PHP, XHTML, CSS2

Área de Ingeniería Telemática

Hoy

- Más sobre PHP: algunos ejemplos
- Más sobre HTML: XHTML y CSS2

Hoy

- **Más sobre PHP: algunos ejemplos**
- Más sobre HTML: XHTML y CSS2

Ejemplo: formulario (info.php)

```
<?php
session_start();
include("inicio.php");
?>
<html>
<head>
  <title>T&iacute;tulo</title>
</head>
<body>
<?php $_SESSION['MiDato'] = "Esto quiero guardar"; ?>
<form enctype="multipart/form-data" action="info2.php?parametro=valor" method="POST">
  Nombre <input type="text" name="nombre" /><br /><br />
  <input type="hidden" name="MAX_FILE_SIZE" value="300000" />
  Enviar fichero: <input name="imagen" type="file" /><br />
  <input type="submit" value="Enviar fichero" />
</form>
</body>
<?php include("fin.php"); ?>
```

Ejemplo: formulario (info.php)

```
<?php
session_start();
include("inicio.php");
?>
<html>
<head>
  <title>T&iacute;tulo</title>
</head>
<body>
<?php $_SESSION['MiDato'] = "Esto quiero guardar"; ?>
<form enctype="multipart/form-data" action="info2.php?parametro=valor" method="POST">
  Nombre <input type="text" name="nombre" /><br /><br />
  <input type="hidden" name="MAX_FILE_SIZE" value="300000" />
  Enviar fichero: <input name="imagen" type="file" /><br />
  <input type="submit" value="Enviar fichero" />
</form>
</body>
<?php include("fin.php"); ?>
```

```
<?php
$server = "localhost";
$usuario = "lir_user";
$password = "lir_pass";
$link = mysql_connect($server, $usuario, $password);
mysql_select_db("lir_db", $link);
?>
```

Ejemplo: formulario (info.php)

```
<?php
session_start();
include("inicio.php");
?>
<html>
<head>
  <title>T&iacute;tulo</title>
</head>
<body>
<?php $_SESSION['MiDato'] = "Esto quiero guardar"; ?>
<form enctype="multipart/form-data" action="info2.php?parametro=valor" method="POST">
  Nombre <input type="text" name="nombre" /><br /><br />
  <input type="hidden" name="MAX_FILE_SIZE" value="300000" />
  Enviar fichero: <input name="imagen" type="file" /><br />
  <input type="submit" value="Enviar fichero" />
</form>
</body>
<?php include("fin.php"); ?>
```

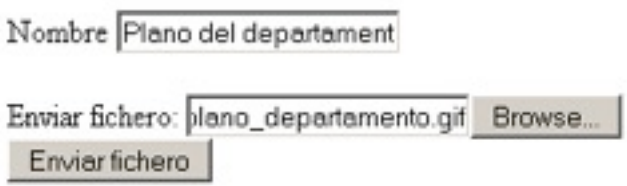
```
<?php
$server = "localhost";
$usuario = "lir_user";
$password = "lir_pass";
$link = mysql_connect($server, $usuario, $password);
mysql_select_db("lir_db", $link);
?>
```

```
<?php mysql_close($link); ?>
```

Ejemplo: formulario (info.php)

```
<?php
session_start();
include("inicio.php");
?>
<html>
<head>
  <title>T&iacute;tulo</title>
</head>
<body>
<?php $_SESSION['MiDato'] = "Esto quiero guardar"; ?>
<form enctype="multipart/form-data" action="info2.php?parametro=valor" method="POST">
  Nombre <input type="text" name="nombre" /><br /><br />
  <input type="hidden" name="MAX_FILE_SIZE" value="300000" />
  Enviar fichero: <input name="imagen" type="file" /><br />
  <input type="submit" value="Enviar fichero" />
</form>
</body>
<?php include("fin.php"); ?>
```

```
<?php
$server = "localhost";
$usuario = "lir_user";
$password = "lir_pass";
$link = mysql_connect($server, $usuario, $password);
mysql_select_db("lir_db", $link);
?>
```




```
<?php mysql_close($link); ?>
```

