8. CONCLUSIONES

Este proyecto ha sido un reto para mí, ya que antes de comenzar no tenía conocimientos de casi ninguna de las tecnologías utilizadas en este proyecto, como son PHP, HTML, XHTML, CSS y javascript.

El hecho de trabajar en un proyecto de Open Source, donde el trabajo realizado tenía un impacto real en una comunidad tan importante como la comunidad Moodle, me atrajo enormemente e impulsó mis ganas de aprender y colaborar con todo el equipo liderado por Marc Alier.

Lo que más valoro de este proyecto es sin lugar a dudas el nivel tan grande de formación que he adquirido. No solo por las nuevas herramientas y tecnologías que he aprendido sino también por el hecho de tener que tocar código que mucha gente había tocado ya. Esto es algo que no había hecho durante la carrera, siempre se ha empezado desde cero y es muy importante haber tenido esta experiencia ya que creo que esta situación va a ser habitual en el mundo laboral.

Por otro lado pienso que la metodología de trabajo usada por el director del proyecto, es la idónea para realizar proyectos de esta envergadura. El hecho de trabajar en parejas hace que el trabajo realizado sea de mayor calidad ya que continuamente compartíamos el código y refinábamos los procesos que había hecho el otro miembro del grupo. Seguir un sistema de trabajo con ciclos muy cortos, la mayoría de las veces de una semana, facilitaba a tener muy claro cual era el objetivo a cumplir y el trabajo que se debía realizar.

Las reuniones semanales han permitido definir el tiempo dedicado a las tareas, y han ayudado a que, teniendo cada semana unos objetivos parciales muy definidos, los objetivos globales se pudiesen cumplir.

El hecho de estar en contacto permanentemente con el director, ha sido de gran ayuda a la hora de plantear nuevas estrategias y de priorizar las tareas que se debía realizar dentro del proyecto global.

Estar en un proyecto grande, con tantas personas trabajando, me ha permitido también aprender del trabajo de otras personas, y debido al excelente clima de trabajo de todo el equipo ha sido una experiencia enriquecedora también a nivel humano.

Ha sido un proyecto difícil de realizar, al que he tenido que dedicarle muchas horas, pero me queda una gran experiencia, y cuando miro hacia atrás, me asombro de la cantidad de nuevos conocimientos que he adquirido y asimilado, y me satisface el haberlos aplicado a un proyecto que está vivo y que seguirá evolucionando en el futuro.

Ahora comienza una nueva etapa para mí, tanto en lo laboral como en lo personal. Sin embargo, espero poder disponer de algún tiempo para poder continuar colaborando, aunque sea ocasionalmente, con el Dfwikiteam y con la comunidad Moodle en general.

9. BIBLIOGRAFÍA

Crom
http://morfeo.upc.es/crom/index.php?lang=es

Moodle
http://moodle.org/index.php?lang=es_utf8

SourceForge
http://sourceforge.net/

Accesibilidad Web
http://www.w3c.es/divulgacion/guiasbreves/Accesibilidad%20
http://www.sidar.org/index.php
http://www.nosolousabilidad.com/articulos/accesibilidad.htm
http://griho.udl.es/ipo/pdf/07Accesi.pdf

Herramientas para la accesibilidad Web
http://www.tawdis.net/taw3/cms/es
http://www.sidar.org/hera/
http://www.accesible.com.ar/examinator/

XHTML
http://www.w3c.es/Divulgacion/Guiasbreves/XHTML
http://www.w3.org/TR/xhtml1/
http://www.w3schools.com/tags/tag_script.asp

Herramientas para validar el código en XHTML
http://validator.w3.org/
https://addons.mozilla.org/es-ES/firefox/addon/60
http://www.getfirebug.com/
http://www.microsoft.com/downloads/details.aspx?FamilyID=e59c3964-672d-4511-bb3e-2d5e1db91038&displaylang=en

Blogs del DFWikiteam

Blog mío y de Jose - http://katiajose.blogspot.com/

Blog Albert Gasset - http://albert719.wordpress.com/tag/pfc/

Blog Jose Antonio Rodríguez - http://33pfc2006.blogspot.com/

Blog Laura Montagut - http://wikiwii.blogspot.com/

Blog Marc Fité i Santandreu - http://cartrons.blogspot.com/

Blog Modesto y Ester - http://modestocaballero.com/blogpfc/

Blog Pigui - http://elpigui-i-lawiki.blogspot.com/

Blog Uri - http://enochrooted.blogspot.com/

10. ANEXO

10.1. MANUAL XHTML

Lo primero que debemos hacer para desarrollar un documento XHTML válido, es indicarle al navegador el tipo de documento que va a recibir. Actualmente existen tres tipos de documentos XHTML que podemos usar:

- **Transicional**: permite la utilización de algunas etiquetas de formato dentro del código de las páginas para ser soportadas por navegadores antiguos que tienen problemas con las CSS
- **Estricto**: se utiliza en combinación con W3C CSS, siguen al pie de la letra las convenciones marcadas separando completamente las etiquetas de formato del contenido (el formato se apoya exclusivamente en la CSS).
- **Frameset**: se utiliza cuando se divide la pantalla en varios marcos o frames.

Nuestro primer objetivo será conseguir que la nwiki sea válida para el estándar XHTML Transicional. Una vez conseguido, trabajaremos para que cumpla el segundo tipo explicado, Estricto, más restrictivo que el anterior.

