

TECNOLOGIES DE DESENVOLUPAMENT PER A INTERNET I WEB

GRAU D'ENGINYERIA INFORMÀTICA - CURS 2013/2014

TEMA 1 – *LLENGUATGES DE PROGRAMACIÓ WEB*

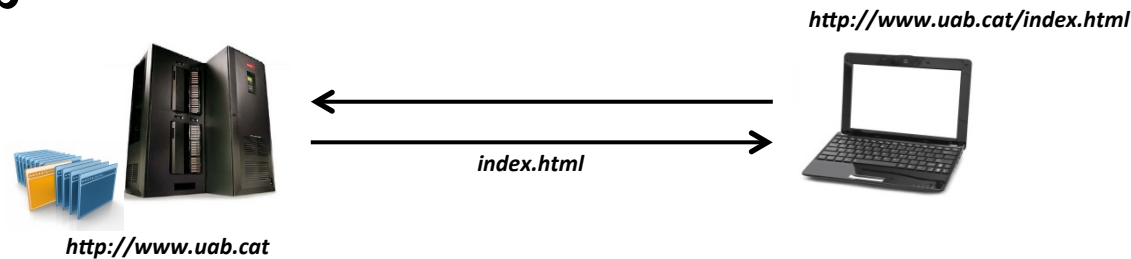


- 1. LLENGUATGES PROGRAMACIÓ WEB**
2. ARQUITECTURA DE LA WEB
3. SERVIDORS D'APLICACIONS
4. DISSENY D'APLICACIONS WEB





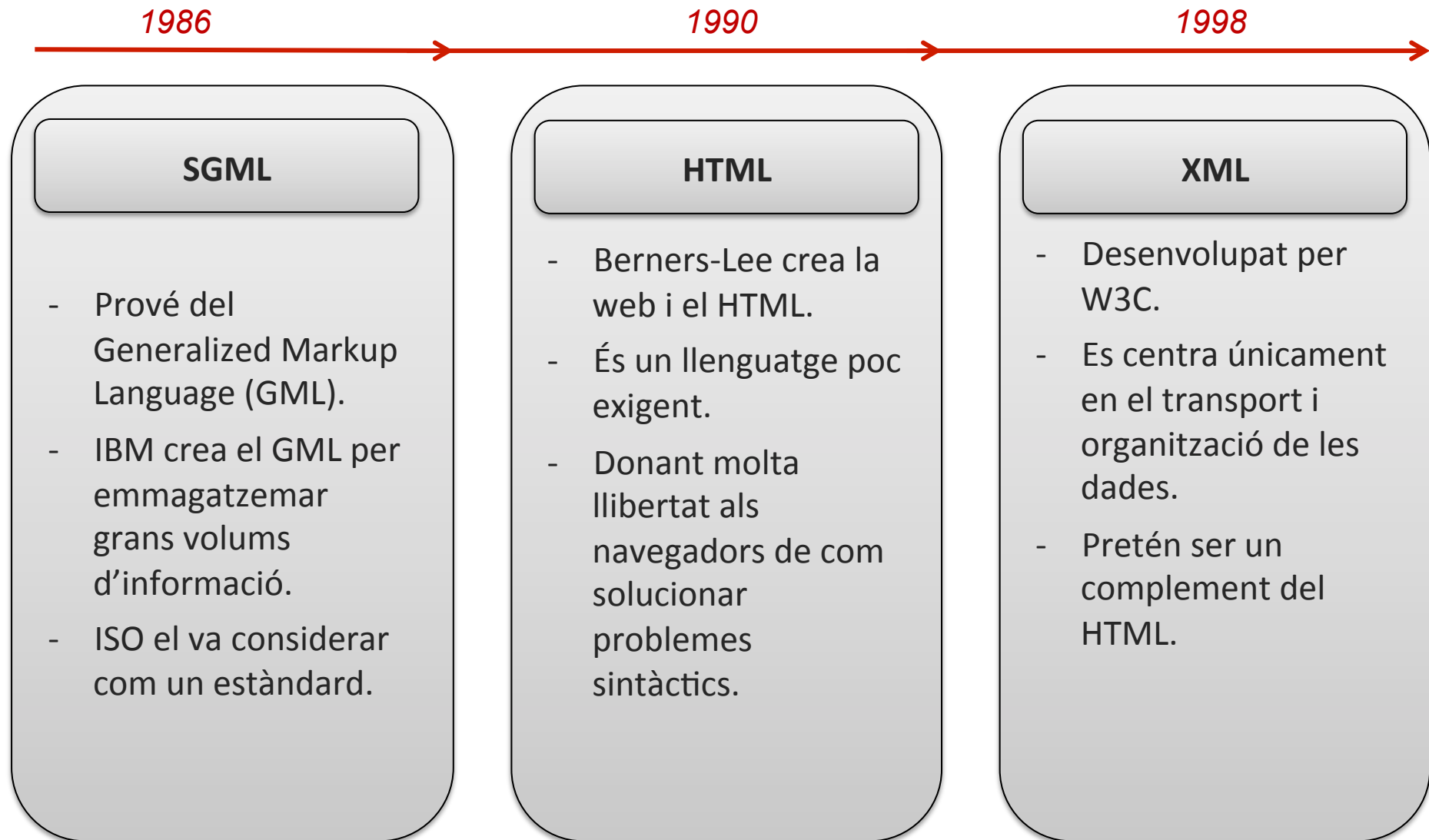
Documents Web

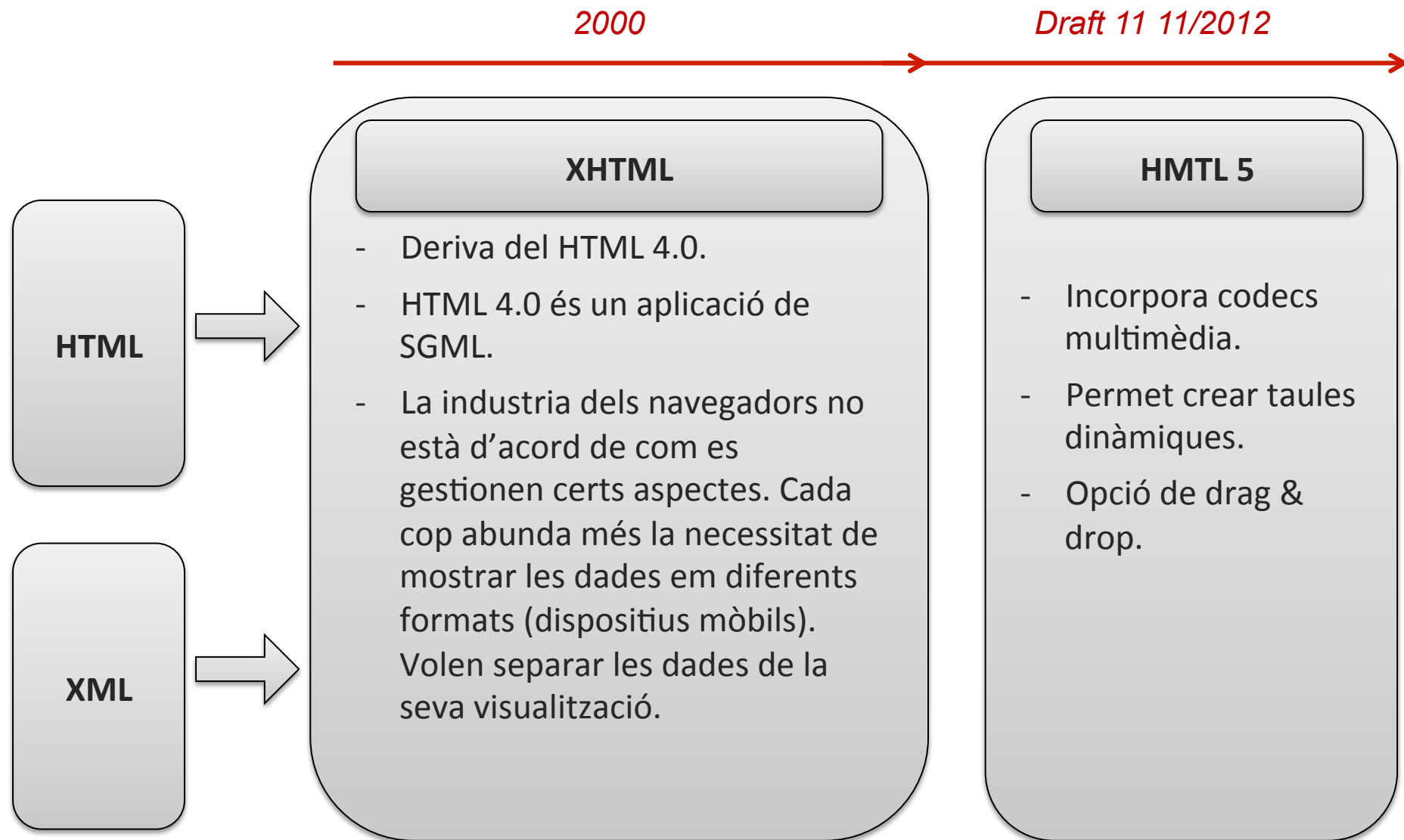


Funcionament (simplificat):

- 1) L'usuari especifica una URL en el navegador.
- 2) El navegador esdevé un client HTTP obrint una connexió utilitzant el protocol HTTP cap al servidor especificat en l'URL.
- 3) El servidor rep la petició del client i comprova que el document HTML existeixi en la seva biblioteca.
- 4) Si el document existeix, el retorna al client i s'acaba la connexió.
- 5) Si no existeix, retorna un codi d'error i s'acaba la connexió .

**Què són capaçs de visualitzar
els navegadors web?**







¡PREGUNTA D'EXAMEN!

Quin impacte té sobre els dispositius l'ús de XHTML?



1. LLENGUATGES PROGRAMACIÓ WEB

1.1 – DOCUMENTS WEB

