



Pràctica 4: Disseny web

Aplicacions i serveis d'internet — iTIC

Francisco del Àguila López Aleix Llusà Serra Alexis López Riera

9 d'abril de 2013

Índex

1	Organització	2
1.1	Objectius	2
1.2	Condicions	2
1.3	Lliurables	2
1.4	Material necessari	2
2	Introducció al llenguatge HTML	2
2.1	Versions d'HTML	3
2.2	Estructura bàsica d'una pàgina web HTML	3
2.3	Etiquetes bàsiques	4
2.4	Fulls d'estil CSS	5
2.5	Eines de desenvolupament	6
2.6	Disseny bàsic de la pàgina web	6
3	Funcionalitats interactives: JavaScript i HTML5	7
3.1	Formularis amb HTML5 i JavaScript	8
3.2	Disseny d'un petit programa en JavaScript	10
3.3	Disseny d'un formulari	11

Resum

Llenguatges web (WWW): HTML, JavaScript

1 Organització

1.1 Objectius

El objectius d'aquesta pràctica són:

1. Assolir nocions bàsiques del llenguatge HTML.
2. Entendre i aprendre a crear fulls d'estil.
3. Incorporar elements interactius a la pàgina web dissenyada, utilitzant JavaScript.
4. Aprendre a crear i a validar formularis en HTML5.

1.2 Condicions

- La pràctica està calibrada per a ésser treballada en equips de dues persones.
- La durada de la pràctica és d'1 setmana.

1.3 Lliurables

Heu d'entregar:

- Una carpeta comprimida on hi haurà tots els fitxers necessaris per poder provar la pàgina web dissenyada amb totes les seves característiques i funcionalitats.

1.4 Material necessari

Per dur a terme la pràctica cal tenir instal·lades eines de navegació web. En el cas de GNU/Debian i equiparables cal que instal·leu els paquets `iceweasel` o `firefox`. També ens caldrà un editor de text per crear els nostres fitxers HTML. Així cal que, també, instal·leu el paquet `emacs`.

TASCA PRÈVIA 1 En aquesta pràctica no caldrà treballar en cap màquina virtual. Per tant, instal·leu els paquets anteriors si no els teniu, en el vostre computador.

2 Introducció al llenguatge HTML

L'HTML(HyperText Markup Language) és un llenguatge que permet descriure hipertext, és a dir, text presentat de forma estructurada per mitjà de paraules clau. En principi direm que HTML no és un llenguatge de programació sinó que és un llenguatge d'etiquetes(tags) que comuniquen al navegador quina és l'informació a mostrar per pantalla i el seu format.

2.1 Versions d'HTML

Des de la creació de l'HTML hi hagut molts canvis en el propi llenguatge. A mesura que passava el temps els diferents navegadors "desenvolupaven" més el llenguatge per oferir més funcions als usuaris, però cadascun a la seva manera. Un petit exemple és que a la versió HTML3(1995) l'Internet Explorer per fer un efecte tenia una etiqueta corresponent i Netscape (l'altre navegador que competia amb Microsoft) en tenia una altre. Això com podeu veure portava molts problemes, ja que si volies dissenyar una web que s'interpretés bé als dos navegadors, havies de fer dues versions diferents de la web.

Al 1998 aquesta guerra entre navegadors es va acabar i va arribar el W3C (World Wide Web Consortium). Aquest consorci és l'encarregat de desenvolupar estàndards i marcar unes regles pel disseny de webs. El seu objectiu és garantir la compatibilitat i acord entre els diferents membres de la indústria a l'hora d'adoptar noves tecnologies web.

Fins ara, la versió més utilitzada d'HTML és la 4.01. Si dissenyem una pàgina amb aquesta versió podem estar molt segurs que la gran majoria de navegadors mostraran el contingut de forma correcta. Dins d'aquesta versió hi ha variants, que són les següents:

- HTML 4.01 Strict: En aquesta versió no es poden utilitzar etiquetes obsoletes. És a dir, etiquetes de versions més antigues.
- HTML 4.01 Transitional: En aquest tipus de document es poden utilitzar les etiquetes de totes les versions HTML(igual a 4.01 o inferiors).
- HTML 4.01 Frameset: En aquest tipus de document es suporten els frames. Els frames són uns marcs en mode de subfinestres dins de la mateixa web. Avui en dia no s'utilitza, ja que hi ha altres formes per dissenyar pàgines web sense fer ús dels frames.