La declaración del tipo utilizado se debe hacer al inicio de cualquier documento que forme parte de la web. A continuación vemos la correcta aplicación de dicha declaración, que conocemos como DOCTYPE:

```
<?xml version="1.0" encoding="iso-8859-1"?>

<!DOCTYPE html PUBLIC "-//WC3//DTD XHTML 1.0 Transitional//EN"
http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Se tiene que destacar que siempre que el documento vaya a usar caracteres latinos (acentos y eñes) es necesario indicar una codificación compatible con los mismos. La codificación por default de XHTML es UTF-8, misma que tiene conflictos con algunos caracteres de uso

común en nuestra lengua, por eso es importante asegurarse que la codificación sea iso-8859-1, como se ve en el ejemplo de arriba.

A continuación se describe 5 reglas básicas que cubren la mayor parte de los errores de un documento XHTML.

1. Si no lo cierras, no sirve

En XHTML no puede haber elementos sin cierre. Un debe cerrarse con un , un <li> debe cerrarse con un , y así todos los elementos.
Aquellos que son independientes como
, <hr> y los <input>, deben cerrarse a si mismos de esta forma:
, <hr /> y <input />.

2. Anida de forma lógica

Los elementos en un documento XHTML deben cerrarse en el orden inverso que fueron abiertos.
De tal forma que debe cerrarse en el orden . No puedes saltarte un cierre, eso convertiría tu documento en "Mal Formado", y fallaría su validación.

3. No uses mayúsculas

Tanto los nombres de las etiquetas como los nombres de los atributos deben escribirse en minúsculas.
Esto quiere decir que la siguiente etiqueta: <DIV CLASS="EncabezadoPrincipal"> es incorrecta, su forma correcta sería: , nota que el valor del atributo si puede tener mayúsculas, la regla se aplica para los nombres de las etiquetas y atributos, no para los valores o contenido de los mismos.

4. Usa las comillas

Todos los atributos de cualquier etiqueta deben encerrarse entre comillas.

La forma correcta de asignar valores a los atributos de una etiqueta es la siguiente: <img src="mi_imagen.gif" height="50" width="150" alt="Mi Foto">.

5. No lo uses para diseñar

Piensa en XHTML como el medio para organizar y estructurar tu documento, no como un medio para darle formato. Olvídate de los <font>, los <... background>, <...align>, etcétera.

10.2. MANUAL WEBLIB

10.2.1. Introducción

Todas las funciones contienen únicamente aquellos atributos de los tags que se utilizan para el desarrollo de la nwiki y siguen el XHTML Strict. De este modo conseguimos que las etiquetas creadas con la utilización de este fichero cumplan el estándar.

La mayoría de las propiedades de estilo que no cumplan el estándar se pueden definir en el class o style. Estos atributos están contenidos en todas las etiquetas. Su función es la de asignar al tag un formato o estilo, cumpliendo las normas de XHTML Strict. Este formato puede estar definido en una hoja de estilo (css), y por lo tanto utilizaríamos el atributo class para asignarle la clase ya definida, o mediante el style, en caso de que no este declarado.

Ejemplo:

$propiedades->style = 'width:1%;white-space:nowrap';
wiki_change_column($propiedades);

sería

</td> <td style="width:1%;white-space:nowrap">

10.2.2. Creación de tags

Para la creación de tags, todas las funciones siguen la misma estructura. En estas se abre y se cierra la etiqueta HTML, añadiendo las propiedades necesarias (exceptuando algunas que se detallaran más adelante)

Todas las funciones siguen también un mismo patrón para el orden de sus parámetros:

- **Parámetro $info**

Si la etiqueta tiene información entre la apertura y cierre de ésta, la función tendrá un primer parámetro ($info), que contendrá dicha información. A excepción de las tablas y listas, que se explicarán más adelante.

- **Parámetro $propert**

El parámetro $propert es un objeto que contiene las diferentes propiedades asociadas a las etiquetas creadas.

Ejemplo de asignación de propiedades a una etiqueta:
$propert->src="image.gif";
$propert->alt="imagen";
wiki_img($propert);
Sería:
<img src="image.gif" alt="imagen">

Los eventos se ubicarán también dentro del objeto. Éstos se escribirán como un string y contendrán los diferentes tipos de eventos y su función. Por ejemplo: **$propert->events = Xonfocus="this.select()" Xonchange="updateMembers(this)"**

- **Parámetro $return**

Todas las funciones tienen como último parámetro la variable $return, inicializada por defecto a false.
En caso de ser false, la función escribiría el tag directamente. Y si lo cambiásemos a true, la función retornaría una variable con la etiqueta.

10.2.3. Excepciones

La creación de algunos tags no siguen las normas explicadas anteriormente.

Estas excepciones son:

- **Creación de form y div**

Debido a la longitud del parámetro $info en algunos ejemplos de FORM y DIV, hemos visto conveniente realizar una segunda función, que a diferencia de la primera, no realiza el cierre de la etiqueta, y por lo tanto, no recibe como parámetro $info. Añadiendo además sus respectivas funciones de cierre, independientes de las anteriores.

Con esto conseguimos más claridad en el código y un manejo más sencillo de las funciones.

- **Creación de tablas**

Para la creación de tablas tenemos varias funciones:

wiki_table_start($propert, $return=false)
Inicia tabla, fila y columna, con las respectivas propiedades. El atributo $propert contiene los atributos de la tabla y los de la primera columna creada. Los atributos de la columna estarán identificados por un "td" final, por ejemplo $propert->aligntd.

En el caso de que queramos que la tabla contenga una cabecera, $propert->header se tiene que poner a true. Los atributos de la cabecera también están identificados por un "th" al final, por ejemplo $propert->alignth.

wiki_table_end($return=false)
Cierra una columna, fila y tabla.
Si tenemos una tabla que solo contiene una cabecera, tendremos que poner $propert->header a true.

wiki_change_column($propert, $return=false)
Cierra columna e inicia una nueva con las propiedades correspondientes.
Si estamos definiendo la cabecera, tendremos que poner $propert->header a true.

wiki_change_row($propert, $return=false)

Cierra columna y fila e inicia una nueva fila y una columna con las propiedades correspondientes.
En el caso de finalizar la cabecera, se tiene que poner $propert->header a true para que se cierre la cabecera.

Ejemplo:

```
$propert->border = '0';
$propert->width = '100%';
$propert->aligntd = 'right';
wiki_table_start($propert);
echo '1';
wiki_change_row();
echo '2';
wiki_table_end();
```

sería