1.2 – LLENGUATGES DE PROGRAMACIÓ

1.3 – ASPECTES BÀSICS DE SEGURETAT

2. ARQUITECTURA DE LA WEB

3. SERVIDORS D'APLICACIONS

4. DISSENY D'APLICACIONS WEB





1. PROGRAMACIÓ WEB

1.1 – DOCUMENTS WEB

- HTML5
- Models de presentació CSS
- Llenguatges especialitzats SVG, SMIL, i RSS

1.2 – LLENGUATGES DE PROGRAMACIÓ

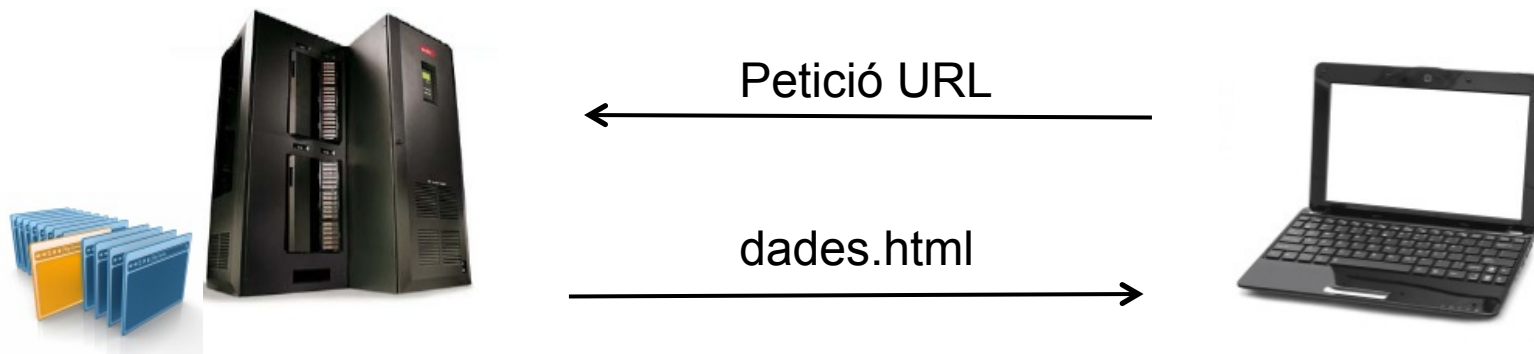
1.3 – ASPECTES BÀSICS DE SEGURETAT





HTML5.

HTML 5 és la última versió del Hypertext Markup Language (HTML). És el canvi més important fet fins el moment ja que introdueix funcionalitats específiques que permeten definir el concepte d'aplicacions web.





HTML5.

Les principals característiques diferenciadores del HTML5 són:

1. Incorporació d'un canvas que permet drag & drop.
2. Totes les noves característiques estan basades en HTML, CSS, DOM, i Javascript.
3. Utilització de tags multimèdia, vídeo i àudio. Pretén reduir el número de plugins externs com Flash.
4. Millor gestió dels errors.
5. Més marcadors ("tags") per tal de substituir scripts.
6. HTML5 vol ser independent de dispositiu.