Ejemplo: leyendo el formulario (info2.php)

```
<?php
session_start();
include("inicio.php");
?>
<pre>
<?php
print "Get: "; print_r($_GET);
print "Post: "; print_r($_POST);
print "Request: "; print_r($_REQUEST);
print "Files: "; print_r($_FILES);
print "Session: "; print_r($_SESSION);
print "Server: "; print_r($_SERVER);
print "Env: "; print_r($_ENV);
?>
</pre>
<?php
include("fin.php");
?>
```

Los nombres que son clave en el array asociativo $_SESSION, deben tener un nombre válido de variable. Es decir, no pueden empezar por número

¿Que hay en $_GET, $_POST, $_REQUEST ?

```
Get: Array
([parametro] => valor )

Post: Array
(
    [nombre] => Plano del departamento
    [MAX_FILE_SIZE] => 300000
)
Request: Array
(
    [parametro] => valor
    [nombre] => Plano del departamento
    [MAX_FILE_SIZE] => 300000
    [PHPSESSID] => 26465da5723cccfacc1869072db5a8f7
)
```

$FILES, $_SESSION

```
Files: Array
(
[imagen] => Array
    (
    [name] => plano_departamento.gif
    [type] => image/gif
    [tmp_name] => /tmp/phpx1UEna
    [error] => 0
    [size] => 192884
    )
)

Session: Array
(
[MiDato] => Esto quiero guardar
)
```

$_SERVER

```
Server: Array
(
        [HTTP_HOST] => localhost
        [HTTP_USER_AGENT] => Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7.12) Gecko/20050915 Firefox/1.0.7
        [HTTP_ACCEPT] => text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
        [HTTP_ACCEPT_LANGUAGE] => en-us,en;q=0.5
        [HTTP_ACCEPT_ENCODING] => gzip,deflate
        [HTTP_ACCEPT_CHARSET] => ISO-8859-1,utf-8;q=0.7,*;q=0.7
        [HTTP_KEEP_ALIVE] => 300
        [HTTP_CONNECTION] => keep-alive
        [HTTP_REFERER] => http://locahost/info.php
        [HTTP_COOKIE] => PHPSESSID=26465da5723cccfacc1869072db5a8f7
        [CONTENT_TYPE] => multipart/form-data; boundary=---------------------------8767276471703
        [CONTENT_LENGTH] => 193310
        [PATH] => /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/bin/X11
        [SERVER_SIGNATURE] => Apache/2.0.54 (Unix) DAV/2 PHP/4.4.0 Server at localhost Port 80
        [SERVER_SOFTWARE] => Apache/2.0.54 (Unix) DAV/2 PHP/4.4.0
        [SERVER_NAME] => localhost
        [SERVER_ADDR] => 192.168.2.40
        [SERVER_PORT] => 80
        [REMOTE_ADDR] => 192.168.2.1
        [DOCUMENT_ROOT] => /www
        [SERVER_ADMIN] => fran@navarparty.org
        [SCRIPT_FILENAME] => /www/info2.php
        [REMOTE_PORT] => 4079
        [GATEWAY_INTERFACE] => CGI/1.1
        [SERVER_PROTOCOL] => HTTP/1.1
        [REQUEST_METHOD] => POST
        [QUERY_STRING] => parametro=valor
        [REQUEST_URI] => /info2.php?parametro=valor
        [SCRIPT_NAME] => /info2.php
        [PHP_SELF] => /info2.php
        [PATH_TRANSLATED] => /www/info2.php
        [argv] => Array
```

$_ENV

```
Env: Array
(
    [HZ] => 100
    [TERM] => linux
    [SHELL] => /bin/bash
    [HUSHLOGIN] => FALSE
    [USER] => root
    [LD_LIBRARY_PATH] => /programas/apache2/lib:
    [PATH] => /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/
      bin:/usr/bin/X11
    [MAIL] => /var/mail/root
    [PWD] => /programas/apache2
    [LANG] => es_ES@euro
    [HOME] => /root
    [SHLVL] => 2
    [LOGNAME] => root
    [_] => /programas/apache2/bin/httpd
)
```