```
<table width="100%" border="0">
<tr><td align="right">
1
</td></tr><tr><td>
2
</td></tr>
</table>
```

- **Creación de listas**

Para crear las listas tenemos varias funciones que siguen la misma mecánica que las utilizadas para la creación de tablas.

wiki_start_ul($propert, $return=false)
Inicia una lista no ordenada y el primer ítem

wiki_end_ul($return=false)

Cierra el último ítem y la lista no ordenada

wiki_change_li($propert, $return=false)

Cambio de ítem

Para las listas ordenadas las funciones son iguales.

10.3. CSS

10.3.1. Manual de instalación nwiki.css

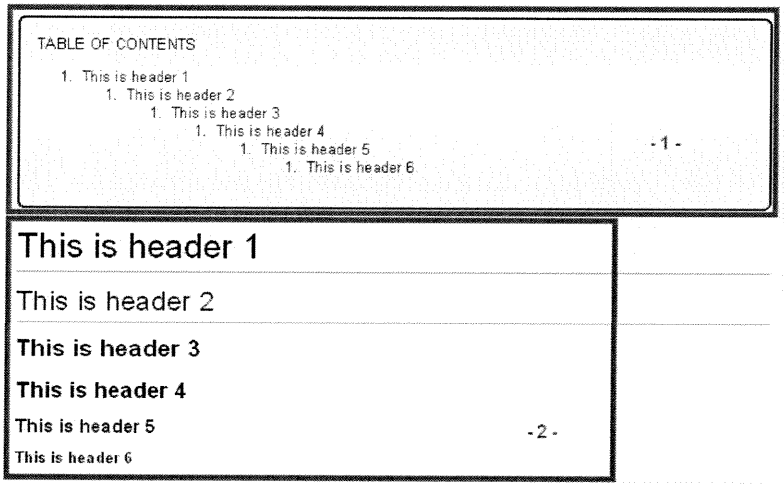
Para que moodle cargue la hoja de estilos de la nwiki, es necesario cambiar el archivo styles.php situado dentro de la carpeta de instalación de moodle en moodle\theme\standard\ por el descargado en el paquete de instalación de la nwiki.

Este nuevo archivo styles.php realiza las mismas funciones que el de moodle, pero además añade una función para cargar la hoja de estilos de la nwiki.

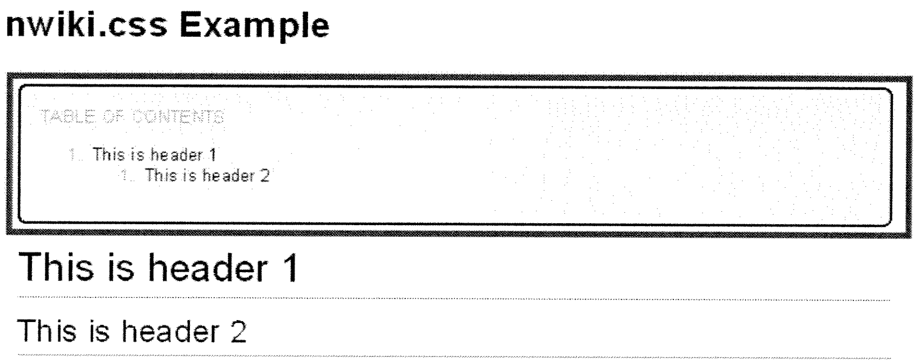
Puesto que el fichero modificado se almacena en una carpeta de la wiki siguiendo la misma estructura que moodle, al descomprimir la wiki se nos reemplazará el archivo style.php.

10.3.2. Manual de clases nwiki.css

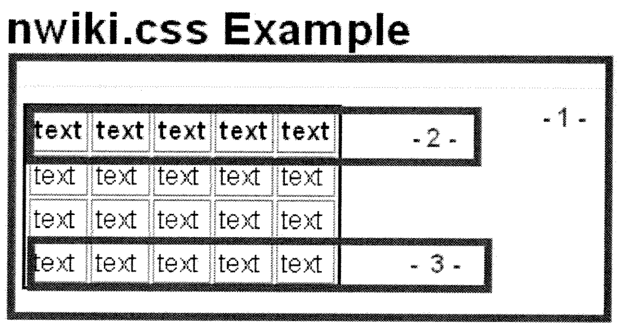
nwiki.css Example



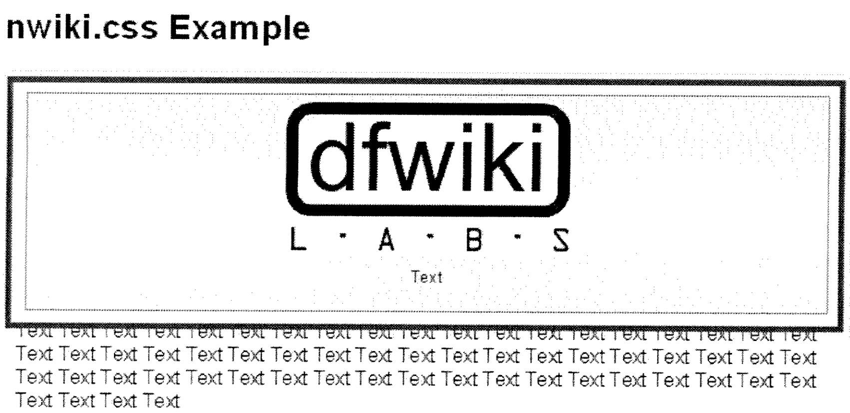
Clase toc(1) para div y nwiki(2) para headers



Clase nwikisintaxtoc for div



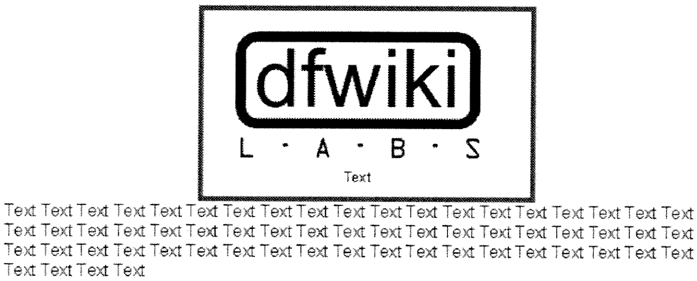
Clase nwikitable para table(1) th(2) y td(3)



Clase img para div

---

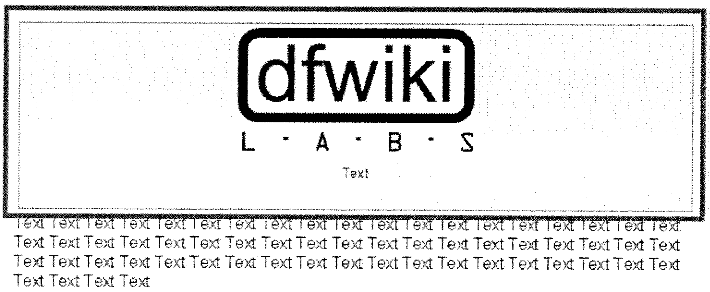
nwiki.css Example



Clase imgcaption para div

---

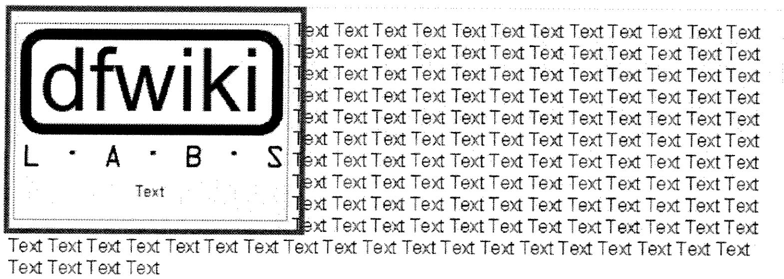
nwiki.css Example



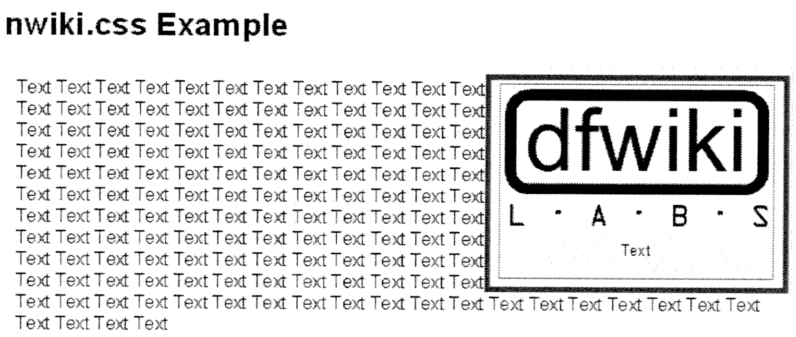
Clase imgcenter para div

---

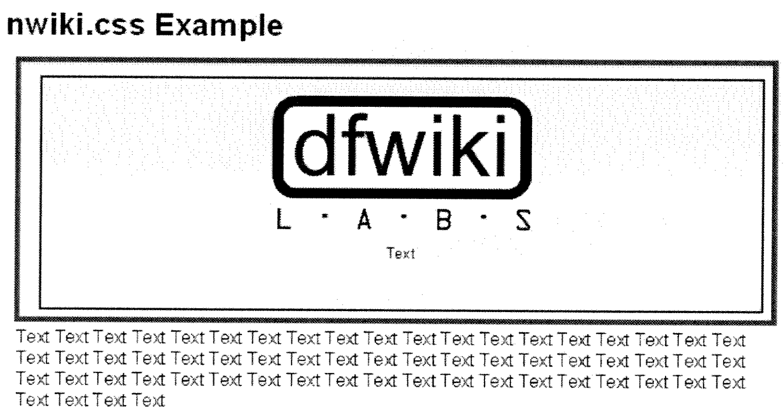
nwiki.css Example



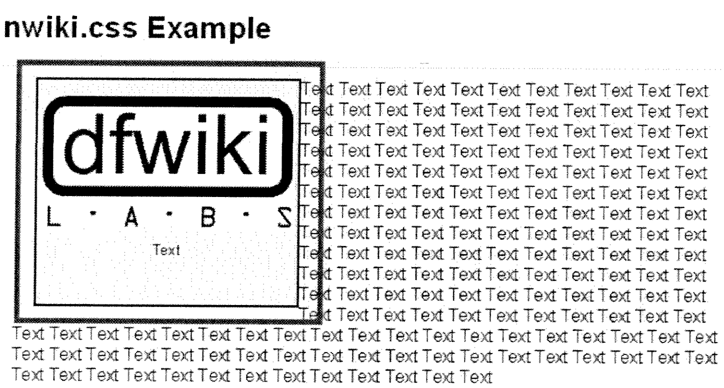
Clase imgleft para div



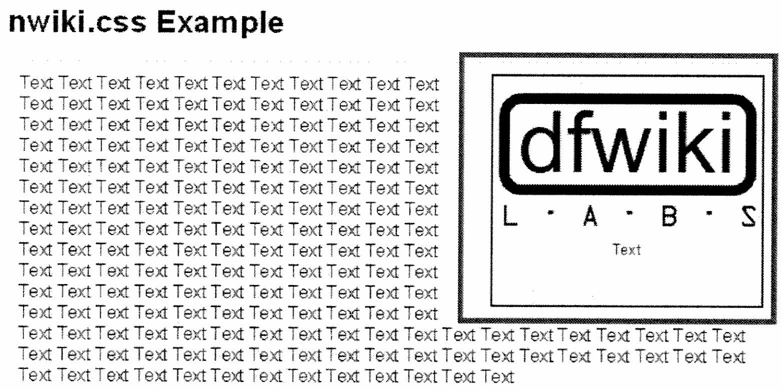
Clase imgright para div



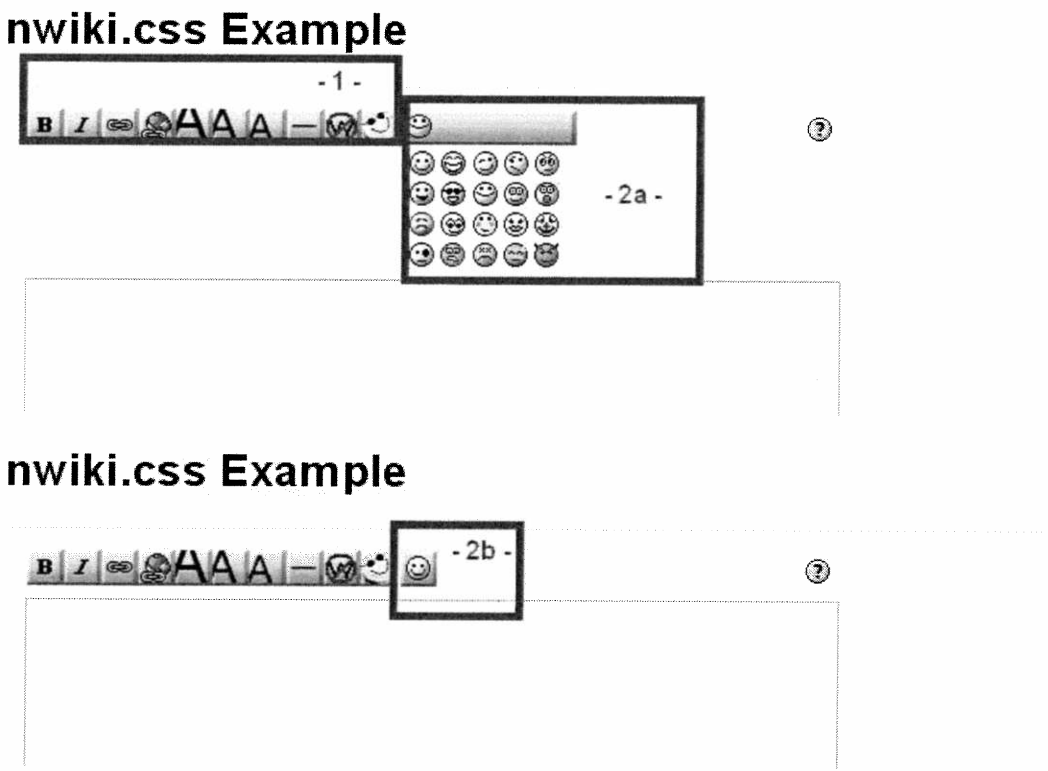
Clase imgthumbnailcenter para div



Clase imgthumbnailleft para div

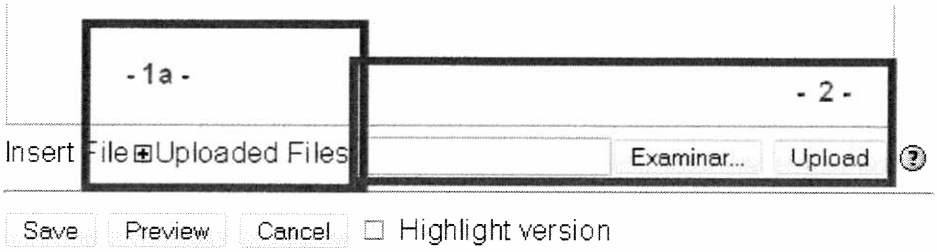


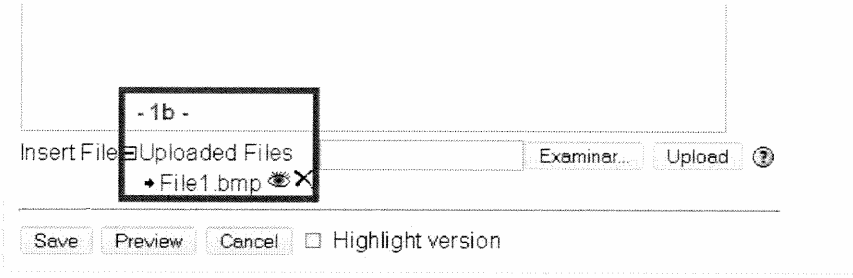
Clase imgthumbnailright para div



Clase nwikitambuttons(1) para td y class wiki_listme+icsme(2) for ul

---

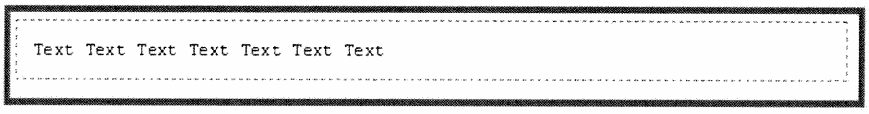




Clase wiki_listme+wiki_folding(1) para ul y nwikiuptam(2) para td

---

nwiki.css Example

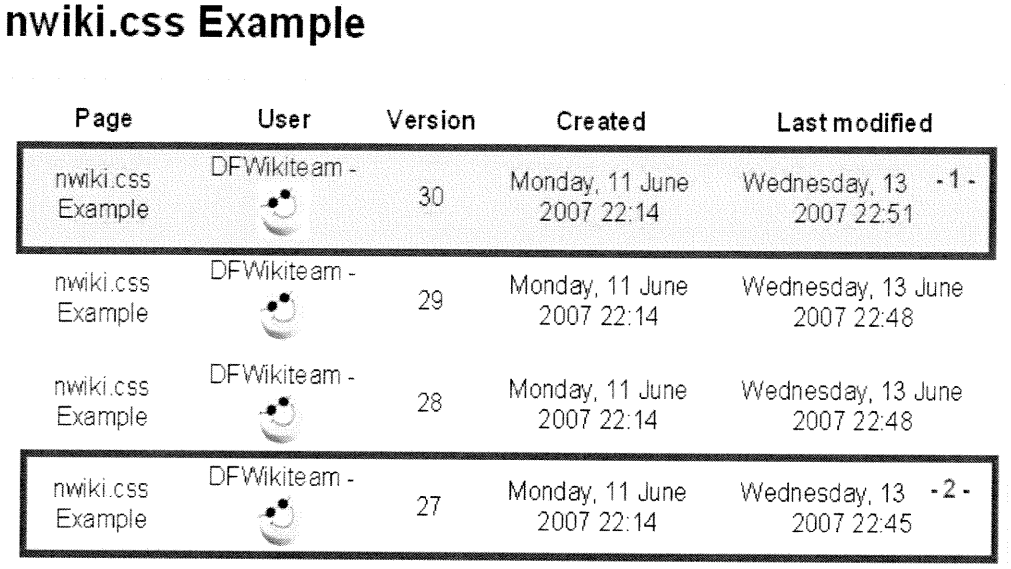


Clase quote para pre

---

nwiki.css Example

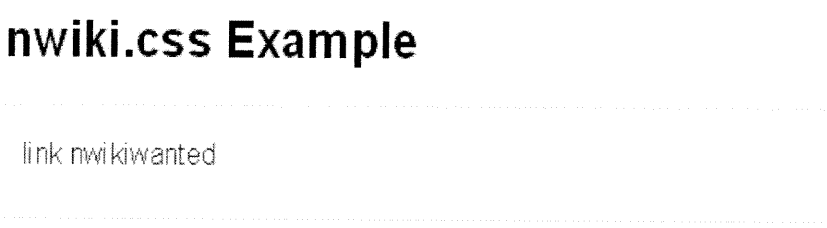
Clase references para div



nwiki.css Example

Page	User	Version	Created	Last modified
nwiki.css Example	DFWikiteam -	30	Monday, 11 June 2007 22:14	Wednesday, 13 2007 22:51 -1-
nwiki.css Example	DFWikiteam -	29	Monday, 11 June 2007 22:14	Wednesday, 13 June 2007 22:48
nwiki.css Example	DFWikiteam -	28	Monday, 11 June 2007 22:14	Wednesday, 13 June 2007 22:48
nwiki.css Example	DFWikiteam -	27	Monday, 11 June 2007 22:14	Wednesday, 13 2007 22:45 -2-

Clase nwikihighlight(1) y nwikibargroundblanco(2)



nwiki.css Example

link nwikiwanted

Clase nwikiwantedlink para link

10.3.3. Anexo para programadores DFwikiteam

Además de las clases anteriormente citadas, existen una serie de clases definidas en la hoja de estilos de la nwiki, y en la propia hoja de estilos standard de moodle, que nos ayudan a dar formato los diferentes tags de HTML.

Clases textleft, textright y textcenter

Clase incluida en nwiki.css que alinea texto a la izquierda, derecha y centro respectivamente.

Clases nwikileftnow, nwikirightnow y nwikicenternow para td

Clase incluida en nwiki.css. Añade a una celda la alineación de texto (izquierda, derecha y centro respectivamente) y el atributo NOWRAP. Su efecto será que el texto de esa celda no puede ser partido en varias líneas, para adecuarse al tamaño de la celda. Cuando esto sucede se dice que la tabla ha envuelto (wrap en inglés) el texto. El significado de NOWRAP es por tanto no envolver el texto. Si es necesario para mantener el texto en una sola línea se ignorarán los atributos WIDTH y HEIGHT si fueron usados.

Clase except

Clase incluida en nwiki.css, colorea de rojo el texto contenido.

Clase underline

Clase incluida en nwiki.css, subraya el texto contenido.

Clases boxalignleft, boxalignright y boxaligncenter

Clase propia de la hoja de estilos de moodle. Alinea las divisiones de HTML (tablas, divs, parrafos...) a la izquierda, derecha y centro respectivamente.

Clase generalbox

Clase propia de la hoja de estilos de moodle. Añade al tag un borde sólido, dándole apariencia de caja contenedora.

10.4. PHPDOC DE LA NWIKI

Para crear la documentación del código de la NWiki se ha utilizado la herramienta PHPDocumentor.

Link a la documentación del código de la NWiki:

http://morfeo.upc.es/crom/file.php/11/wikidoc/index.html

10.4.1. PHPDoc de la capa de presentación

/mod/wiki/weblib.php

Description

This library contains functions to generate XHTML strict output

- author: David Castro, Ferran Recio, Jordi Piguillem, UPC, and members of DFWikiteam listed at http://morfeo.upc.edu/crom
- author: DFWiki LABS
- author: Marc Alier i Forment
- version: $Id: weblib.php,v 1.25 2007/06/09 12:50:23 katiuska Exp $
- license: GNU Public License