Mentre que amb XHTML existeixen un conjunt de DTD: XHTML 1.0 Strict, XHTML 1.0 Transitional, i XHTML 1.0 Frameset. En HTML5, és més simple, existint només un sol tipus de DTD, el HTML.



HTML5.

- HTML5, al igual que tots els seus antecessors, és un llenguatge basat en marcadors, etiquetes, o més coneguts com tags.
- HTML5 esta basat en elements, o cada element té dos propietats bàsiques: atributs i contingut.
- En general un element té un tag d'inici <nom-element> i un tag final </nom-element>.
- Els atributs del element es troben en la etiqueta d'inici, i el contingut es troba dins del tag inici i final.

<nom-element atribut="valor">Contingut</nom-element>



Document Type Definition (DTD)

Degut a les prestacions del navegadors web, per interpretar les dades rebudes pel servidor, cal indicar al navegador (que es troba en el client) com volem que les dades rebudes per part del servidor es renderitzin. Per aquest motiu s'utilitza el doctype declaration.

El doctype declaration li diu al navegador amb quina versió del HTML està escrita la web que ha de renderitzar. Aquest procés es fa referint el Document Type Definition (DTD). De manera que, el DTD especifica les regles utilitzades pel llenguatge, així els navegadors poder renderitzar de forma correcta el contingut.

En la actual especificació del HTML hi ha forces doctypes, i les seves diferències no són gaire clares.



Document Type Definition (DTD)

Doctype	Prestacions	Exemple
XHTML 1.0 Strict	només inclou els elements i atributs no marcats com a “deprecated” en HTML 4.0.1	<code><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"></code>
XHTML 1.0 Transitional	equivalent al HTML 4.0.1, inclou també elements i atributs “deprecated”.	<code><DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd"></code>
XHTML 1.0 Frameset	equivalent al HTML 4.0.1, però també inclou la possibilitat de definir frames.	<code><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd"></code>
HTML5		<code><!DOCTYPE html></code>



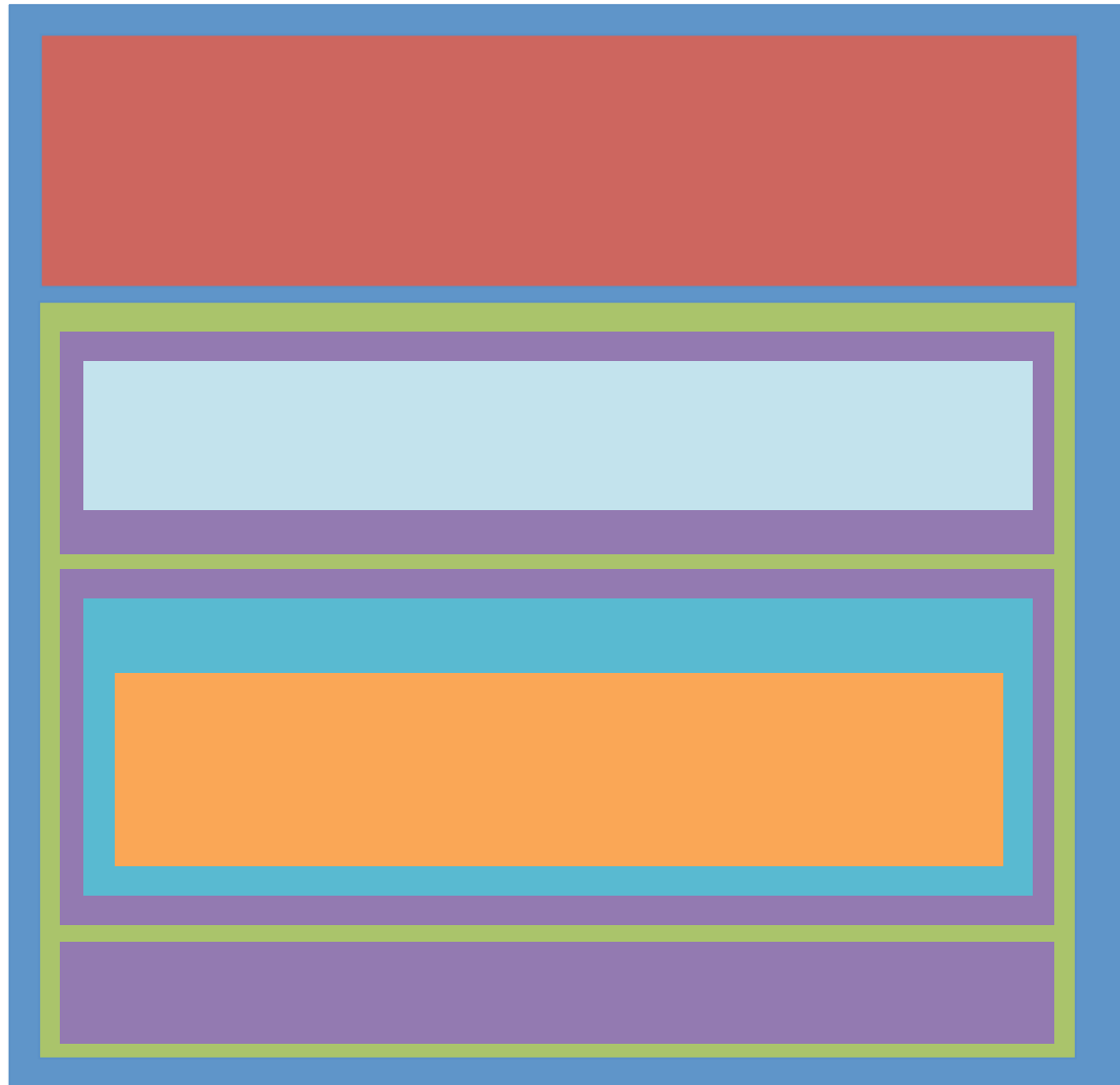
Elements estructurals

Els elements estructurals del contingut de la web són els que permeten organitzar el contingut de la web en capçalera, secció, article, informació apart, peu de pàgina, links a altres webs.



Elements estructurals

```
<!DOCTYPE html>
<html>
  <head>
    <title>
      títol
    </title>
  </head>
  <body>
    <header> capçalera
    <nav>
      links
    </nav>
  </header>
  <section>
    <article>
      contingut
    <aside>
      extra
    </aside>
  </article>
</section>
  <footer>
    Footer
  </footer>
</body>
</html>
```





Elements estructurals

Bàsicament existeixen 2 seccions: el **head** i el **body**.