Ejemplo: Cómo NO hay que completar valores

```
<?php session_start(); include("inicio.php"); ?>
<html>
<html>
<head>
  <title>Ejemplo con selects</title>
</head>
<body>
<form method="post" action="info.php">
  <select name="day">
    <option value="1">1</option>
    <option value="2">2</option>
    <option value="3">3</option>
         ... etc ...
    <option value="31">31</option>
  </select>
  <select name="month">
    <option value="1">Enero</option>
    <option value="2">Febrero</option>
    <option value="3">Marzo</option>
         ... etc ...
    <option value="12">Diciembre</option>
  </select>
<select name="year">
   <option value="2005"> 2005 </option>
   <option value="2006"> 2006 </option>
   <option value="2007"> 2007 </option>
         ... etc ...
   <option value="2014"> 2014 </option>
  </select>
</form>
</body>
</html>
<?php include("fin.php"); ?>
```

Cómo SÍ hay que completar valores

```
<?php session_start(); include("inicio.php"); ?>
<html>
<html>
<head>
  <title>Ejemplo con selects</title>
</head>
<body>
<form method="post" action="info.php">
 <select name="day">
<?php
for($i=1;$i<=31;$i++)
  print "<option value=\"$i\">$i</option>\n";
?>
  </select>
  <select name="month">
<?php
$meses = array("Enero", "Febrero", "Marzo",...,"Diciembre");
foreach($meses AS $clave => $actual){
?>
    <option value="<?php echo ($clave+1);?>"><?php echo $actual;?></option>
<?php
?>
  </select>
  <select name="year">
<?php
$init_year = strftime("%Y");
$end_year = $init_year+10;
for($i=$init_year;$i<$end_year;$i++)
  print "<option value=\"$i\">$i</option>\n";
?>
  </select>
</form>
</body>
</html>
<?php include("fin.php"); ?>
```

Mandar mails

- PHP puede utilizar el Mail Transport Agent (MTA) del sistema para enviar e-mails
- Función **mail()**

```
bool mail ( string to, string subject, string message [,
string additional_headers [, string
additional_parameters]] )
```

- Ejemplo:

```
mail("alumno@unavarra.es", "Datos del programa", "Los
datos son:\n\n1\n2\n3\n\n y ya está",
"From:felix.espina@unavarra.es\r
\n"."CC:otro@unavarra.es");
```

Variables

- Las variables son locales a su "scope"
- Si queremos que "traspasen" estos límites, usar global
- Este comportamiento no se cumple con las "superglobals" puesto que existen en todo el script: $_GET, $_POST, $_SERVER, $_SESSION, etc
- Ejemplo:

```
$a = "DatoA";
$b = "DatoB";
function hacer_algo($a)
  {
  global $b;
  global $_GET['c'];

  return $a.$b.$_GET['c'];
  }
```

Variables

- Las variables son locales a su "scope"
- Si queremos que "traspasen" estos límites, usar global
- Este comportamiento no se cumple con las "superglobals" puesto que existen en todo el script: $_GET, $_POST, $_SERVER, $_SESSION, etc
- Ejemplo:

```
$a = "DatoA";
$b = "DatoB";
function hacer_algo($a)
  {
  global $b;
  global $_GET['c'];

  return $a.$b.$_GET['c'];
  }
```

Includes

- Cuando se realiza un include() de un fichero:
- El código en él se ejecuta. Por ejemplo: primer ejemplo de hoy
- Las funciones en él, no se ejecutan, sólo “se definen”. Permiten su uso, pero no son ejecutadas hasta que se llaman
- Con las conexiones y demás elementos de patrón “Singleton”, con require_once() o include_once