Functions

wiki_a (line 489)

Defines an anchor. An anchor can be used in two ways:

1. To create a link to another document by using the href attribute.
2. To create a bookmark inside a document, by using the name or id attribute.

void wiki_a ([string $info = ''], [array $propert = null], [bool $return = false])

- string $info: defines the text.
- array $propert: is an object with several properties.
    - $propert->href, The target URL of the link.
    - $propert->name, Names an anchor. Use this attribute to create a bookmark in a document.
    - $propert->rel, Specifies the relationship between the current document and the target URL.
    - $propert->type, Specifies the MIME (Multipurpose Internet Mail Extensions) type of the target URL.
    - $propert->class, class of this anchor.
    - $propert->style, style of this anchor.
    - $propert->events, event attributes.
- bool $return: whether to return an output string or echo now.

---

wiki_b (line 1494)

Font style element. Renders as bold text.

void wiki_b ([string $info = ''], [array $propert = null], [bool $return = false])

- string $info: defines the text.
- array $propert: is an object with several properties.
    - $propert->class, class of this element.
    - $propert->style, style of this element.
    - $propert->events, event attributes.
- bool $return: whether to return an output string or echo now.

---

wiki_br (line 23)

Defines the start of a table.

void wiki_br ([integer $num = 1], [bool $return = false])

- bool $return: whether to return an output string or echo now.
- integer $num: num number of lines break.

---

wiki_change_column (line 207)

Change column.

void wiki_change_column ([array $propert = null], [bool $return = false])

- array $propert: is an object with several properties.
    - $propert->header, specifies if the table has a header.
    - $propert->align, specifies the horizontal alignment of cell content.
    - $propert->valign, specifies the vertical alignment of cell content.
    - $propert->colspan, indicates the number of columns this cell should span.
    - $propert->rowspan, indicates the number of rows this cell should span.
    - $propert->id, the id of the element.
    - $propert->class, the class of the element.
    - $propert->style, an inline style definition.
    - $propert->events, event attributes.
- bool $return: whether to return an output string or echo now.

---

wiki_change_li (line 1770)

Defines the start of a list item.

void wiki_change_li ([array $propert = null], [bool $return = false])

- array $propert: is an object with several properties.
    - $propert->class, class of this element.
    - $propert->style, style of this element.
    - $propert->events, event attributes.
- bool $return: whether to return an output string or echo now.

---

wiki_change_row (line 265)

Change row.

void wiki_change_row ([array $propert = null], [bool $return = false])

- array $propert: is an object with several properties.
  - $propert->header, specifies if the table has a header.
  - $propert->align, specifies the horizontal alignment of cell content.
  - $propert->valign, specifies the vertical alignment of cell content.
  - $propert->colspan, indicates the number of columns this cell should span.
  - $propert->rowspan, indicates the number of rows this cell should span.
  - $propert->id, the class of the element.
  - $propert->class, the class of the element.
  - $propert->classtr, the class of the element.
  - $propert->style, an inline style definition.
  - $propert->events, event attributes.
- bool $return: whether to return an output string or echo now.

---

wiki_convert_tabrows_to_tree (line 1898)

void wiki_convert_tabrows_to_tree ($tabrows, $selected, $inactive, $activated)

- $tabrows
- $selected
- $inactive
- $activated

---

wiki_convert_tree_to_html (line 1834)

void wiki_convert_tree_to_html ($tree, [$row = 0])

- $tree
- $row

---

wiki_div (line 1304)

Defines a division/section in a document.

void wiki_div ([string $info = ''], [array $propert = null], [bool $return = false])

- string $info: defines the text.
- array $propert: is an object with several properties.
    - $propert->class, class of this division/section.
    - $propert->style, style of this division/section.
    - $propert->events, event attributes.
- bool $return: whether to return an output string or echo now.

---

wiki_div_end (line 1367)

Print the end of a division/section.

void wiki_div_end ([bool $return = false])

- bool $return: whether to return an output string or echo now.

---

wiki_div_start (line 1338)

Defines a division/section in a document.

void wiki_div_start ([array $propert = null], [bool $return = false])

- array $propert: is an object with several properties.
    - $propert->id, id of this division/section.
    - $propert->class, class of this division/section.
    - $propert->style, style of this division/section.
    - $propert->events, event attributes.
- bool $return: whether to return an output string or echo now.

---

wiki_end_ol (line 1689)

Defines the end of an ordered list.

void wiki_end_ol ([bool $return = false])

- bool $return: whether to return an output string or echo now.

---

wiki_end_ul (line 1747)