- El **head** conté informació què descriu la web. Aquest “tag” pot contenir dos “tags” més, el **title** i el **meta**. El **title** indica el títol de la web, mentre que el **meta** conté informació de la codificació de caràcters, informació útil per motors de cerca.

`<meta http-equiv="content-type|default-style|refresh" content=" ">`

Valor	Descripció
content-type	Especifica la codificació dels caràcters del document. <meta http-equiv="content-type" content="text/html; charset=UTF-8">
default-style	Especifica l'estil predefinit. <meta http-equiv="default-style" content="XXX">
refresh	Defineix l'interval de temps en que es refresca la web. <meta http-equiv="refresh" content="300">



Elements estructurals

```
<!DOCTYPE html >
<head>
<title>web amb exemples</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
</head>
<body> Importància</body>
</html>
```

[UTF.html](#)

[ISO.html](#)

- El **body** conté els elements i text que es mostren en la pàgina web: text, taules, botons, links, imatges, etc



HTML5 - Tags bàsics

L'objectiu de l'element **body** és contenir el text i els elements HTML en el navegador. Normalment el text s'organitza amb els elements estructurals.

- *Headings*: Permet posar titulars de diferents mides. Els tags `<h1>...</h1>` pel més gran i `<h6>...</h6>` pel més petit. Utilitzat per destacar seccions.
- *Paràgraf*: s'identifiquen amb el tag `<p>...</p>`. Utilitzat per agrupar text.
- *Links*: s'identifiquen amb el tag `<a>...</a>`. Utilitzat anar a una altre web.
- *Imatges*: s'identifiquen amb el tag `<img>...</img>`. Utilitzat per inserir imatges.
- *Línies*: el tag `<hr>` crea una línia horitzontal. Utilitzat com a separador.
- *Comentaris*: En HTML també és possible afegir comentaris amb el tag `<!-- ... -->`, de manera que serà informació no renderitzada pel navegador.



HTML5 - Tags bàsics

- *Donar format al text*: els tags de per donar format al text en HTML poden ser de tipus lògics o físics:
 - Lògics: depèn de cada navegador interpretar cada estil. ****, ****, **<cite>**, **<code>**, **<dfn>**, **<kbd>**, **<samp>**, **<var>**.
 - Físics: aporten instruccions de format específiques pel navegador. ****, **<i>**, **<big>**, **<small>**, **<sub>**, **<sup>**, **<strike>**, **<u>**.
 - Caràcters especials: utilitzats per mostrar caràcters que s'utilitzen per identificar tags, o el de copyright. **"**, **©**, **&**, ** **, **—**, **|**

[html5_tags_basics.html](#)



HTML5 - Llistes

- *Llistes*: permeten organitzar termes i les seves definicions. Existeixen tres tipus de llistes:
 1. *Llistes de definició*: útil per a informació que consisteix en un nombre de termes i les seves descripcions. Els “tags” `<dl>...</dl>` defineixen la llista. Cada terme de la llista és taguejat amb `<dt>...</dt>`, i la seva definició amb `<dd>...</dd>`.
 2. *Llistes ordenades*: s'identifiquen amb el “tag” `<ol type=“X”>...</ol>`. **X** permet identificar el tipus de bullet que utilitzarem: **1** per números, **A** per lletres majúscules, **a** per lletres minúscules, **I** per números romans en majúscula i, **i** per números romans en minúscula.
 3. *Llistes desordenades*: s'identifiquen amb el “tag” `<ul type=“X”>...</ul>`. L'atribut **type** permet identificar el tipus de bullet que utilitzarem, tenim tres opcions: **disc**, **square**, i **circle**.



[html5_llistes.html](#)



HTML5 – Links

- *Links*: són elements que ens permeten anar d'una pàgina a una altre. S'identifiquen amb el tag `<a href="">...</a>`. Hi han dos tipus de links depenen del valor del atribut href. Links absoluts i links relatius.
 1. Links absoluts: utilitzat per lincar a una web externa.
 2. Links relatius: utilitzat per lincar a una pàgina web interna.
 3. Links a un id definit: utilitzat, per exemple, per anar una secció concreta d'una pàgina.



[html5_links.html](#)

W3C ofereix un servei gratuït per a la validació de llenguatge de marques a <http://validator.w3.org/> que validarà el nostre document HTML i comprovarà la seva sintaxis.



HTML5 – Imatges

El tag **** és un tag buit que permet inserir imatges, pot contenir una sèrie d'atributs.

- *Font de la imatge*: ruta de la imatge a carregar, pot ser local o externa.

```

```

```

```

- *Mida de la imatge*: permet definir la mida de la imatge, amplada i alçada.

```
width="304" height="228"
```

- *Crea una imatge com a link*:

```
<a><img></a>
```

- *Crear una imatge amb zones d'interacció*:

Les coordenades del mapa d'interacció són relatives a la mida de la imatge.

[html5_images.html](#)



HTML5 – Taules

Les taules es defineixen amb el tag **<table>**, les taules poden contenir una sèrie de tags que ens permeten definir:

- *Files i columnes:* **<tr>** i **<td>**, permet definir les files i les columnes en cada fila.
- *La capçalera d'una columna:* **<th>**, incorpora una columna de tipus capçalera.
- *Descripció de la taula:* **<caption>**, s'incorpora just després del tag table.
- *Format d'estil per columnes:* **<colgroup>**, permet definir l'estil per un conjunt de columnes.
- *Format d'estil per files:* **<thead>**, **<tbody>**, i **<tfoot>**, permeten definir l'estil de la font per files.

[html5_taulas.html](#)



HTML5 - Tags multimèdia

Avui en dia les eines multimèdia són dels recursos més consumits en la web. Per aquest motiu HTML5 incorpora un conjunt de “tags” que permeten la interacció per part del usuari amb elements multimèdia.

Cal tenir present que algunes de les prestacions no estan incorporades en la majoria de navegadors.

- **<audio>....</audio>** permet definir elements de so i interactuar amb ells utilitzant diferents atributs com: *autoplay, controls, loop, preload* o *src*.

[html5_audio.html](#)

- **<video>....</video>** permet definir elements de video i interactuar amb ells utilitzant diferents atributs com: *autoplay, controls, height, loop, muted, poster, preload, src* o *width*.