Actualment, es troba la versió HTML5, en mode experimental. Tot i així, molts navegadors web ja la suporten perquè comporta molts avantatges. Amb l'HTML5 s'han creat moltes funcionalitats que estalvien utilitzar en molts casos el JavaScript o fins i tot el Flash.

En aquesta pràctica utilitzarem dos versions d'HTML. Per crear la pàgina principal i la d'imatges utilitzarem la versió 4.01 Transitional. I per crear la pàgina corresponent al formulari utilitzarem la versió HTML5.

Per especificar les diferents versions, al principi del document haurem d'escriure:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

i en el cas del document d'HTML5 simplement:

```
<!DOCTYPE HTML>
```

Per més informació podeu visitar: http://www.w3schools.com/tags/tag_doctype.asp

2.2 Estructura bàsica d'una pàgina web HTML

L'esquelet bàsic d'una pàgina web dissenyada amb HTML és la següent:

```
<!DOCTYPE HTML>  
<html>  
  <head>  
    <title>Títol de la pàgina web</title>
```

```

...
</head>
<body>
  Cos de la pàgina
</body>
</html>

```

En la primera línia es troba una etiqueta relativa a la versió/variant de HTML. En realitat no és necessària, però indica l'estàndard que utilitzem i ens servirà per validar el codi.

A continuació hi ha les etiquetes que indiquen que el text és html. Totes les etiquetes s'han d'obrir i tancar. Tot el contingut que hi ha entremig, pertany en aquest bloc (el quin marca l'etiqueta). La capçalera és la secció compresa entre les etiquetes de head. En ella es troba el títol (entre les etiquetes title). Aquest títol no apareix a la pàgina en si, sinó que és el text que apareix a dalt la finestra del navegador.

El cos de la pàgina està delimitat per les etiquetes body. És important dir que l'etiqueta encarregada d'obrir el cos de la pàgina pot tenir atributs. Els atributs són una espècie d'opcions que s'apliquen al element (etiqueta) on estan col·locats. En el cas del cos, per exemple podríem tenir:

```
< body bgcolor="blue" text="green" background="fitxer_local.gif" >
```

Aquests atributs són:

- **bgcolor** serveix per especificar un color pel fons de la pàgina. El valor d'aquest atribut és pot posar amb els noms en anglés dels colors o en el format: #rrggb. És a dir el caràcter # seguit de 6 lletres o números en l'ordre: Red, Green, Blue.
- **text** serveix per especificar el color del text. És el mateix format que el bgcolor.
- **background** ens permet especificar una imatge pel fons de la pàgina.

2.3 Etiquetes bàsiques

Comentaris en el codi Els comentaris els podem posar de la següent forma:

```
<!-- Comentari -->
```

Paràgrafs Per definir un nou paràgraf s'utilitzen les etiquetes següents: **<p>**El text del paràgraf **</p>**

Salt de línia. Un salt de línia en HTML s'escriu com a: **
**

Formats de text: negreta i cursiva Per escriure un text en negreta, s'ha d'incloure entre les etiquetes: **** i ****. En el cas de la cursiva, les etiquetes són: **<i>**i **</i>**.

Enllaços Per incloure enllaços a altres pàgines web o fitxers, tan externs com locals, s'utilitza l'etiqueta **a**. El format és el següent: ** nom_del_enllaç **. On el **nom_del_enllaç** serà el nom del link que es veurà a la pàgina web.