Defines the end of an unordered list.

void wiki_end_ul ([bool $return = false])

- bool $return: whether to return an output string or echo now.

---

wiki_form (line 324)

Creates a form for user input. A form can contain textfields, checkboxes, radio-buttons and more. Forms are used to pass user-data to a specified URL.

void wiki_form ([string $info = ''], [array $propert = null], [bool $return = false])

- string $info: defines input info.
- array $propert: is an object with several properties.
    - $propert->action, defines where to send the data when the submit button is pushed.
    - $propert->method, the HTTP method for sending data to the action URL.
    - $propert->enctype, defines where to send the data when the submit button is pushed.
    - $propert->id, the id of the element.
    - $propert->class, the class of the element.
    - $propert->style, an inline style definition.
    - $propert->events, event attributes.
- bool $return: whether to return an output string or echo now.

---

wiki_form_end (line 412)

Print the end of a form.

void wiki_form_end ([bool $return = false])

- bool $return: whether to return an output string or echo now.

---

wiki_form_start (line 373)

Creates a form for user input. A form can contain textfields, checkboxes, radio-buttons and more. Forms are used to pass user-data to a specified URL.

void wiki_form_start ([array $propert = null], [bool $return = false])

- array $propert: is an object with several properties.
    - $propert->action, defines where to send the data when the submit button is pushed.
    - $propert->method, the HTTP method for sending data to the action URL.
    - $propert->enctype, defines where to send the data when the submit button is pushed.
    - $propert->id, the id of the element.
    - $propert->class, the class of the element.
    - $propert->style, an inline style definition.
    - $propert->events, event attributes.
- bool $return: whether to return an output string or echo now.

---

wiki_hr (line 1270)

Inserts a horizontal rule.

void wiki_hr ([array $propert = null], [bool $return = false])

- array $propert: is an object with several properties.
    - $propert->class, class of this style horizontal rule.

- $propert->style, style of this style horizontal rule.
- $propert->events, event attributes.

- bool $return: whether to return an output string or echo now.

---

wiki_img (line 1531)

Defines an image.

void wiki_img ([array $propert = null], [bool $return = false])

- array $propert: is an object with several properties.
  - $propert->src, the URL of the image to display.
  - $propert->alt, defines a short description of the image.
  - $propert->height, defines the height of an image.
  - $propert->width, sets the width of an image.
  - $propert->class, the class of this image.
  - $propert->style, style of this image.
  - $propert->events, event attributes.
- bool $return: whether to return an output string or echo now.

---

wiki_input_button (line 539)

Defines the start of an input field where the user can enter data.

void wiki_input_button ([array $propert = null], [bool $return = false])

- array $propert: is an object with several properties.
  - $propert->name, defines a unique name for the input element.
  - $propert->value, defines the text on the button.
  - $propert->disabled, disables the input element when it first loads so that the user can not write text in it, or select it.
  - $propert->size, defines the size of the input element.
  - $propert->id, a unique id for the element.
  - $propert->class, the class of the element.

- $propert->style, an inline style definition.
- $propert->events, event attributes.

- bool $return: whether to return an output string or echo now.

---

wiki_input_checkbox (line 593)

Defines the start of an input field where the user can enter data.

void wiki_input_checkbox ([array $propert = null], [bool $return = false])

- array $propert: is an object with several properties.
  - $propert->name, defines a unique name for the input element.
  - $propert->value, value for the input element.
  - $propert->disabled, disables the input element when it first loads so that the user can not write text in it, or select it.
  - $propert->size, defines the size of the input element.
  - $propert->checked, indicates that the input element should be checked when it first loads.
  - $propert->id, a unique id for the element.
  - $propert->class, the class of the element.
  - $propert->style, an inline style definition.
  - $propert->events, event attributes.
- bool $return: whether to return an output string or echo now.

---

wiki_input_file (line 650)

Defines the start of an input field where the user can enter data.

void wiki_input_file ([array $propert = null], [bool $return = false])

- array $propert: is an object with several properties.
  - $propert->name, defines a unique name for the input element.
  - $propert->disabled, disables the input element when it first loads so that the user can not write text in it, or select it.

- $propert->size, defines the size of the input element.
- $propert->accept, a comma-separated list of MIME types that indicates the MIME type of the file transfer.
- $propert->id, a unique id for the element.