[html5_video.html](#)



HTML5 - Tags multimèdia

- **<source>....</source>** permet definir elements d'àudio o vídeo, permeten al navegador seleccionar el arxiu a reproduir depenent del format suportat pel navegador.

```
<!DOCTYPE html>
<html>
<body>
<audio controls="controls">
<source src="audio.ogg" type="audio/ogg">
<source src="audio.mp3" type="audio/mpeg">
El navegador no suporta el form d'audio ogg ni mp3!!
</audio>
</body>
</html>
```



HTML5 - Tags multimèdia

- **<embed>....</embed>** defineix un container per a una aplicació externa o contingut interactiu, sol requerir un pluguin. El tipus d'arxius externes suportats es poden consultar a <http://www.iana.org/assignments/media-types/index.html>

```
<!DOCTYPE html>
<html>
<body>
<embed src="helloworld.swf">
</body>
</html>
```

- **<track>....</track>** permet especificar els subtítols, i altres arxius de text mentre s'està reproduint element d'àudio o vídeo.

A data d'avui la majoria de navegadors suporten els “tags” multimèdia que hem vist, a excepció del “tag” <track>...</track>, què no és suportat per cap dels navegadors més rellevants.



HTML5 – Formularis

Els formularis s'utilitzen per enviar dades del client al servidor.

Un formulari HTML pot contenir diferents elements per inserir dades: camps de text, checkboxes, radio-buttons, submit buttons, etc. A més un formulari pot contenir llistes de selecció, inserció de text, etiquetes, etc.

Els formularis estan definits amb el tag **<form>...</form>** què pot tenir els següents atributs:

- *name*: indica el nom del formulari. Útil per referenciar en javascript o les dades del formulari un cop han estat enviades.
- *accept-charset*: llista separada per espais que conté la codificació de caràcters acceptada pel servidor. (UTF-8, ISO-8859-1)
- *action*: URL on s'enviaran les dades un cop s'hagi sotmès el formulari. Pot contenir una URL absoluta o relativa.
- *autocomplete*: Quan aquest atribut es troba a on el navegador dona l'opció al usuari d'emplenar el formulari amb dades que han estat introduïdes anteriorment. (on | off)



HTML5 – Formularis

- *method*: especifica com s'envien les dades al servidor. (get | post).
 - get: 1) concatena les dades en la URL. `name=value&name=value`
2) la longitud de la URL està limitada a 3000 caràcters.
3) no utilitzar mai el get per enviar dades sensibles.
4) útil per a enviar formularis en què l'usuari vol guardar la URL de les dades enviades.
 - post: 1) les dades no són visibles en la URL, es troben incorporades en el body del http request.
2) no té limitacions de mida.
3) l'usuari no pot guardar la URL amb les dades enviades.
- *target*: permet indicar on es mostraran les dades retornades pel servidor. En una pàgina nova (`_blank`), en la mateixa pàgina (`_self`)



HTML5 – Formularis

En els formularis en HTML5 existeix el tag **<input>** què és utilitzat per inserir dades per part del client. Aquest tag té un atribut, el **type** què permet definir el tipus d'element de que volem utilitzar per per inserir les dades.

Cada element input també conté l'atribut **name**, necessari per identificar les dades en el servidor.

`<input type="text | password | radio .....">`

html5_formularis1.html

Els input type més rellevants són:

- *text*: camps on l'usuari pot inserir text. L'atribut size permet delimitar la longitud màxima.
- *Password*: defineix un camp password, el text inserit es mostra amb bullets.
- *Radio buttons*: permet al usuari seleccionar només una opció d'un conjunt.



HTML5 – Formularis

- *checkbox*: permet al usuari seleccionar cap o varies opcions.
- *select*: permet al usuari seleccionar un element d'una llista desplegable. Es possible preseleccionar l'opció.
- *submit*: envia les dades del client al servidor. Les dades s'envien a la URL especificada en per l'atribut action del tag form.
- *fieldset* and *legend*: permeten organitzar les dades del formulari. *Fieldset* incorpora un marc i *legend* la llegenda del tipus de dades.

[html5_formularis2.html](#)



¡PREGUNTA D'EXAMEN!

Els tags en HTML tenen definits atributs. Quina és la seva funció?



1. PROGRAMACIÓ WEB

1.1 – DOCUMENTS WEB

- HTML5
- Models de presentació CSS
- Llenguatges especialitzats SVG, SMIL, i RSS

1.2 – LLENGUATGES DE PROGRAMACIÓ

1.3 – ASPECTES BÀSICS DE SEGURETAT





Models de presentació CCS

Els Cascading Style Sheets (CCS), és un llenguatge utilitzat per definir la presentació del document estructurat escrit en HTML o XML i per extensió XHTML.

La idea dels CSS és separar l'estructura del documento de la seva presentació.

Avantatges:

- 1) La tipografia i el layout de la pàgina es poden controlar més fàcilment.
- 2) L'estil es separa de l'estructura.
- 3) Els estils es poden guardar de forma independent.
- 4) Els documents transmesos són més petits.
- 5) El manteniment de la web és més fàcil.



Models de presentació CCS

- Les possibles propietats de declaració per especificar són: **background color**, **color**, **font-family**, **font-size**, **font-style**, **font-weight**, **line-height**, **margin**, **margin-left**, **margin-right**, **text-align**, **text-decoration**, i **width**.

Els monitors mostren el color mitjançant la combinació d'intensitats de Red, Green i Blue, més conegut com a color **RGB**. Cada intensitat té un rang de valors compresos del 0 al 255. Els color també poden ser definits en valor Hexadecimal.

Cal tenir present que depenen del sistema operatiu, navegador i monitor els valors RGB es poden mostrar en diferents intensitats. Per a solventar aquesta problemàtica s'utilitza la **Web Safe Color Palette**, formada per un conjunt de 216 colors que es mostren d'igual manera en els diferents sistemes operatius, navegadors i monitors.

<http://webdevfoundations.net/color/>



Models de presentació. CCS

- La sintaxis dels CSS és relativament senzilla, en general està format per 3 elements:
 1. *Selector*: són els elements de la web que es veuran afectats per la declaració, en poden haver més d'un en cada declaració. El selector pot ser un ID que haguem definit, una classe, o un element del DOM.
 2. *Propietat*: quina és la propietat que volem modificar, no totes les propietats estan disponibles per tots els selectors.
 3. *Valor*: valor que volem assignar a la propietat.

```
selector [, selector 2, selector 3 ....] {  
    propietat : valor;  
    propietat 2: valor;  
}
```



Models de presentació CCS

Hi ha 4 mètodes per incorporar CSS en webs: *inline*, *embedded*, *external*, i *imported*.

