



Pràctica 1: Disseny web

Interfícies d'usuari — iTIC

Sebastià Vila-Marta

Aleix Llusà Serra

16 de febrer de 2017

Índex

1	Organització	2
1.1	Objectius	2
2	HTML	2
2.1	Estructura	2
2.2	El menú de navegació	3
2.3	Validació	3
3	CSS	3
3.1	Estil	4
3.2	Un clàssic: el menú de navegació	4
4	Responsive web design	5
4.1	Frameworks	5
4.2	El menú de navegació responsive	5
5	JS	6
5.1	DOM	6
5.1.1	El menú de navegació clicable	7
5.2	BOM	7
5.3	Un clàssic: el carrusel	7
6	Templating	8
6.1	TAL	8
6.1.1	Una plantilla pel menú de navegació	8
6.1.2	Plantilles TAL amb macros	10
6.2	Pyramid	12
6.2.1	La primera plantilla	12
6.2.2	Les plantilles amb macros	13

Resum

Llenguatges web: HTML, CSS, JS

1 Organització

En aquesta pràctica dissenyarem una interfície web per a una aplicació de xarxa social. Aquells que hagueu seguit l'assignatura de Bases de Dades podeu continuar la pràctica de xarxa social. Aquells que hagueu seguit l'assignatura d'Aplicacions i Serveis d'Internet podeu continuar la pràctica de disseny web i la de xat.

1.1 Objectius

Els objectius d'aquesta pràctica són:

TASCA PRÈVIA 1 Llegeix la següent introducció al disseny web i situeu la funció de cadascun dels llenguatges que hi intervenen.

<https://docs.webplatform.org/wiki/Beginners>

2 HTML

Referència principal: <https://docs.webplatform.org/wiki/html>

En aquesta secció dissenyem la part HTML de la interfície. Utilitzem el llenguatge HTML per a descriure l'estructura dels documents web que formen part de la interfície.

El llenguatge HTML descriu semànticament els continguts dels documents web. És important descriure correctament els documents perquè tant poden visualitzar-se en navegadors diversos com ser interpretats per altres agents d'usuari automàtics, com per exemple els indexadors web.

2.1 Estructura

Penseu quantes pàgines web ha de tenir la interfície d'usuari per a la xarxa social, quines relacions hi ha entre elles i quina estructura HTML té cadascuna.

Per exemple, hi ha d'haver:

- Una capçalera per a posar-hi el logo i el títol de la pàgina
- Una barra de navegació
- Un fil d'Ariadna
- Una pàgina d'entrada d'usuari
- Un formulari per a editar les dades personals
- Un formulari per a afegir noves amistats
- Una pàgina per veure les amistats
- Accions per a acceptar o rebutjar les amistats
- Un peu de pàgina amb informació legal i detalls de contacte

Podeu orientar-vos amb els elements d'estructura d'HTML5 https://docs.webplatform.org/wiki/guides/html_structural_elements

TASCA 2 Implementeu els documents HTML per a la vostra interfície d'usuari. En aquest apartat implementeu **només** codi HTML, és a dir l'estructura dels documents. En els apartats següents ja definireu els estils dels documents.

2.2 El menú de navegació

Us proposem un codi HTML amb el menú de navegació:

```
<!DOCTYPE html>
<html>
<head>

<meta charset="UTF-8" />
<title>El menú de navegació</title>

</head>
<body>

<nav>
<ul>
  <li class="actual"><a href="inici.html">Inici</a></li>
  <li><a href="amistats.html">Amistats</a></li>
  <li><a href="dades.html">Dades</a></li>
  <li class="dreta"><a href="ajuda.html">Ajuda</a></li>
</ul>
</nav>

</body>
</html>
```

2.3 Validació

Un cop hagueu dissenyat els documents HTML, valideu que segueixen l'estàndard HTML i que per tant són interpretats correctament en qualsevol navegador.

TASCA 3 Llegiu la documentació següent i valideu el vostre codi

https://docs.webplatform.org/wiki/guides/html_validation

3 CSS

Referència principal: <https://docs.webplatform.org/wiki/css>

En aquesta secció dissenyem la part CSS de la interfície. Utilitzem el llenguatge CSS per a descriure l'estil de presentació dels documents web que formen part de la interfície.

3.1 Estil

Penseu un estil per a la interfície d'usuari de la xarxa social. Dibuixeu un esquema amb el format visual i la posició a la pàgina per a cadascun dels elements de l'estructura.