Acceso al sistema y ficheros

- Podemos ejecutar comandos del sistema

```
system("ls -al /etc/init.d")
```

- Trabajar con ficheros y directorios

```
$dir = "/etc/php5/";

// Abrir directorio conocido y leer sus contenidos
if (is_dir($dir)) {
   if ($dh = opendir($dir)) {
       while (($file = readdir($dh)) !== false) {
              echo "filename:$file:filetype:".filetype($dir.$file)."\n";
       }
       closedir($dh);
   }
}
```

Clases y objetos

- Siempre que sea posible, usar clases y objetos
- Pese a que PHP4 realiza una implementación "particular", funciona y da buenos resultados
- La orientación a objetos siempre permite escalabilidad y control sobre el programa
- Si no se usan objetos, por lo menos dividir el programa en partes y trabajar con funciones e includes

Clases y objetos PHP 4

```
<?php
class Carrito {
    var $items;  // Items en nuestro carrito de compras

    function agregar_item($artnr, $num) {
        $this->items[$artnr] += $num;
    }

    function retirar_item($artnr, $num) {
        if ($this->items[$artnr] > $num) {
            $this->items[$artnr] -= $num;
            return true;
        } elseif ($this->items[$artnr] == $num) {
            unset($this->items[$artnr]);
            return true;
        } else {
            return false;
        }
    }
}
?>
```

```
<?php
class Carrito_Con_Nombre extends Carrito {
    var $duenyo;

    function definir_duenyo ($nombre) {
        $this->duenyo = $nombre;
    }
}
?>
```

Clases y objetos PHP 4

```
<?php
   $carrito = new Carrito;
   $carrito->agregar_item("10", 1);

   $otro_carrito = new Carrito;
   $otro_carrito->agregar_item("0815", 3);
?>
```

```
<?php
   $carrito_n = new Carrito_Con_Nombre;  // Crear un carrito con nombre
   $carrito_n->definir_duenyo("kris");   // Nombrar el carrito
   print $carrito_n->duenyo;             // imprimir el nombre del dueño
   $carrito_n->agregar_item("10", 1);    // (funcionalidad heredada de
                                            carrito)
?>
```

Clases y objetos PHP 4

- ***NO*** es posible separar la definición de una clase en varios archivos.
- ***NO*** es posible separar la definición de una clase en bloques PHP diferentes
- Las clases pueden ser extensiones de otras clases
  - La clase extendida o derivada tiene todas las variables y funciones de la clase base
  - **No** es posible substraer de una clase, es decir, remover la definición de cualquier función o variable existente
  - Una clase extendida siempre depende de una clase base única, lo que quiere decir que no se soporta herencia múltiple
  - Las clases son extendidas usando la palabra clave 'extends'
- ¡Las clases deben ser definidas antes de ser usadas! => el orden en que se definen las clases es importante.

Clases y objetos PHP 4

- En PHP 4, sólo se permiten inicializadores constantes para variables *var*
- Para inicializar variables con valores no-constantes => función de inicialización (constructora) que sea llamada automáticamente cuando un objeto es construido a partir de la clase

```
<?php
    class Carrito {
        /* Ninguna de estas expresiones funciona en PHP 4. */
        var $fecha_hoy = date("Y-m-d");
        var $nombre = $primer_nombre;
        var $duenyo = 'Fred ' . 'Jones';
        /* Aunque, las matrices que contienen valores constantes funcionan */
        var $items = array("VCR", "TV");
    }

    /* Asi es como debe declararse. */
    class Carrito {
        var $fecha_hoy;
        var $nombre;
        var $duenyo;
        var $items = array("VCR", "TV");

        function Carrito() {
            $this->fecha_hoy = date("Y-m-d");
            $this->nombre = $GLOBALS['primer_nombre'];
            /* etc. . . */
        }
    }
?>
```

Hoy

- Más sobre PHP: algunos ejemplos
- **Más sobre HTML: XHTML y CSS2**

Contenidos

- XHTML 1.0
  - Modo de renderizado
  - Box Model
  - Tipos de elementos
  - Posicionamiento de elementos
- CSS 2