1. *Inline*: inclosos en el body de la web com un atribut d'un "tag" d'HTML. L'estil només s'aplica al element que conté el atribut. Útil pels correus electrònics.

[html5_estil_inline.html](#)

2. *Embedded*: definits el el header de la web, s'apliquen a tota la pàgina web.

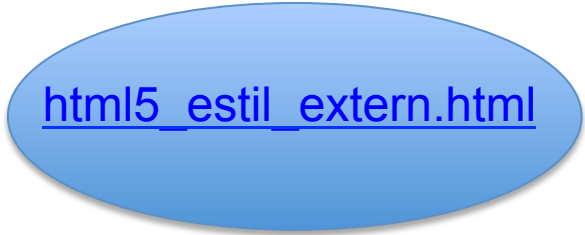
[html5_estil_embedded.html](#)



Models de presentació CCS

Hi ha 4 mètodes per incorporar CSS en webs: *inline*, *embedded*, *external*, i *imported*.

3. *External*: definits en un fitxer extern. Aquest fitxer és localitza utilitzant el “tag” **<link />** definits en el header, l'estil és associat a la pàgina web.



[html5_estil_extern.html](#)

4. *Imported*: similar al external, però el el link fa referència a una pàgina web.



¡PREGUNTA D'EXAMEN!

*Creus què és més interessant utilitzar CSS externs?
justifica la resposta.*



¡PREGUNTA D'EXAMEN!

Tenim un id selector definit en un CSS extern. I el tornem a definir dins de la web (fitxer html), amb valors diferents. Quin serà el resultat al mostrar un element que utilitzi aquest atribut?



1. PROGRAMACIÓ WEB

1.1 – DOCUMENTS WEB

- HTML5
- Models de presentació CSS
- Llenguatges especialitzats SVG, SMIL, i RSS

1.2 – LLENGUATGES DE PROGRAMACIÓ

1.3 – ASPECTES BÀSICS DE SEGURETAT





Llenguatges especialitzats - SVG

Scalable Vector Graphics (SVG): és una especificació per descriure gràfics vectorials bidimensionals, estàtics o dinàmics, utilitzant XML. Es va convertir en una recomanació del W3C el 2001 i avui en dia és suportat per la majoria de navegadors web.

Els objectes geomètrics són objectes definits per un conjunt d'atributs, alguns obligatoris i altres d'opcionals. Es poden definir: **línies, rectangles, cercles, el·lipsis**, o **imatges** (*.jpg, *.png o *.svg). Tots els elements es troben dins d'una **finestra**.

Avantatges:

1. Les imatges SVG poden ser creades i editades amb qualsevol editor de texts.
2. Són escalables.
3. Es poden imprimir a alta qualitat i qualsevol resolució.
4. Es pot fer zoom sense degradació visual.
5. SVG és un estàndard obert.



Llenguatges especialitzats - SVG

- Finestra:

```
<svg width="300" height="120" xmlns="http://www.w3.org/2000/svg" version="1.1"> .....</svg>
```

- Línia:

```
<svg xmlns="http://www.w3.org/2000/svg" width="100%" height="100%">  
<line x1="50" y1="300" x2="250" y2="100" stroke-width="5"/>  
</svg>
```

- Rectangle:

```
<svg xmlns="http://www.w3.org/2000/svg" width="100%" height="100%">  
<rect x="10" y="10" width="150" height="50" fill="#9d1d20" stroke="grey" stroke-width="2"/>  
</svg>
```

- Cercle:

```
<svg xmlns="http://www.w3.org/2000/svg" width="100%" height="100%">  
<circle cx="50" cy="50" r="30" fill="lightblue" stroke="black" stroke-width="2"/>  
</svg>
```



Llenguatges especialitzats - SVG

- El·lipse:

```
<svg xmlns="http://www.w3.org/2000/svg" width="100%" height="100%">  
<ellipse cx="80" cy="50" rx="60" ry="40" fill="lightblue" stroke="black" stroke-width="2"/>  
</svg>
```

- Imatges: Si volem mostrar una imatge raster aquest és mostrada dins d'un rectangle que s'ha de definir previamente.

```
<svg xmlns="http://www.w3.org/2000/svg" width="100%" height="100%">  
<rect x="0" y="0" width="200" height="70" fill="lightgrey"/>  
<image xlink:href="toplogo.gif" x="10" y="10" width="140" height="40"/>  
</svg>
```

Per aplicar la mateixa transformació a tots un conjunt d'elements utilitzarem el “tag” **<g>.....</g>**. Dins d'aquest tag podem posar els elements que volem que continguin les mateixes transformacions.

[svg.html](#)



Una de les prestacions més rellevants d'introduir gràfics amb SVG és el fet de poder fer les donar dinamisme a les imatges.

Per aconseguir aquest efecte cal modificar els elements d'alguns dels atributs de cada objecte geomètric.



[svg_dinamic.html](#)



Llenguatges especialitzats - RSS

Really Simple Syndication (RSS). És un format per publicar informació en un format molt breu basat en XML. El clients es poden subscriure a un canal de RSS i automàticament se'ls avisa de les actualitzacions.

Avantatges:

1. Forma senzilla la manera de visualitzar i compartir contingut web.
2. Permet als usuaris visualitzar de forma ràpida les novetats d'una web.
3. Permet organitzar la informació per part del usuari.

Inconvenients:

1. Els usuaris necessiten d'aplicacions específiques per subscriure's a un canal de notícies.
2. El contingut web és diferent al contingut RSS.

Les que webs que solen utilitzar RSS són:

1. Webs de notícies.
2. Empreses, per llistar nous productes.
3. Calendaris, per mostrar nous esdeveniments.



Els elements bàsics que defineixen un canal RSS són: **<channel>... </channel>**, **<title>...</title>**, **<link>...</link>**, i el **<item>...</item>**.

El **<channel>** el podem veure com una categoria dins un portal. En el cas d'una web de notícies, **<channel>** englobaria tots els **<items>** d'una mateixa categoria.



[noticies.rss](#)



Llenguatges especialitzats - SMIL

Synchronized Multimedia Integration Language (SMIL). És un estàndard del W3C per a presentacions multimèdia. Permet integrar àudio, vídeo, imatges, text o qualsevol altre contingut multimèdia. El problema d'SMIL és que requereix d'un plugin i que hi ha poc suport per als navegadors actuals.