Llistes Per crear llistes no ordenades, és a dir, sense enumeració, es fa de la següent forma:

```

<ul>
  <li type="circle"> Un element, un cercle com a símbol.</li>
  <li type="square"> Un altre element, amb un quadrat.</li>
  <li type="disc"> Un altre element, amb un cercle(però en aquest cas
    és ple.</li>
</ul>

```

I si volem crear llistes enumerades és d'aquesta forma:

```
<ol>
  <li value="3"> Aquesta llista comença amb valor 3.</li>
  <li> Aquest serà l'element 4.</li>
  <li> Aquest el 5.</li>
</ol>
```

Imatges Per incloure imatges es fa de la següent forma: `` . En aquest exemple es veu l'atribut `alt`. Aquest atribut serveix per especificar el text que es mostrarà si l'imatge no es pot carregar, o mentre es carrega o un cop carregada, es passa el ratolí per sobre.

Hi ha altres atributs que es poden afegir que són interessants, com: `width` que especifica l'amplada de l'imatge en píxels. `height` que serà l'altura de l'imatge. I `border` que especifica l'ample de la línia de contorn de l'imatge (si es posa a 0 equival a no tenir línia de contorn). Per situar les imatges al lloc on vulguem de la pàgina crearem paràgrafs. On a dins d'aquests paràgrafs hi hauran les imatges, encara que no hi ha d'haver obligatòriament text.

2.4 Fulls d'estil CSS

CSS és un llenguatge utilitzat en la presentació de documents HTML. La filosofia de CSS es basa en intentar separar lo que es l'estructura de la pàgina web de la seva presentació. És a dir, l'idea és tenir un fitxer de text HTML on hi hagi el contingut de la web i un fitxer .css que descriu la forma en la qual es presenta la web.

Com ja s'ha vist a l'apartat anterior, s'ha definit algun estil i color en HTML. Nosaltres per fer-ho més senzill, definirem alguns aspectes en el propi codi HTML i d'altres en un fitxer d'estil CSS. Això només és una petita introducció dels fitxers d'estil, ja que poden arribar a ser molt complexes i potents.

Primer de tot, en la pàgina web s'ha d'importar el full d'estil que es vol aplicar. Per importar-lo s'ha de fer dins de la capçalera (`head`) de la següent forma: `<link rel="stylesheet" type="text/css" href="fitxer.css">`

Per entendre com es defineixen els estils amb CSS, anem a veure un exemple:

```
H1 {color:black; font-family:arial; font-size:20pt; TEXT-ALIGN:center}
```

- `h1` és l'etiqueta a la qual s'apliquen les característiques que es defineixen a continuació. Aquesta etiqueta ha d'existir en HTML i tot el text que escrivim dins d'aquesta etiqueta tindrà el format que especifiquem ara, en el full d'estil .css. Abans no s'ha comentat, però les etiquetes d'HTML: `h1,h2,h3,...` serveixen per modificar la mida de la lletra. Ara nosaltres, en el full d'estil podrem especificar per cada etiqueta quin format volem.
- `color` ens especifica el color de la lletra.
- `font-family` ens especifica l'estil de lletra. Els possibles estils de lletra en realitat dependran dels quins tinguem instal·lats al nostre sistema.
- `font-size` ens especifica la mida de la lletra.
- `text-align` ens permet posicionar el text: `left`, `center`, `right`, `middle`, `top` i `bottom`.

2.5 Eines de desenvolupament

Per fer el desenvolupament de les nostres pàgines web, utilitzarem 3 eines diferents:

