în memorie date de la intrare;

- La modificarea datelor unui element se modifica *instance data*;



- La submit, *instance data* se serializează (XML) și se trimite serverului.


# XForms: Controale Text

- `<input>`, `<textarea>` și `<secret>`  
(echivalent `<input type="password">`)
- Comportament similar HTML;
- Permit tipuri de date mai bogate;
- Suporta *input modes*: text  
internațional, *case conversion*,  
*predictive input*;

# XForms: Controale de tip listă

- `<select>` - alegerea a zero, una sau mai multe opțiuni;
- `<select1>` - alege o singură opțiune;
- Nivel de abstractizare ridicat:
  - Fiecare dispozitiv poate afișa după cerințele sale;
  - Control sporit asupra aspectului prin CSS;
  - Atributul `appearance`.

# XForms: Controale noi

- **<output>** - afișează inline *instance data*.
  - “Comanda celor **4** produse costa **17** RON.”
- **<range>** - permite selectarea continuă a unei valori între 2 extreme. 
- **<upload>** - pe lângă fișiere sistem obișnuite, putem conecta acest control și la scannere, camere foto digitale, microfoane, video etc.

# XForms în XHTML

- Singura tehnologie pentru formulare din XHTML 2.0;
- +20% funcționalitate = -80% *scripting*;
- Datorită XPath suportă calcule, validări specifice, vizibilitatea controalelor etc.;
- Datorită XML Schema înglobează *pattern*-uri, restricții de lungime etc.
- ...*Exemple practice* 😊

# XML Events

- Specifică o acțiune ce trebuie să aibă loc la declanșarea unui eveniment;
- Înlocuiește DOM Event și folosește XML Document Object Model;
- Utilizatorul poate declara evenimente proprii și cum trebuie reacționat la declanșarea lor.

# XML Events: componente

- Un eveniment este asociat unei ținte:

```
<html>
  <head><title>Ronin</title></head>
  <body>
    <ul id="blog_menu">
      <li href="about.html">About me</li>
      <li href="archi.html">Archive</li>
      <li href="inspi.html">Inspiration</li>
    </ul>
  </body>
</html>
```

# XML Events: componente

- Evenimentul parcurge calea de la rădăcină până la țintă și înapoi:

(root) → html → body → ul → li → ul → body → html → (root)

- Drumul până la țintă: *capture phase*;
- Drumul înapoi: *bubbling phase*.

# XML Events: componente

- În orice etapă din drum, evenimentul poate întâlni un obiect înregistrat ca *observer*.
- *Observer*-ul așteaptă un eveniment, iar la întâlnirea sa, execută o acțiune.
- *Observer*-ul este creat de către un *listener*.

# XML Events: componente

```
<ev:listener observer="blog_menu"  
    event="mousedown"  
    handler="#scriptul_meu" />
```

- *Listener*-ul face întreaga lista *observer*;
- La click pe orice element al listei, *observer*-ul (**blog\_menu**) execută **scriptul\_meu**.

# XFrames

- Înlocuiește elementul `<frameset>`
- Un nou tip de document ce poate fi folosit cu XHTML, SVG etc.
- Documentul XFrames definește doar cum se compun mai multe documente într-unul singur.

# XFrames (*exemplu*)

- Un document: un rând, două coloane

```
<frames
  xmlns="http://www.w3.org/2002/06/xframes">
  <style> ... </style>
  <row>
    <frame id="navigation"/>
    <frame id="main"/>
  </row>
</frames>
```

# XFrames

- Fiecare frame poate avea un URI implicit;
- Putem decide ce afișăm într-un frame folosind un URI cu sintaxă specială:  
[http://exemplu.ro/#frames\(navigation=nav.svg,main=m.html\)](http://exemplu.ro/#frames(navigation=nav.svg,main=m.html))
- Avantaje:
  - Bookmarking & butonul “Back”;
  - Framejack;

# Alte noutăți

- Orice element poate face referință la medii alternative prin atributul **src**:  

```
<p src="palat.jpg"  
  type="image/jpeg">  
Palatul Culturii</p>
```
- Adăugarea atributelor **property** și **about**, facilitând conversia din XHTML în RDF / XML.

# Rezumat

- Îmbunătățiri după XHTML 1.1
  - Secțiuni, titluri, legături, liste, imagini
- XForms
- XML Events
- XFrames