

**TInput**

(PHP 4, PHP 5)

Localización: Tin.php

**Descripción:**

Esta clase permite crear elementos de un formulario de forma muy sencilla, los elementos que permite crear son:  
text  
password  
chackbox  
radio  
select  
submit  
reset  
textarea  
file  
hidden  
image  
button  
\*datepicker  
\*\*new

\* Este elemento permite seleccionar una fecha a partir de un calendario grafico  
\*\* permite introducir dentro de la estructura cualquier elemento ejemplo un sistema captcha

La esencia del funcionamiento de esta clase es general código html (elementos de formularios) de forma automatizada agilizando el proceso de creación, como se verá en un ejemplo posteriormente.

**Los metodos:**

**TInput**($input\_type)  
**setInput**($input\_name,$input\_label,$input\_id,$input\_parameters,$arr\_case\_content='')  
**getInput**()  
**getInput\_tab**()  
**renderInput**()  
**getInputvalues**($typekey='num')

**Los parametros:**

**TInput**($input\_type)

$input\_type: Recibe los nombres de los elementos que queremos crear entre comillas (cadena de texto) y separados por coma (,), los nombres de los elementos son los que aparecen en la descripción.

**setInput**($input\_name,$input\_label,$input\_id,$input\_parameters,$arr\_case\_content='')

$input\_name: Recibe los nombres (de la etiqueta name de HTML) de los elementos a crear, entre comillas (cadena de texto) separado por slash y coma (/,), o se puede usar la funcion **do\_arr**() para crear un arreglo con los nombres de los elementos, pero si se usa de esta forma el resto de los parámetros deben usar también la función **do\_arr**(). El motivo por el cual se separan los parámetros por “/,” es porque en los idiomas que existe ingles, francés, español, etc. Las oraciones ni las palabras se separan por /, esto garantiza que no haya conflicto o errores a la hora de interpretar los parámetros.

$input\_label: Este parámetro solo funciona para cuando se usa el método **getInput\_tab**() que significa obtener el código generado por el método, tabulado o sea los label en una columna y los componentes en otra al lado

$input\_id: Este parámetro le agrega la etiqueta id de html, a los parámetros, separado por “/,”, la cual permite la manipulación del DOM mediante java script

$input\_parameters: Mediante este parámetro se pueden pasar el resto de los parámetros html de las etiquetas como:

text  
password  
chackbox  
radio  
select  
submit  
reset   
file   
textarea  
hidden  
image  
button

También separado por /, note que acá no se incluye el datepicker y el new.

$arr\_case\_content: este parámetro es opcional solo lo usan el datepicker,el textarea y el select

-Para datepicker permite guardar la configuración del mismo:

Estos parámetros de configuración son:

'%d-%m-%Y', <<formato de la fecha

singleClick : true,

monthsInRow : 1,

vertical : true,

date: today

El resto se explica por sí solo, pero no es necesario aprenderse de memoria todos los parámetros del datepicker ya que existen actualmente 2 constantes que te permiten abreviarte este paso estas son :

DATEPICKER: configura el datepicker para que solo se pueda seleccionar los días posteriores al día actual, o sea no te deje seleccionar días que ya pasaron

DATEPICKER1: es la configuración más común.

Para más información visite: <http://www.dynarch.com/projects/calendar>

Para el textarea permite guardar el contenido del mismo en este parámetro, supongamos que creamos un textarea y necesitamos que tenga un texto, no hay ningún parámetro como el value de un text, por tanto esta es la solución.

Para el select mediante este parámetros se le pasan las etiquetas options, nos podemos auxiliar de la función **setSelect\_optiontag**("opcion1/,opcion2/,opcion3/,opcion4")

Hay que tener en cuenta que este parámetro es un arreglo cuyas key son numéricas, y cuyos espacios no tienen por que estar lleno todos, por ejemplo si yo creo un textarea de tercero el espacio en que debe estar guardado el texto a mostrar en el textaera es el dos porque los arreglos empiezan en 0.