- $propert->class, the class of the element.
- $propert->style, an inline style definition.
- $propert->events, event attributes.

- bool $return: whether to return an output string or echo now.

---

wiki_input_hidden (line 701)

Defines the start of an input field where the user can enter data.

void wiki_input_hidden ([array $propert = null], [bool $return = false])

- array $propert: is an object with several properties.
  - $propert->name, defines a unique name for the input element.
  - $propert->value, value for the input element.
  - $propert->id, a unique id for the element.
  - $propert->class, the class of the element.
  - $propert->style, an inline style definition.
  - $propert->events, event attributes.
- bool $return: whether to return an output string or echo now.

---

wiki_input_image (line 750)

Defines the start of an input field where the user can enter data.

void wiki_input_image ([array $propert = null], [bool $return = false])

- array $propert: is an object with several properties.
  - $propert->name, defines a unique name for the input element.
  - $propert->value, value for the input element.

- $propert->disabled, disables the input element when it first loads so that the user can not write text in it, or select it.
- $propert->size, defines the size of the input element.
- $propert->alt, defines an alternate text for the image.
- $propert->src, defines the URL of the image to display.
- $propert->id, a unique id for the element.
- $propert->class, the class of the element.
- $propert->style, an inline style definition.
- $propert->events, event attributes.

- bool $return: whether to return an output string or echo now.

---

wiki_input_password (line 809)

Defines the start of an input field where the user can enter data.

void wiki_input_password ([array $propert = null], [bool $return = false])

- array $propert: is an object with several properties.
  - $propert->name, defines a unique name for the input element.
  - $propert->value, value for the input element.
  - $propert->disabled, disables the input element when it first loads so that the user can not write text in it, or select it.
  - $propert->size, defines the size of the input element.
  - $propert->id, a unique id for the element.
  - $propert->class, the class of the element.
  - $propert->style, an inline style definition.
  - $propert->events, event attributes.
- bool $return: whether to return an output string or echo now.

---

wiki_input_radio (line 863)

Defines the start of an input field where the user can enter data.

void wiki_input_radio ([array $propert = null], [bool $return = false])

- array $propert: is an object with several properties.
    - $propert->name, defines a unique name for the input element.
    - $propert->value, value for the input element.
    - $propert->disabled, disables the input element when it first loads so that the user can not write text in it, or select it.
    - $propert->size, defines the size of the input element.
    - $propert->checked, indicates that the input element should be checked when it first loads.
    - $propert->id, a unique id for the element.
    - $propert->class, the class of the element.
    - $propert->style, an inline style definition.
    - $propert->events, event attributes.
- bool $return: whether to return an output string or echo now.

---

wiki_input_reset (line 919)

Defines the start of an input field where the user can enter data.

void wiki_input_reset ([array $propert = null], [bool $return = false])

- array $propert: is an object with several properties.
    - $propert->name, defines a unique name for the input element.
    - $propert->value, value for the input element.
    - $propert->disabled, disables the input element when it first loads so that the user can not write text in it, or select it.
    - $propert->size, defines the size of the input element.
    - $propert->id, a unique id for the element.
    - $propert->class, the class of the element.
    - $propert->style, an inline style definition.
    - $propert->events, event attributes.
- bool $return: whether to return an output string or echo now.

---

wiki_input_submit (line 972)

Defines the start of an input field where the user can enter data.

void wiki_input_submit ([array $propert = null], [bool $return = false])

- array $propert: is an object with several properties.
    - $propert->name, defines a unique name for the input element.
    - $propert->value, value for the input element.
    - $propert->disabled, disables the input element when it first loads so that the user can not write text in it, or select it.
    - $propert->size, defines the size of the input element.
    - $propert->id, a unique id for the element.
    - $propert->class, the class of the element.
    - $propert->style, an inline style definition.
    - $propert->events, event attributes.
- bool $return: whether to return an output string or echo now.

---

wiki_input_text (line 1027)

Defines the start of an input field where the user can enter data.

void wiki_input_text ([array $propert = null], [bool $return = false])

- array $propert: is an object with several properties.
    - $propert->name, defines a unique name for the input element.