XHTML 1.0

```
<html>
<head>
  <title>Titulo del documento</title>
</head>
<body>
  <p>Este es un HTML 4.01 correcto<br>
  pero un XHTML 1.0 incorrecto
</body>
</html>
```

```
<html>
<head>
  <title>Titulo del documento</title>
</head>
<body>
  <p>Este es un HTML 4.01 correcto<br />
  y un XHTML 1.0 correcto</p>
</body>
</html>
```

- Diferencias principales con HTML 4.01
  - Elementos XHTML apropiadamente anidados
  - Documentos bien escritos
  - Nombres y atributos de etiquetas en minúsculas
  - Todos los elementos cerrados, incluidos elementos vacíos
  - Los elementos se identifican sólo por 'id' únicos
  - Únicos que pueden usar 'name' elementos de formulario
  - Todos los atributos tienen valor
  - Preferiblemente script y style externos ➔ evitar CDATA y errores con < &]]>

XHTML 1.0

- XHTML 1.0 tiene los mismos elementos y atributos que HTML 4.01
- Si documento XHTML 1.0 puro (sin otro lenguaje de marcado) no hay diferencia entre este y uno HTML 4.01 siguiendo rigidez XML
- Se debería de enviar como application/xhtml+xml ó application/xml, pero se puede enviar como text/html
  - IE no lo entiende, resto si ➔ IE necesita text/html
  - Soluciones:
    - W3C: aprovechar bug de IE: http://www.w3.org/MarkUp/2004/xhtml-faq
    - Gurus: programación en servidor: HTTP_ACCEPT, header
    - Web: enviar todo como text/html

XHTML y CSS

- CSS 1 principalmente para dar formatos de fuentes, tamaños, colores, ...
- CSS 2 para posicionamiento de elementos (layout) + lo anterior
  - Mayor separación entre contenido y forma
  - Facilidad de rediseño (http://www.csszengarden.com/)
  - Inicialmente mayor complejidad al desarrollar => ya no se hace con tablas sino con div (capas)
  - Aumenta la usabilidad y accesibilidad
  - Combinación de plataforma/navegador modifica apariencia:
    - No soportan todos el mismo nivel
    - Hay cosas que las representan mal (especialmente IE)

Contenidos

- XHTML 1.0
  - Modo de renderizado
  - Box Model
  - Tipos de elementos
  - Posicionamiento de elementos
- CSS 2

Modo de renderizado

- Navegadores tiene 2 formas de comportarse:
  - Quirks Mode: como navegadores antiguos (<v4)
  - Strict Mode: intentando seguir estándar (modernos)

  http://www.quirksmode.org/css/quirksmode.html

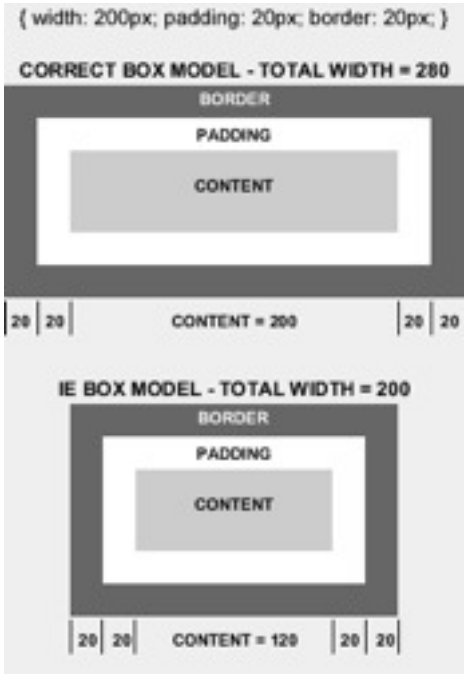
- Estándar XHTML recomienda poner prolog xml antes que DOCTYPE ➔ IE 6 salta a Quirks ➔ soluciones:
  - No poner prolog ➔ Todos los navegadores modernos en Strict
  - Poner prolog ➔ IE 6 no renderiza como un navegador moderno