1. L'editor de text Emacs. Tots els fitxers HTML i css els crearem amb l'emacs, editor que tots vosaltres ja coneixeu.
2. Un validador online d'HTML http://validator.w3.org/#validate_by_upload.
Per assegurar que les pàgines dissenyades, compleixen els estàndards marcats pel W3C, validarem els fitxers a través de la web anterior. Aquesta validació és molt important per assegurar que la nostre web es veurà correctament a tots els navegadors.
3. Eines de desenvolupament del Firefox. El propi navegador Firefox té unes eines pensades pels desenvolupadors de pàgines web. Aneu a *Eines* de Firefox i trobareu una pestanya anomenada: *Desenvolupador web*. En aquesta pestanya activeu *Barra d'eines per a desenvolupadors*. Per fer-ho també podeu prémer les tecles: *Maj+F2*.
Us apareixerà una línia de comandes i al final d'aquesta línia un símbol d'una clau. Si premeu aquest botó, s'obriran les diferents eines.
Per una banda tenim la consola web que ens mostrarà possibles errors que hi ha a la nostre pàgina. Una eina molt interessant es l'Inspector, que ens va mostrant quin fragment de codi HTML corresponent a l'àrea per on passem el ratolí. Pot ser interessant per veure les nostres pàgines, però també per veure com estan dissenyades altres pàgines web que ens agradin.
Una altre eina, que la trobem en forma d'icona (com una llibreta), és el bloc de proves. En aquest bloc de proves podem provar els nostres scripts de JavaScript.

Per ajudar-nos a crear les nostres pàgines web, fulls d'estil, els scripts de JavaScript, etc. consultarem la web (citada ja anteriorment): <http://www.w3schools.com/>.

En aquesta web podem trobar tota l'informació necessària sobre les etiquetes d'HTML, exemples de web o d'scripts, codificació dels caràcters (per exemple, per escriure lletres amb accents) i molta més informació.

2.6 Disseny bàsic de la pàgina web

Un cop adquirides les nocions bàsiques sobre el llenguatge, ja podem fer el nostre primer disseny de la pàgina web. Aquesta pàgina ha de tenir els elements següents:

1. Missatge de benvinguda.
2. Un apartat amb les dades del grup que ha dissenyat la web.
3. Un apartat amb enllaços a pàgines web favorites.
4. Un enllaç a una altra pàgina web (també vostre) que contingui fotografies.

TASCA 2 Dissenyeu dues pàgines web. Una serà la principal i ha de tenir els elements anteriors. I l'altre serà una pàgina web que contindrà fotografies.

TASCA 3 Definiu-vos un únic fitxer de text amb extensió .css que serveixi d'estil per a tots els documents HTML que formen part de la vostra pàgina.

3 Funcionalitats interactives: JavaScript i HTML5

JavaScript és un llenguatge de programació interpretat que permet crear petits programes per realitzar accions dins de l'àmbit d'una pàgina web. Amb aquests programes podem crear efectes especials en les pàgines i interactuar amb l'usuari visitant de la web. El navegador web és l'encarregat d'interpretar les instruccions de Javascript i executar-les per realitzar aquests efectes o interaccions.

Javascript és un llenguatge amb moltes possibilitats, permet la programació de petits scripts com serà el nostre cas, però també permet fer programes més grans, orientats a objectes, estructures de dades complexes, etc.

On s'escriu el codi JavaScript? La programació de JavaScript es realitza dins del propi document HTML. Per tant, en el codi font de la pàgina web hi tindrem els nostres scripts juntament amb la resta del codi HTML. Per fer-ho s'han d'incloure unes etiquetes HTML per delimitar l'script. Aquestes etiquetes són: `<script>` i `</script>`.

Sintaxi del llenguatge El llenguatge JavaScript té una sintaxi molt semblant a la de Java (ja que està basat en ell) i també molt semblant al C.

Per afegir comentaris dins del codi s'utilitza la doble barra: `//` o per fer comentaris de varies línies s'utilitza el signe `/*` per començar i el `*/` per acabar.

En aquest llenguatge s'han de respectar les majúscules i les minúscules.

Per separar instruccions s'utilitza el caràcter `;`.

Com s'executen els scripts? Existeixen dues formes per executar un script de JavaScript: al carregar la pàgina o com resposta a accions de l'usuari.

Si s'executa al carregar la pàgina es pot dir que és una execució directa ja que el navegador quan llegeix la pàgina i troba un script va interpretant les línies de codi seqüencialment.

L'altre forma d'executar scripts és la resposta a esdeveniments. Els esdeveniments són accions que realitza l'usuari i que programes com JavaScript estan preparats per detectar-los i realitzar accions com a resposta. Existeixen molts tipus d'esdeveniment: `onmouseover`, `onmouseout`, `onclick`, etc.