¡PREGUNTA D'EXAMEN!

Quines tecnologies utilitzaries per visualitzar gràfics vectorials de forma dinàmica en una web? Justifica la teva resposta?



1. PROGRAMACIÓ WEB

1.1 – DOCUMENTS WEB

1.2 – LLENGUATGES DE PROGRAMACIÓ

- Llenguatges de la banda del Servidor
- Llenguatges de la banda del Client

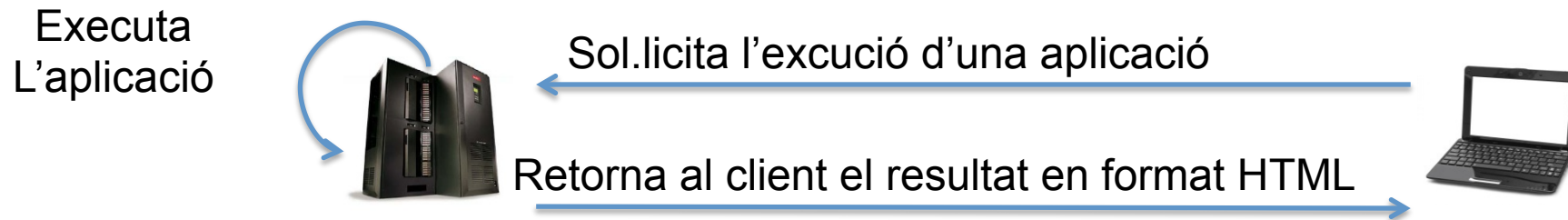
1.3 – ASPECTES BÀSICS DE SEGURETAT





Llenguatges de la banda del Servidor - HTTP CGI

Common Gateway Interface (CGI) és un estàndard que permet la generació de codi HTML en el servidor mitjançant fitxers executables i enviar-lo posteriorment al client. Els programes que s'executen en el servidor solen ser standalone, i poden ser scripts o programes compilats.



- Cada comanda que s'executa, genera un procés què pot consumir molts recursos del servidor.
- Si s'executa el mateix script per molt usuaris de forma concurrent es pot saturar el servidor.
- Una manera de disminuir la carga del servidor és utilitzar llenguatges compilats enlloc d'interpretats.



Llenguatges de la banda del Servidor - PHP

Personal Home Page (PHP) és un llenguatge interpretat pel servidor web que genera una web en HTML que és retornada al client. PHP és inserit en codi HTML.

- Permet la connexió a diferents BD com MySQL, Oracle, PostgreSQL, MicrosoftSQL Server i altres.
- Sintaxi de programació semblant a C i Perl.
- Pot ser executat en Linux i Windows.
- El codi PHP és inserit entre el tag `<?php...?>`.
- LAMP, WAMP, MAMP.



[example.php](#)



Llenguatges de la banda del Servidor - ASP

Active Server Pages (ASP) és una tecnologia de Microsoft, que té una gran similitud amb PHP en quan a funcionalitat.

- La seva sintaxi és molt semblant a Visual Basic i C#.



Llenguatges de la banda del Client - JavaScript

JavaScript és el llenguatge més popular actualment. És el llenguatge per HTML en pàgines web, utilitzat en PCs, portàtils, tablets, smartphones i altres.

- És un llenguatge interpretat molt lleuger.
- Pot ser inserit en codi HTML.
- JavaScript inserit en codi HTML pot ser executat en tots els navegadors web actuals. Per interactuar amb la web JavaScript utilitza una implementació del Document Object Model (DOM).
- JavaScript és un llenguatge molt fàcil d'aprendre.



Llenguatges de la banda del Client - JavaScript

JavaScript permet moltes interaccions amb HTML:

1. Permet escriure en un arxiu HTML.
2. Reaccionar a events.
3. Modificar el contingut del fitxer HTML.
4. Canviar les imatges que visualitza un fitxer HTML.
5. Modificar estils.
6. Validar valors d'entrada.



[javascript.html](#)

El codi JavaScript pot estar situat en:

1. En el tag del head.
2. En el tag del body (vist en el primer exemple de javascript).
3. En un fitxer extern.



[javascripthead.html](#)



[javascriptextern.html](#)



Llenguatges de la banda del Client - AJAX

Asynchronous JavaScript and XML (AJAX): és una tècnica per a crea pàgines web dinàmiques.

AJAX permet que les web s'actualitzin de forma asíncrona intercanviant petites quantitats d'informació amb el servidor. Això permet actualitzar, només, parts de les web sense haver de recarregar tota la web sencera.

Les web que no utilitzen AJAX han de recarregar tota la web quan el contingut és modificat.

Actualment aplicacions com Google Maps, Gmail, Youtube, i Facebook utilitzen AJAX.



[ajax.html](#)



Llenguatges de la banda del Client - Java i Java Applets

Java és un llenguatge de programació d'alt nivell orientat a objectes.

Els **Java Applets** programes Java, és la inclusió de programes Java dins d'altres programes, habitualment pàgines web HTML.

Hello.java

```
// Hello.java
import javax.swing.JApplet;
import java.awt.Graphics;

public class Hello extends JApplet {
    public void paint(Graphics g) {
        g.drawString("Hello, world!", 65, 95);
    }
}
```

Exemple9.html

```
<html>
  <head>
    <title>Exemple Applet</title>
  </head>
  <body>
    <applet code="Hello.class" width="200" height="200">
    </applet>
  </body>
</html>
```

[javaApplet.html](#)



Llenguatges de la banda del Client - Flash

Flash és un llenguatge de programació encarat a animacions, vídeo, i definir webs interactives.

- En llenguatge flash permet manipular imatges vectorials i raster, generant animacions.
- Suporta streaming d'àudio i vídeo de forma bidireccional.
- Permet capturar els inputs del usuari a través del mouse, teclat, càmera o micròfon.
- No està actualitzat per dispositius mòbils degut als recursos computacionals necessaris.



1. PROGRAMACIÓ WEB

1.1 – DOCUMENTS WEB

1.2 – LLENGUATGES DE PROGRAMACIÓ

- Llenguatges de la banda del Servidor
- Llenguatges de la banda del Client

1.3 – ASPECTES BÀSICS DE SEGURETAT





¡PREGUNTA D'EXAMEN!