TASCA 4 Implementeu el CSS per als documents HTML de l'apartat anterior. En aquest apartat implementeu **només** codi CSS, sense usar JavaScript. Si us cal, però, podeu retocar el codi HTML que hagueu dissenyat anteriorment; segurament us caldrà afegir identificadors o classes.

3.2 Un clàssic: el menú de navegació

Us proposem un estil per al menú de navegació i un codi CSS que l'implementa. Referències: https://docs.webplatform.org/wiki/tutorials/creating_basic_navigation_menus i http://www.w3schools.com/css/css_navbar.asp.



```
nav ul {
  background-color: #333;
  padding: 0;
}

nav li {
  display: inline;
}

nav a {
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  display: inline-block;
}

nav a:hover {
  background-color: #111;
}

nav li.actual a {
  background-color: #4CAF50;
}

nav li.dreta {
  float: right;
}
```

Només us cal desar el codi CSS en un fitxer `exemple.css` i incloure el codi següent al `head` de tots els documents HTML:

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

TASCA 5 Personalitzeu el menú al vostre gust. Per exemple podeu afegir elements desplegable http://www.w3schools.com/howto/howto_css_dropdown.asp.

4 Responsive web design

Referència principal: https://docs.webplatform.org/wiki/concepts/mobile_web/responsive_design

Visiteu les pàgines web que hagueu dissenyat amb navegadors de diversos dispositius: mòbil, escriptori, etc. O bé proveu de redimensionar el navegador, des de la mínima amplada fins a la màxima. Creieu que és usable la interfície en totes les formes?

L'objectiu del disseny web responsive és aconseguir una bona usabilitat de la web en qualsevol dispositiu. Com a requeriment, una web responsive s'adapta dinàmicament al dispositiu en comptes de tenir un document diferent per a cada dispositiu. S'usen tècniques d'HTML5 i de CSS3, tot i que per adaptar a navegadors antics també es complementa amb JavaScript. Principalment s'utilitza la tècnica de *media queries* de CSS3 i es defineixen les mides de forma relativa en comptes de mides fixes.

4.1 Frameworks

Referències: <http://www.w3schools.com/w3css/default.asp> i <http://fontawesome.io>

Per a la implementació d'estils web se solen utilitzar frameworks, per exemple Bootstrap o Materialize. En aquest apartat utilitzem el framework W3.CSS, que és responsive nativament i utilitza quasi únicament CSS, i la biblioteca d'icones Font Awesome.

TASCA 6 Llegiu les documentacions dels frameworks i utilitzeu-los com a ajuda per a implementar els vostres estils.

4.2 El menú de navegació responsive

Us proposem un codi CSS responsive per al menú de navegació. El menú és el mateix que abans però per a resolucions petites es mostra col·lapsat en una icona. Referència: http://www.w3schools.com/w3css/w3css_navigation.asp.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>El menú de navegació respoonsive</title>

<meta name="viewport" content="width=device-width, initial-scale=1" />
<link rel="stylesheet" href="http://www.w3schools.com/lib/w3.css" />
<link rel="stylesheet" href="http://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.4.0/css/
font-awesome.min.css">

<style>

@media (max-width:600px){
  .hide-small { display:none;}
```

```

    input.boto-menu:checked ~ li.hide-small { display: block; }
    #mainnav label { cursor: pointer;}
}

input.boto-menu {display: none;}

</style>

</head>
<body>

<nav id="mainnav">

<ul class="w3-navbar w3-large w3-black w3-left-align">
  <li class="w3-hide-medium w3-hide-large">
    <label class="w3-hover-grey w3-btn" for="mainnav-boto">
      <i class="fa fa-bars"></i> Menú
    </label>
  </li>
  <input id="mainnav-boto" class="boto-menu" type="checkbox"/>

  <li class="hide-small w3-green"><a href="inici.html">Inici</a></li>
  <li class="hide-small"><a href="amistats.html">Amistats</a></li>
  <li class="hide-small"><a href="dades.html">Dades</a></li>
  <li class="hide-small w3-right"><a href="ajuda.html">Ajuda</a></li>
</ul>

</nav>

</body>
</html>

```

5 JS

Referència principal: <https://docs.webplatform.org/wiki/javascript>

En aquesta secció dissenyem la part JavaScript de la interfície. Utilitzem el llenguatge JavaScript per a programar comportaments dels documents web que són difícils o impossibles d'implementar amb HTML i CSS. En aquesta secció ens referim al JavaScript que s'executa com a script a la banda del client. JavaScript, però, és també un llenguatge que s'utilitza a la banda de servidor.

5.1 DOM

Referència principal: <https://docs.webplatform.org/wiki/dom>

L'*HTML Document Object Model* (DOM) és una interfície de programació d'aplicacions per als documents web. Defineix un model estàndard dels documents web i té una interfície de programació que permet accedir i canviar l'estructura, el contingut i l'estil dels documents. El DOM té estructura d'arbre i cada element d'un document és un *objecte* en l'arbre.

Mitjançant JavaScript es pot interactuar amb el DOM per a programar documents web dinàmics.

5.1.1 El menú de navegació clicable

Us proposem un canvi en el menú de navegació anterior. Ara el collapse del menú s'acciona mitjançant codi JavaScript.