    - $propert->value, value for the input element.
    - $propert->disabled, disables the input element when it first loads so that the user can not write text in it, or select it.
    - $propert->size, defines the size of the input element.
    - $propert->maxlength, defines the maximum number of characters allowed in a text field.
    - $propert->readonly, indicates that the value of this field cannot be modified.
    - $propert->id, a unique id for the element.
    - $propert->class, the class of the element.

    - $propert->style, an inline style definition.
    - $propert->events, event attributes.
- bool $return: whether to return an output string or echo now.

---

wiki_label (line 1084)

Defines a label to a control. If you click the text within the label element, it is supposed to toggle the control.

void wiki_label ([string $info = ''], [array $propert = null], [bool $return = false])

- string $info: defines the text.
- array $propert: is an object with several properties.
    - $propert->for, defines which form element the label is for, Set to an ID of a form element.
    - $propert->id, a unique id for the element.
    - $propert->class, the class of the element.
    - $propert->style, an inline style definition.
    - $propert->events, event attributes.
- bool $return: whether to return an output string or echo now.

---

wiki_link (line 438)

Defines the relationship between two linked documents.

void wiki_link ([array $propert = null], [bool $return = false])

- array $propert: is an object with several properties.
    - $propert->rel, defines the relationship between the current document and the targeted document.
    - $propert->type, specifies the MIME type of the target URL.
    - $propert->href, the target URL of the resource.
    - $propert->class, the class of the element.
    - $propert->style, an inline style definition.

- $propert->events, event attributes.
- bool $return: whether to return an output string or echo now.

---

wiki_optgroup (line 1180)

This element allows you to group choices. When you have a long list of options, groups of related choices are easier to handle.

void wiki_optgroup (string $options, [array $propert = null], [bool $return = false])

- string $options: defines an option in the drop-down list.
- array $propert: is an object with several properties.
  - $propert->label, defines the label for the option group.
  - $propert->disabled, disables the option-group when it first loads.
  - $propert->id, the id of the element.
  - $propert->class, the class of the element.
  - $propert->style, an inline style definition.
  - $propert->events, event attributes.
- bool $return: whether to return an output string or echo now.

---

wiki_option (line 1225)

Defines an option in the drop-down list.

void wiki_option ([string $info = ''], [array $propert = null], [bool $return = false])

- string $info: defines the text.
- array $propert: is an object with several properties.
  - $propert->disabled, specifies that the option should be disabled when it first loads.
  - $propert->selected, specifies that the option should appear selected (will be displayed first in the list).
  - $propert->label, defines a label to use when using <optgroup>.
  - $propert->value, defines the value of the option to be sent to the server.

  - $propert->class, the class of the element.
  - $propert->style, an inline style definition.
  - $propert->events, event attributes.
- bool $return: whether to return an output string or echo now.

---

wiki_paragraph (line 1460)

Defines a paragraph.

void wiki_paragraph ([string $info = ''], [array $propert = null], [bool $return = false])

- string $info: defines the text in a paragraph.
- array $propert: is an object with several properties.
  - $propert->class, class of this division/section.
  - $propert->style, style of this division/section.
  - $propert->events, event attributes.
- bool $return: whether to return an output string or echo now.

---

wiki_print_tabs (line 1801)

Returns a string containing a nested list, suitable for formatting into tabs with CSS.

void wiki_print_tabs (array $tabrows, [string $selected = NULL], [array $inactive = NULL], [array $activated = NULL], [$return = false])

- array $tabrows: An array of rows where each row is an array of tab objects
- string $selected: The id of the selected tab (whatever row it's on)
- array $inactive: An array of ids of inactive tabs that are not selectable.
- array $activated: An array of ids of other tabs that are currently activated
- $return

---

wiki_script (line 1620)

Defines a script, such as a JavaScript.

void wiki_script ([string $info = ''], [array $propert = null], [bool $return = false])

- string $info: defines the text.
- array $propert: is an object with several properties.
    - $propert->type, indicates the MIME type of the script.
    - $propert->src, defines a URL to a file that contains the script .
- bool $return: whether to return an output string or echo now.

---

wiki_select (line 1129)

Creates a drop-down list.

void wiki_select (string $options, [array $propert = null], [bool $return = false])

- string $options: defines an option in the drop-down list.
- array $propert: is an object with several properties.