En aquesta pràctica s'utilitzarà l'esdeveniment `oninput` per quan l'usuari estigui situat sobre el camp de confirmar la contrasenya. Per fer ús d'esdeveniments en HTML s'han d'afegir com atributs: `<etiqueta_HTML ... esdeveniment=instrucció>`. On *instrucció* pot ser la crida a una funció creada amb JavaScript.

Per tant l'idea és que quan l'usuari estigui escrivint en el camp de confirmació de password, es cridi una funció que comprovi si aquell camp és igual al del password.

Funcions Les funcions en JavaScript es declaren com: `function nom_funcio(paràmetres){ instruccions de la funció}`. Hem d'estar alerta on escriure la declaració ja que ho hem de fer abans d'on estigui la crida.

Mètodes del JavaScript Per fer que un script mostri algun missatge escrit a la pàgina, es pot utilitzar el mètode `write` que pertany a l'objecte `document`. Per tant, si fem: `document.write('El text que volem escriure')` podrem escriure a la pàgina el que vulguem.

Si en canvi, volem fer sortir un missatge emergent d'alerta, tenim el mètode `alert` que pertany a `window`, i només cal fer: `alert('El missatge')`.

Per demanar una entrada a l'usuari de forma emergent, tenim el `prompt`.

En aquesta pràctica els quins utilitzarem seran: `value`, `getElementById(id)`, `setCustomValidity('')`. El `value` ens permet obtenir el valor del objecte en qüestió. El `getElementById(id)` ens

permet obtenir l'objecte amb un id concret. I el `setCustomValidity('missatge')` que estableix un missatge de validació personalitzat pel objecte on s'aplica(en el nostre cas un camp de formulari).

3.1 Formularis amb HTML5 i JavaScript

Els formularis són una característica de l'estàndard HTML. Permeten als autors recollir informació dels visitants de la web. Aquests formularis poden resultar útils per reunir informació personal, de contacte, opinions, etc.

Tal com s'ha dit, amb l'HTML5, en molts casos ja no ens caldrà utilitzar scripts de JavaScript. Un exemple clar d'això és la validació de formularis.

La validació de formularis serveix per comprovar que les dades que ha entrat l'usuari són en el format correcte i compleixen les especificacions que es demanen i que no hi hagi camps en blanc(quan són camps obligatoris). Aquestes validacions han sigut un problema que sempre s'ha solucionat amb l'utilització de Javascript, ja sigui amb codi propi o llibreries creades per tercers. Però amb l'arribada del HTML5 es poden validar molts camps del formulari sense fer ús de cap script.

En aquesta pràctica dissenyarem un formulari i farem la seva validació amb HTML5 i utilitzarem JavaScript només per validar si dos camps són iguals.

A continuació es mostra un formulari com a exemple:

```
<!DOCTYPE HTML>
<html>
  <head>
    <TITLE>Formulari amb HTML5</TITLE>
    <script>
      function check(input) {
        if (input.value != document.getElementById('correu').value) {
          input.setCustomValidity('El correu ha de ser el mateix!');
        }
        else{
          // input is valid -- reset the error message
          input.setCustomValidity('');
        }
      }
    </script>
  </head>
  <body>
    <form id="FORMULARI" action="" method="">
      <br>
      <label>Nom d'usuari: </label>
      <input name="Nom d'usuari" id="nom_usuari" type="text"
        required="required"/>
      <br>
      <label>Correu electr&ograve;nic: </label>
      <input name="correu" id="correu" type="email"
        required="required"/>
      <br>
      <label>Repetir correu electr&ograve;nic: </label>
      <input name="Repeticio_correu" id="rep_correu" type="email"
        oninput="check(this)"/>
      <br>
      <label>Província: </label>
      <input name="província" id="província" list="prov" type="text"/>
    </form>
  </body>
</html>
```



```

<datalist id="prov">
  <option value="Barcelona">
  <option value="Girona">
  <option value="Lleida">
  <option value="Tarragona" >
</datalist>
<br>
<br>
<input type="submit" value="Enviar" id="boto_enviar"/>
<input type="reset" value="Esborrar" id="boto_reset"/>
</form>
</body>
</html>

```

En aquest formulari tenim un script que comprova que el correu electrònic és el mateix que el del camp de confirmació del correu.