```
<nav id="mainnav">

<ul class="w3-navbar w3-large w3-black w3-hide-medium w3-hide-large">
  <li class="w3-hover-grey w3-btn" onclick="document.getElementById('mainnav-ul').style.
    display='block'"><i class="fa fa-bars"></i> Menú
  </li>
</ul>

<ul id="mainnav-ul" class="w3-navbar w3-large w3-black w3-left-align hide-small">
  <li class="w3-green"><a href="inici.html">Inici</a></li>
  <li><a href="amistats.html">Amistats</a></li>
  <li><a href="dades.html">Dades</a></li>
  <li class="w3-right"><a href="ajuda.html">Ajuda</a></li>
  <li class="w3-hide-medium w3-hide-large w3-btn"><a href="" onclick="document.getElementById
    ('mainnav-ul').style.display='none'" class="w3-center-align">Close X</a></li>
</ul>
</nav>
```

5.2 BOM

El *Browser Object Model* (BOM) és una interfície de programació per als objectes dels navegadors web. A diferència del DOM, no hi ha un estàndard BOM sinó que cada navegador pot implementar-lo lliurament.

Mitjançant el BOM es pot, per exemple, redimensionar la finestra del navegador, canviar el text de la barra d'estat, mostrar finestres emergents, executar funcions diferides o gestionar les cookies.

Per exemple, el codi següent mostra un avís al cap de 5 segons d'haver visitat la pàgina.

```
function avis() {
  alert("Recordeu sempre de llegir les documentacions");
}

setTimeout(avis, 5000);
```

5.3 Un clàssic: el carrusel

Us proposem un codi JS per a mostrar un *carrusel* a la pàgina. En un carrusel es mostren cíclicament imatges, notícies o altres tipus de contingut que es vol destacar. Referència: http://www.w3schools.com/howto/howto_js_slideshow.asp

```
<div class="carrusel-container">
  <div class="carrusel-element">
    
  </div>

  <div class="carrusel-element">
    
  </div>
</div>
```

```

</div>

<div class="carrusel-element">
  
</div>
</div>
<script>
var slideIndex = 0;
showSlides();

function showSlides() {
  var i;
  var slides = document.getElementsByClassName("carrusel-element");
  for (i = 0; i < slides.length; i++) {
    slides[i].style.display = "none";
  }
  slideIndex++;
  if (slideIndex > slides.length) {slideIndex = 1}
  slides[slideIndex-1].style.display = "block";
  setTimeout(showSlides, 2000);
}
</script>

```

6 Templating

El *templating web* és una manera de definir l'estructura dels documents web mitjançant plantilles. Una plantilla té una estructura similar als documents que s'han de visualitzar però conté variables en comptes de les dades concretes de cada document. Així, cada document final que es visualitza és una plantilla renderitzada amb unes dades en concret.

Hi ha diversos llenguatges de plantilles i diverses eines per a renderitzar-les. En aquest apartat utilitzem el llenguatge *Template Attribute Language* (TAL) i renderitzem les plantilles mitjançant Pyramid [<http://www.pylonsproject.org/>].

6.1 TAL

Referència principal: <http://chameleon.readthedocs.io/en/latest/reference.html>

TAL és un llenguatge que serveix per a dissenyar plantilles d'HTML o de XML i alhora que aquestes plantilles siguin vàlides com a documents HTML o XML.

6.1.1 Una plantilla pel menú de navegació

Us proposem una plantilla TAL pel document anterior de menú de navegació. En aquesta plantilla, que anomenem `plantilla1.pt`, hi ha tres variables:

- *titol* per a configurar tant el títol de la capçalera com el del cos,
- *menu.actiu* per a inserir o no el menú de navegació i
- *menu.actual* per a senyalar amb color verd quin és l'element actiu del menú de navegació.