    - $propert->disabled, disables the drop-down list.
    - $propert->name, defines a unique name for the drop-down list.
    - $propert->multiple, specifies that multiple items can be selected at a time.
    - $propert->size, defines the number of visible items in the drop-down list.
    - $propert->id, the id of the element.
    - $propert->class, the class of the element.
    - $propert->style, an inline style definition.
    - $propert->events, event attributes.
- bool $return: whether to return an output string or echo now.

---

wiki_size_text (line 1426)

The h1 to h6 tags define headers. h1 defines the largest header. h6 defines the smallest header.

void wiki_size_text ([string $info = ''], [string $num = 1], [array $propert = null],
[bool $return = false])

- string $info: defines the text.

- string $num: defines the number of the header.
- array $propert: is an object with several properties.
    - $propert->class, class of this division/section.
    - $propert->style, style of this division/section.
    - $propert->events, event attributes.
- bool $return: whether to return an output string or echo now.

---

wiki_span (line 1391)

Used to group inline-elements in a document.

void wiki_span ([string $info = ''], [array $propert = null], [bool $return = false])

- string $info: defines the text.
- array $propert: is an object with several properties.
    - $propert->class, class of this division/section.
    - $propert->style, style of this division/section.
    - $propert->events, event attributes.
- bool $return: whether to return an output string or echo now.

---

wiki_start_ol (line 1656)

Defines the start of an ordered list.

void wiki_start_ol ([array $propert = null], [bool $return = false])

- array $propert: is an object with several properties.
    - $propert->class, class of this element.
    - $propert->style, style of this element.
    - $propert->classli, class of this element.
    - $propert->styleli, style of this element.
    - $propert->events, event attributes.
- bool $return: whether to return an output string or echo now.

---

wiki_start_ul (line 1714)

Defines the start of an unordered list.

void wiki_start_ul ([array $propert = null], [bool $return = false])

- array $propert: is an object with several properties.
  - $propert->class, class of this element.
  - $propert->style, style of this element.
  - $propert->classli, class of this element.
  - $propert->styleli, style of this element.
  - $propert->events, event attributes.
- bool $return: whether to return an output string or echo now.

---

wiki_table_end (line 174)

Print the end of a table.

void wiki_table_end ([array $propert = null], [bool $return = false])

- array $propert: is an object with several properties.
  - $propert->header, specifies if the table has a header.
- bool $return: whether to return an output string or echo now.

---

wiki_table_start (line 72)

Defines the start of a table.

void wiki_table_start ([array $propert = null], [bool $return = false])

- array $propert: is an object with several properties.
  - $propert->border, specifies the border width. Set border="0" to display tables with no borders!

- $propert->width, specifies the width of the table.
- $propert->padding, specifies the space between the cell walls and contents.
- $propert->spacing, specifies the space between cells.
- $propert->class, the class of the element.
- $propert->id, the id of the element.
- $propert->style, an inline style definition.
- $propert->classtr, the class of the element.
- $propert->aligntd, specifies the horizontal alignment of cell content.
- $propert->valigntd, specifies the vertical alignment of cell content.
- $propert->colspantd, indicates the number of columns this cell should span.
- $propert->rowspantd, indicates the number of rows this cell should span.
- $propert->classtd, the class of the element.
- $propert->idtd, the class of the element.
- $propert->styletd, an inline style definition.
- $propert->header, specifies if the table has a header.
- $propert->alignth, specifies the horizontal alignment of cell content.
- $propert->valignth, specifies the vertical alignment of cell content.
- $propert->colspanth, indicates the number of columns this cell should span.
- $propert->rowspanth, indicates the number of rows this cell should span.
- $propert->idth, the class of the element.
- $propert->classth, the class of the element.
- $propert->styleth, an inline style definition.
- $propert->events, event attributes.

- bool $return: whether to return an output string or echo now.

---

wiki_textarea (line 1579)

Defines a text-area (a multi-line text input control).

void wiki_textarea ([number $info = ''], [array $propert = null], [bool $return = false])

- number $info: defines the text.
- array $propert: is an object with several properties.

  - $propert->cols, specifies the number of columns visible in the text-area.
  - $propert->rows, specifies the number of rows visible in the text-area.
  - $propert->name, specifies a name for the text-area.
  - $propert->class, the class of this image.
  - $propert->style, style of this image.
  - $propert->events, event attributes.
- bool $return: whether to return an output string or echo now.