Pel que fa al cos del formulari, al obrir l'etiqueta de `form` hi ha uns atributs que en el nostre cas omplirem de forma correcta. L'atribut `id` simplement identifica el formulari dins del text HTML. L'`action` és l'acció que es durà a terme quan enviem el formulari i el `method` serveix per especificar com serà enviada l'informació del formulari. En aquesta pràctica, podeu deixar aquests dos atributs sense cap valor, ja que a la pròxima pràctica els omplirem correctament.

Per cada camp que vulguem crear i mostrar a la web, haurem d'especificar el text que volem que es mostri davant del rectangle que omple el visitant mitjançant l'etiqueta `label`. A continuació vindrà l'entrada al formulari. Hi ha diferents elements per fer entrades de formulari: `input`, `button`, `checkbox`, `datalist`, `textarea`, etc. A continuació es descriuen els quins utilitzarem en aquesta pràctica:

- L'`input` permet l'entrada de molt tipus de dades de l'usuari. Amb l'atribut `type` s'especifica el tipus de dada i el navegador ho mostrarà d'una forma o una altra. Uns dels valors possibles per aquest atribut són: `text`, `number`, `email`, `password`, `search`, `url`, `tel`, `date`, `submit`, `reset`, etc.

Gràcies a l'HTML5, escrivint un d'aquests valors per `type` fa que comprovi si l'entrada és amb el format correcte. En els casos dels Smartphones o Tablets amb navegadors compatibles amb HTML5 farà que en cada cas mostri el teclat que toca (per exemple si és de tipus `number`, ens mostrarà directament el teclat numèric).

Un dels atributs nous per l'`input` és el `required` que serveix per indicar que aquella entrada és obligatòria. És a dir que, per validar els camps obligatoris d'un formulari és tan senzill com escriure: `required="required"`.

Un altre atribut interessant és el `pattern` que serveix per especificar limitacions més concretes de l'entrada d'aquell camp. Aquest atribut el podeu utilitzar voluntariament en aquesta pràctica.

Si volem escriure un valor inicial en un dels camps o per exemple escriure un nom sobre d'un botó, utilitzarem l'atribut `value`.

- El `datalist` serveix per oferir una llista de possibles respostes. L'estructura és:

```

<datalist id="xx">
  <option value="a" >
  <option value="b" >
</datalist>

```

Per relacionar aquesta llista a l'entrada d'un camp, s'ha de crear un input corresponent i afegir com atribut: *list="xx"*, on el valor, és l'id del datalist.

- Una textarea és una zona on s'hi poden deixar comentaris. Té la mateixa estructura que un input, però canviant el nom de l'etiqueta. Els atributs interessants d'aquest camp són rows i cols que serviran per limitar el número de files i columnes de la zona de text.

3.2 Disseny d'un petit programa en JavaScript

Observeu el codi següent:

```
<!DOCTYPE HTML>
<html>
<head>
<TITLE>Programa Javascript</TITLE>

<script>

function loadXMLDoc()
{
  var xmlhttp;
  var objecte_div;
  objecte_div=document.getElementById("myDiv");
  xmlhttp=new XMLHttpRequest();
  xmlhttp.onreadystatechange=function()
  {
    if (xmlhttp.readyState==4 /*&& xmlhttp.status==200*/)
    {
      objecte_div.innerHTML=xmlhttp.responseText;
      objecte_div.style.color="green";
    }
  }
  xmlhttp.open("GET","entrada.txt",true);
  xmlhttp.send();
}

</script>

</head>
<body>

<div id="myDiv" style="color:blue"> Hola!!!!!! </div>

<script>
<!--Cada 3 segons, crida la funció loadXMLDoc()-->
setInterval("loadXMLDoc()",3000);
</script>

</body>
</html>
```

En aquest programa de JavaScript podem veure dues funcionalitats més, d'aquest llenguatge:

- Per una banda, al final del codi podeu veure un petit script, amb la crida a una funció predefinida `setInterval`. Aquesta funció serveix per temporitzar una acció. En aquest cas, cada 3000ms cridarà una funció anomenada `loadXMLDoc()`.