```

<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:tal="http://xml.zope.org/namespaces/tal"
      >
<head>
<meta charset="UTF-8">
<title>${titol}</title>

<meta name="viewport" content="width=device-width, initial-scale=1" />
<link rel="stylesheet" href="http://www.w3schools.com/lib/w3.css" />
<link rel="stylesheet" href="http://cdnjs.cloudflare.com/ajax/libs/font-awesome/
4.4.0/css/font-awesome.min.css">

<style>

@media (max-width:600px){
  .hide-small { display:none;}
  input.boto-menu:checked ~ li.hide-small { display: block; }
  #mainnav label { cursor: pointer;}
}

input.boto-menu {display: none;}

</style>

</head>
<body>

<header>
<h1>${titol}</h1>
</header>

<nav id="mainnav" tal:condition="menu.actiu" >

<ul class="w3-navbar w3-large w3-black w3-hide-medium w3-hide-large">
  <li class="w3-hover-grey w3-btn" onclick="document.getElementById('mainnav-ul').style.
display='block'"><i class="fa fa-bars"></i> Menú
  </li>
</ul>

<ul id="mainnav-ul" class="w3-navbar w3-large w3-black w3-left-align hide-small">
  <li tal:attributes="class 'w3-green' if menu.actual=='inici' else ''"><a href="inici.html">
Inici</a></li>
  <li tal:attributes="class 'w3-green' if menu.actual=='amistats' else ''"><a href="amistats.
html">Amistats</a></li>
  <li tal:attributes="class 'w3-green' if menu.actual=='dades' else ''"><a href="dades.html">
Dades</a></li>
  <li tal:attributes="class 'w3-right w3-green' if menu.actual=='ajuda' else 'w3-right'" ><a
href="ajuda.html">Ajuda</a></li>
  <li class="w3-hide-medium w3-hide-large w3-btn"><a href="" onclick="document.getElementById
('mainnav-ul').style.display='none'" class="w3-center-align">Close X</a></li>
</ul>
</nav>

</body>
</html>

```

6.1.2 Plantilles TAL amb macros

Un recurs útil en les plantilles és el d'utilitzar macros per compartir trossos de codi comuns entre plantilles. El llenguatge que acompanya TAL per a definir macros és el *Macro Expansion Template Attribute Language* (METAL).

Us proposem una plantilla METAL amb els elements comuns que hi ha a la `plantilla1.pt` anterior i amb uns buits per tal que la macro sigui configurable. En aquesta plantilla macro, que anomenem `master.pt`, hi ha les variables següents:

- *titol* per a configurar el títol de la capçalera,
- *slot header* per a configurar la primera part del cos,
- el menú de navegació amb les variables ja vistes anteriorment i
- *slot content* per a configurar la part del contingut.

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:tal="http://xml.zope.org/namespaces/tal"
      xmlns:metal="http://xml.zope.org/namespaces/metal"
      metal:define-macro="base">

<head>
<meta charset="UTF-8">
<title>${titol}</title>

<meta name="viewport" content="width=device-width, initial-scale=1" />
<link rel="stylesheet" href="http://www.w3schools.com/lib/w3.css" />
<link rel="stylesheet" href="http://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.4.0/css/
font-awesome.min.css">

<style>

@media (max-width:600px){
  .hide-small { display:none;}
  input.boto-menu:checked ~ li.hide-small { display: block; }
  #mainnav label { cursor: pointer;}
}

input.boto-menu {display: none;}

</style>

</head>
<body>

<header metal:define-slot="header">
<h1>El títol</h1>
</header>

<nav id="mainnav" tal:condition="menu.actiu" >

<ul class="w3-navbar w3-large w3-black w3-hide-medium w3-hide-large">
  <li class="w3-hover-grey w3-btn" onclick="document.getElementById('mainnav-ul').style.
display='block'"><i class="fa fa-bars"></i> Menú
  </li>
```

```

</ul>

<ul id="mainnav-ul" class="w3-navbar w3-large w3-black w3-left-align hide-small">
  <li tal:attributes="class 'w3-green' if menu.actual=='inici' else ''"><a href="inici.html">
    Inici</a></li>
  <li tal:attributes="class 'w3-green' if menu.actual=='amistats' else ''"><a href="amistats.
    html">Amistats</a></li>
  <li tal:attributes="class 'w3-green' if menu.actual=='dades' else ''"><a href="dades.html">
    Dades</a></li>
  <li tal:attributes="class 'w3-right w3-green' if menu.actual=='ajuda' else 'w3-right'" ><a
    href="ajuda.html">Ajuda</a></li>
  <li class="w3-hide-medium w3-hide-large w3-btn"><a href="" onclick="document.getElementById
    ('mainnav-ul').style.display='none'" class="w3-center-align">Close X</a></li>
</ul>
</nav>

<div metal:define-slot="contingut">
</div>

</body>
</html>

```

I aleshores ja podem definir plantilles que utilitzin la macro.