Contenidos

- XHTML 1.0
  - Modo de renderizado
  - Box Model
  - Tipos de elementos
  - Posicionamiento de elementos
- CSS 2

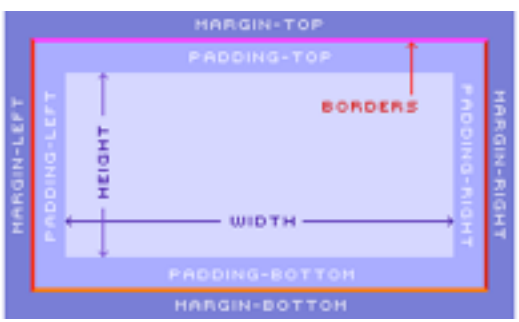
Box Model

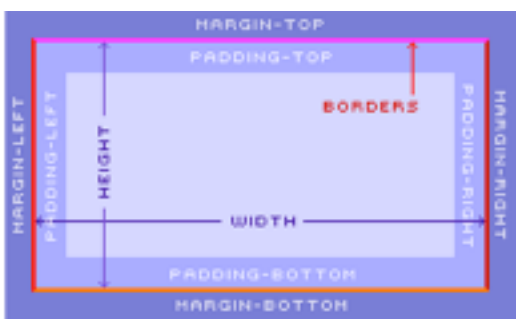
- 2 Box Model:
  - Estándar (W3C): lo aplican navegadores en Strict Mode => navegadores modernos (IE 6 sin prolog)
  - Tradicional: dentro del ancho del elemento incluye el padding y el border => navegadores viejos (IE 6 con prolog)



http://css.maxdesign.com.au/listamatic/about-boxmodel.htm

http://css-discuss.incutio.com/?page=BoxModelHack





Contenidos

- XHTML 1.0
  - Modo de renderizado
  - Box Model
  - Tipos de elementos
  - Posicionamiento de elementos
- CSS 2

Tipos de elementos

- **Block**: están separados del los elementos adyacentes: p, div, ...
- **Inline**: en vez de estar separados de los elementos adyacentes aparecen a continuación de ellos: a, strong, span, cite, ...
- **List-item**: parecidos a los elementos de bloque, pero a la izquierda tienen marcas: li, ul, ol