- Per altre banda tenim la definició de la funció `loadXMLDoc()`. Per entendre aquesta funció ens hem de situar al cas real de quan aquest script s'està executant sobre un servidor. És a dir, un usuari-client obre la pàgina web d'un servidor on hi ha aquest script. Si l'usuari, mitjançant aquest script, consulta un fitxer (que pot estar en el mateix directori on hi ha la web o en una altra URL del mateix servidor) ho necessita fer amb una connexió *HTTP*. El que fa aquesta funció, doncs, és fer un *get* cap a una URL on hi ha el fitxer que volem consultar.

Un cop consultat, escriu el seu contingut a la pàgina web a través de l'element `div`. En la mateixa funció, també, es canvia el color del text d'aquest element.

Per més informació podeu consultar:

<http://www.w3schools.com/ajax/>

<http://en.wikipedia.org/wiki/XMLHttpRequest>

Cal destacar que en aquesta pràctica no provarem aquesta pàgina web amb el script sobre cap servidor. Obrirem la pàgina web localment(perquè estarà en el nostre ordinador). Per tant no té sentit parlar de cap connexió *HTTP*. Fixeu-vos en la línia que hi ha una part comentada: `if (xmlhttp.readyState==4 /*&& xmlhttp.status==200*/)`. Com que no fem servir una connexió *HTTP*, amb la comanda *get* no obtindrem un 200(que equival a un OK); per això hi ha aquesta part comentada. Quan muntem el servidor, on allotjarem la nostre web, haurem d'afegir aquesta part que ara està comentada.

Per provar aquest codi, haureu de crear un fitxer de text anomenat: *entrada.txt* en el directori on tingueu la pàgina web amb l'script anterior.

Imagineu que aquest fitxer simplement conté un 0 o un 1. Ara haureu de modificar l'script anterior i fer que si hi ha un 0 en el fitxer, el contingut de la pàgina es mostri de color blau i si hi ha un 1, de color verd. (Ara el text sempre serà *Hola!!!!!!*, només ha de canviar el color)

TASCA 4 Acabeu d'entendre com funciona l'script anterior i modifiqueu-lo en un nou fitxer HTML, fent que es comporti tal com s'ha especificat. Afegiu l'enllaç d'aquesta nova pàgina a la vostra pàgina principal.

3.3 Disseny d'un formulari

Ara en la vostra pàgina principal hi afegireu un nou enllaç: *Sollicitud d'amistat*. Aquest enllaç ens portarà a una altra pàgina on hi haurà un formulari per a la introducció de les dades de la persona que visita la vostra pàgina. El formulari ha de permetre introduir les següents dades:

- Nom d'usuari
- Password
- Confirmació de password
- Sexe: escollir entre home i dona (*datalist*)
- Correu electrònic
- Comentaris, mitjançant una àrea de text
- Botons *Enviar* i *Esborrar*

Incorporeu el següent conjunt de validacions al formulari de la vostra pàgina personal:

1. Els camps *Nom d'usuari*, *Password* i *Correu electrònic* no poden estar en blanc.
2. S'ha de verificar que *Confirmació de password* coincideix amb *Password*.
3. S'ha de verificar que el *Correu electrònic* té el format correcte.
4. (Opcional) Busqueu informació sobre l'atribut `pattern` i limiteu els formats possibles de *Nom d'usuari* i/o *Password*.

TASCA 5 Dissenyeu un nou fitxer HTML que contingui aquest formulari i afegiu l'enllaç a la vostra pàgina principal.