Per exemple definim una plantilla, que anomenem `inici.pt`, que utilitza la macro i l'omple amb el contingut de la pàgina d'inici:

```

<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:tal="http://xml.zope.org/namespaces/tal"
      xmlns:metal="http://xml.zope.org/namespaces/metal"
      metal:use-macro="main.macros['base']"
>
<head>
<body>

<header metal:fill-slot="header">
<h1>${titol}</h1>
</header>

<div metal:fill-slot="contingut">
Sou a l'inici.
</div>

</body>
</html>

```

i definim una altra plantilla, que anomenem `amistats.pt`, que utilitza la macro i l'omple amb el contingut de les amistats:

```

<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:tal="http://xml.zope.org/namespaces/tal"
      xmlns:metal="http://xml.zope.org/namespaces/metal"
      metal:use-macro="main.macros['base']"
>
<head>
<body>

```

```

<header metal:fill-slot="header">
<h1>${titol}</h1>
</header>

<div metal:fill-slot="contingut">
Aquí hi ha les amistats.
</div>

</body>
</html>

```

6.2 Pyramid

Pyramid és un web framework que es programa en Python orientat a publicar objectes. En aquest apartat primer seguim el tutorial bàsic de Pyramid i després programem una aplicació per a renderitzar les plantilles TAL anteriors.

TASCA PRÈVIA 7 Seguiu el tutorial inicial de Pyramid <http://docs.pylonsproject.org/projects/pyramid/en/latest/narr/firstapp.html>. No continueu als punts següents fins que pugueu visitar una pàgina com per exemple <http://localhost:8080/hello/itic> i obtingueu resposta.

6.2.1 La primera plantilla

Referència principal: <http://docs.pylonsproject.org/projects/pyramid-chameleon/en/latest/>

Us proposem el programa Python següent per a renderitzar la `plantilla1.pt`, de moment és una aplicació Pyramid sense estructura i amb tot el codi en un sol mòdul:

```

import os
from wsgiref.simple_server import make_server
from pyramid.config import Configurator
from pyramid.renderers import get_renderer

_here = os.path.dirname((os.path.realpath(__file__)))

def exemple_plantilla1(request):
    return {
        'titol': 'Plantilles TAL',
        'menu': {
            'actiu': True,
            'actual': 'inici',
        }
    }

```

```

if __name__ == '__main__':
    config = Configurator()
    config.include('pyramid_chameleon')

    config.add_route('plantilla1', '/plantilla1')
    config.add_view(exemple_plantilla1, route_name='plantilla1', renderer=_here+'/plantilla1.pt')

    app = config.make_wsgi_app()
    server = make_server('0.0.0.0', 8080, app)
    server.serve_forever()

```

Fixeu-vos que definim una ruta `/plantilla1` que renderitza el fitxer TAL `plantilla1.pt` mitjançant les variables que retorna la funció `exemple_plantilla1` en un diccionari.

Podeu executar el programa amb la variable d'entorn `PYRAMID_RELOAD_TEMPLATES=1` de manera que no us caldrà reiniciar el servidor per a refrescar qualsevol canvi en les plantilles.

6.2.2 Les plantilles amb macros

Per al segon cas de plantilles amb macro, us proposem d'estendre el programa anterior amb el codi següent:

```

[...]  

from pyramid.renderers import get_renderer  

[...]  

def inici(request):  

    main = get_renderer(_here+'/master.pt').implementation()  

    return {  

        'main': main,  

        'titol': 'Plantilles TAL amb macro',  

        'menu': {  

            'actiu': True,  

            'actual': 'inici',  

        }  

    }  

  

def amistats(request):  

    main = get_renderer(_here+'/master.pt').implementation()  

    return {  

        'main': main,  

        'titol': 'Amistats',  

        'menu': {  

            'actiu': True,  

            'actual': 'amistats',  

        }  

    }

```

```

    }

if __name__ == '__main__':
    [...]

    config.add_route('inici', '/inici.html')
    config.add_view(inici, route_name='inici', renderer=_here+'/inici.pt')
    config.add_route('amistats', '/amistats.html')
    config.add_view(amistats, route_name='amistats', renderer=_here+'/amistats.pt')

    [...]

```

Fixeu-vos que hi ha dues rutes, una per a la pàgina `/inici.html` i l'altra per a `/amistats.html`, i una funció per a cada una que retorna les variables particulars de cada pàgina. A més, a cada funció hi ha la variable *main* que és la renderització de la macro `master.pt`.