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Tipos de elementos</title>
</head>
<body>
 <p>Esto aparece separado de lo siguiente.</p>
 <p>Elemento tipo block, pero <strong>esto es un
elemento tipo inline que aparece</strong> seguido.</p>
 <ul>
 <li>Lista</li>
 </ul>
 <p>Texto normal</p>
</body>
</html>
```

Esto aparece separado de lo siguiente.

Elemento tipo block, pero **esto es un elemento tipo inline que aparece** seguido.

- Lista

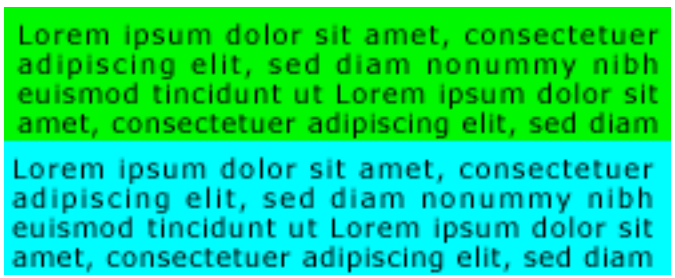
Texto normal

Contenidos

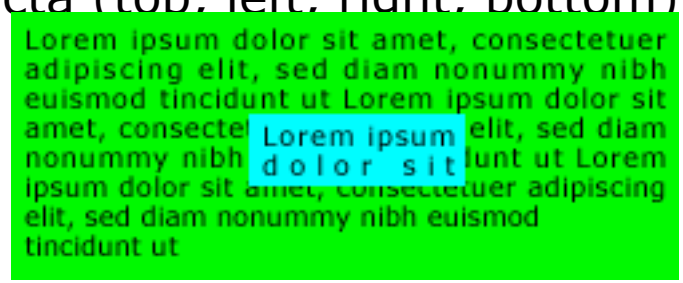
- XHTML 1.0
  - Modo de renderizado
  - Box Model
  - Tipos de elementos
  - Posicionamiento de elementos
- CSS 2

Posicionamiento de elementos

- CSS2 posiciona los elementos de 4 maneras diferentes
- **Estatico**: modo por defecto, mostrar los elementos en el orden natural de lectura.



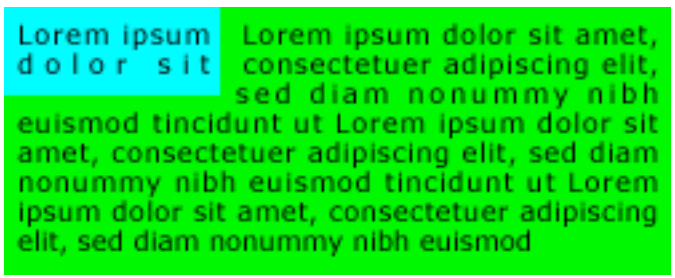
- **Absoluto**: establecer posición exacta (top, left, right, bottom) de un elemento respecto a su padre.



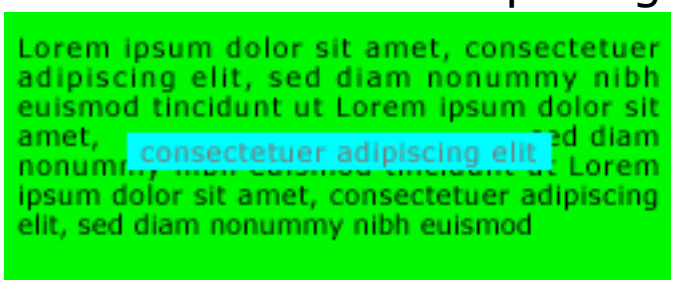
- **Fixed**: subtipo del absoluto. El scroll afecta a un elemento posicionado absolutamente, con fixed no. Fixed tiene posición absoluta no sobre la ventana. Se puede usar para simular frames.

Posicionamiento de elementos

- **Float**: se pone el elemento donde estaría posicionado estáticamente y se saca del flujo normal a la derecha o izquierda. El siguiente elemento flota alrededor suyo.



- **Relativo**: Coge el elemento donde estaría posicionado estáticamente y lo desplaza Zpx hacia abajo y Ypx hacia la derecha. El resto de elementos considera que sigue estando en su posición normal.



Posicionamiento de elementos

- **Problema**:
  - Para hacer layout's interesantes hay que mezclar diferentes posicionamientos
  - Al mezclar posicionamientos resultado muy enrevesado

- En **teoría** se puede poner cualquier posicionamiento a cualquier tipo de elemento
- En **práctica** el posicionamiento general (layout) se hace prácticamente solo con `div`:
  - Consecuencia de bugs, malas implementaciones, ... de los navegadores

Contenidos

- XHTML 1.0
  - Modo de renderizado
  - Box Model
  - Tipos de elementos
  - Posicionamiento de elementos
- CSS 2

CSS 2

- Añade todos los estilos necesarios para posicionamiento y maquetación de layouts
- Prácticamente soportado **completamente** por todos los navegadores modernos => los viejos pasan de estos estilos
- Problema: la implementación de todos los estilos en todos los navegadores no es coherente => bugs de renderizado para ciertos navegadores en ciertas circunstancias
- Por eso se considera más difícil maquetar con CSS, que con tablas
- Navegador más "estándar": Firefox
- Herramienta útil de desarrollo: Web Developer Toolbar

CSS 2

- position: static, relative, absolute, fixed
- float: none, left, right, both
- display: none, block, inline, inline-table, list-item, ...
- ...

Conclusiones

- PHP
- ejemplos uso de POST y GET
- XHTML
- CSS2 y posicionamiento