*Quines són les principals diferències entre llenguatges de la banda del client i del servidor?
Posa un exemple en que sigui convenient l'ús d'un llenguatge de la banda del client i un de la banda del servidor.*



¡PREGUNTA D'EXAMEN!

*Si haguessis de dissenyar una aplicació web per un dispositiu mòbil quins llenguatges utilitzaries?
Justifica la resposta.*



¡PREGUNTA D'EXAMEN!

Tenim un llenguatge de servidor, quina estratègia podem fer servir per reduir la carga del servidor?.



1. PROGRAMACIÓ WEB

1.1 – DOCUMENTS WEB

1.2 – LLENGUATGES DE PROGRAMACIÓ

1.3 – ASPECTES BÀSICS DE SEGURETAT

- Cross-site scripting
- Cross-site request forgery Scripting
- SQL injection





Aspectes bàsics de seguretat.

El 2010 Verizon juntament amb la United States Service Secrets va generar un informe de dades robades a través de la xarxa.

- 40% de les dades robades van ser per hacking.
- 98% de les dades robades procedien de servidors.
- 85% dels atacs van ser considerats fàcils.
- **96% de les dades robades s'haguessin pogut evitar amb mesures fàcils.**

Els dos atacs web més utilitzats són el **Cross-site Scripting**, **Cross-site request forgery Scripting** i el **SQL-injecting**.



Aspectes bàsics de seguretat - Cross-site Scripting

Cross-site Scripting (XSS) explota la confiança que té un usuari en una web.

És una vulnerabilitat web on codi malintencionat és inserit a la banda del client en les pàgines web accessibles per la resta d'usuaris.

De manera què, a través de XSS un persona malintencionada pot inserir codi a una web, de manera que qualsevol usuari, que accedeixi a la web, pot ser afectat pel codi.

- Les dades són robades, normalment, mitjançant un link a URL que conté codi malintencionat.
- L'usuari atacat, clica en aquest link, missatge instantani o simplement llegint una web o un e-mail.
- Habitualment els atacant guarden, la porció del link que conté el codi en HEX.
- Un cop les dades són robades, la web falsa genera una web per l'usuari ataca que conté codi maliciós però en un format que on no és veu que la web hagi estat alterada.



Aspectes bàsics de seguretat - Cross-site Scripting.

- Com podem fer arribar el codi malintencionat al nostre objectiu?
 - Avui en dia les web són totalment dinàmiques. Realitzen moltes peticions gets a altres webs.
 - Un correu electrònic en format HTML.
 - Un post en una xarxa social.
- De manera que tota informació adquirida d'una altre web pot contenir codi maliciós.
 - Script extern: `<script src=http://MisterJaker/flipa.js></script>`
 - Script empotrat: `<script>alert('missatge d'altera')</script>`
 - El tag del body pot contenir un script: `<body onload=alert('missatge d'altera')>`
 - El tag del background pot contenir un script de forma similar al del body: `<body background="javascript:alert('missatge d'altera')>`
 - I la resta de tags com: `<img>...</img>`, `<iframe.....>`, `<link....>`, etc



Aspectes bàsics de seguretat - Cross-site Scripting.

Hi ha dos tipus de categories d'atacs XSS:

- **No persistent:** La URL amb codi malintencionat és enviada al usuari que és volen atacar
- **Persistent:** El codi malintencionat és guardat en una web HTML d'un servidor web. De manera que tots els usuaris que accedeixen a la web HTML alterada del servidor executaran el codi malintencionat.

Com podem protegir-nos d'atacs basats en XSS?

1. Validar i sanejar totes les dades rebudes.
2. Aquests atacs són utilitzats bàsicament per robar cookies, hi ha dues opcions que podem utilitzar:
 - A. Vincular les cookies a l'adreça IP.
 - B. Especificar la cookie com a httponly.



Aspectes bàsics de seguretat - Cross-site request forgery Scripting.

Cross-site request forgery (XSFR) explota la confiança que té una web en un usuari.

Es basa en que un usuari malintencionat envia un link a un usuari que sap que està identificat en una web fent-li executar comandes no desitjades.

`http://bancdolent.com/transaction?to=[ID USUARI]&amount=[QUANTITAT]`

Suposem que els usuaris identificats tenen un forum per deixar les seves consultes, i un usuari malintencionat afegeix el següent

``



Aspectes bàsics de seguretat - Cross-site request forgery Scripting.

Com podem protegir-nos d'atacs basats en XSFR?

1. Requerir una clau secreta cada vegada que fem una operació.
2. Limitar el temps de les sessions.
3. Deshabilitar l'opció de recordar-me.
4. Peticions sense la capçalera **Referer** han de ser rebutjades.



Aspectes bàsics de seguretat - SQL Injection.

SQL Injection s'entén com la inserció de codi SQL en el codi d'una pagina web.

És una vulnerabilitat web on codi malintencionat és inserit a la banda del client per tal de modificar consultes en les BDs emmagatzemades en el servidor.

La idea és modificar les consultes a les BDs i així poder modificar, inserir o eliminar registres, o eliminar taules senceres.

*consulta := "SELECT * FROM usuaris WHERE nom= " + nomUsuari+ ";"*

Si nomUsuari és *Pepet*, no hi haurà cap anomalia, executan-se la seg"uent comanda:

*SELECT * FROM usuaris WHERE nom= 'Pepet';*



Aspectes bàsics de seguretat - SQL Injection.

Ara bé imaginem que enlloc de tenir nomUsuari=Pepet tenim:

"Pepet"; DROP TABLE usuaris; SELECT * FROM dades WHERE nomLIKE '%"

Que donaria lloc a executar les següents instruccions SQL:

```
SELECT * FROM usuaris WHERE nom= 'Pepet';  
DROP TABLE usuaris;  
SELECT * FROM dades WHERE nom LIKE '%';
```

Com podem evitar la SQL Injection?

1. Escapant els caràcters especials

```
$userName = mysql_real_escape_string($userName);  
$query = sprintf("SELECT * FROM usuaris WHERE nom= '%s' $userName);
```



¡PREGUNTA D'EXAMEN!

Describeu un atac basat en cross-scripting i quines tècniques existeixen per evitar-lo.



¡PREGUNTA D'EXAMEN!

Quina diferència hi ha entre un atac persistent i un de no persistent basats en cross-scripting? Posa un exemple de cada.



¡PREGUNTA D'EXAMEN!

En que es basen els atacs de SQL-injection? Quin és el seu principal objectiu? Com els podem evitar?.