Para todos los elementos que se crean el orden de los parámetros es el mismo, por ejemplo si creamos un button de segundo el nombre que le corresponde es el segundo, como se verá en el ejemplo de más abajo.

**getInput**()

Genera y devuelve todo el código de los elementos

**getInput\_tab**()

Lo mismo que el anterior pero devuelve los elementos tabulados con los labels

**renderInput**()

Imprime el código generado

**getInputvalues**($typekey='num')

una vez enviado el formulario recibe las variables enviadas en el mismo orden que están los elementos, en un arreglo para su posterior manipulación. Este método debe ser declarado después de **setInput**

$typekey: por defecto trae el valor num lo cual significa que el arreglo será manipulado por llaves numéricas, el otro valor es txt, entonces se maneja el array con los nombres de los elementos

Ejemplo de jerarquía que debe seguirse :

$obj= new **TInput**(‘[elementos...]’); // creamos la instancia de la clase

$obj-> **setInput**([parámetros…]); //ponemos las propiedades y los parámetros necesarios de aquí para abajo no importa el orden

$obj-> **getInput**(); $obj-> **getInput\_tab**(); **renderInput**() **getInputvalues**($typekey='num')

//estos cuatros métodos se pueden usar en el orden que se desee

Nota1: para que funcionen los elementos deben estar dentro de un formulario, esto se logra implementando la clase **TForm()**, ver TForm.pdf

Nota2: para que los wysiwyg y el calendario funcione debe ponerse la función **include\_js\_css\_enginer**($path='',$lang='es') preferentemente en el head de la pagina ya que esta incluye los archivos javascript, css y de idiomas necesario, el parámetro $path es para indicar el camino donde se encuentran las carpetas del framework, $lang permite especificar el lenguaje a usar debe definirse con las dos primeras letras.

Ejemplo:

$arr[0]=setSelect\_optiontag("opcion1/,opcion2/,opcion3/,opcion4"); //parámetro para las opciones desplegables

$arr[5]=DATEPICKER1; //parámetros necesarios para hacer que el datepicker funcione

$arr[6]="contenido del textarea"; //parámetros para el contenido del textarea

$in=new TInput("select,text,password,checkbox,radio,datepicker,textarea,submit,reset");

$in->setInput(

'nombre\_select/,nombre\_text/,nombre\_pass/,nombre\_chek/,nombre\_radio/,nombre\_date/,nombre\_textarea/,nombre\_sub/,nombre\_reset', //estos son los nombre de cada elemento en ese mismo orden

'label\_select/,label\_text/,label\_pass/,label\_chek/,label\_radio/,label\_date/,labele\_textarea/,label\_sub/,label\_reset',

//texto que aparecerá delante de cada elemento, para que aparezca debemos usar la funcion $in->getInput\_tab()

'id\_select/,id\_text/,id\_pass/,id\_chek/,id\_radio/,id\_date/,id\_textarea/,id\_sub/,id\_reset', //este es el id de cada elemento por el orden que aparecen, se puede dejar en blanco ya que funciona igual

"/,value=texto/,/,checked=true/,/,/,rows=4 cols=50/,value='Enviar formulario'/,value='Restablecer'", //los parametros de cada elemento son los mismos parámetros que se usan en el html, se pueden dejar en blanco

$arr //este parámetro es un arreglo para los elementos que no son del tipo <input type... ejemplo datepicker, textarea, etc. los parametros deben guardarse en el espacio del arreglo que le corresponde a la posición del elemento empezando por el cero

);

///////////////////////////////////creamos el formulario///////////////////////////////////////////////

$form= new TForm();

$form->setForm(

'nombre\_form', //nombre del formulario

'post', //metodo

' ', //action se puede dejar en blanco para que accione sobre la misma pagina que esta definido

'id\_form', //id del formulario

VALIDATE\_FORM, //parámetros que se le pueden pasar al formulario ejemplo onsubmit='return validate()'

$in->getInput\_tab() //este es el contenido del formulario son los elementos que creamos en el bloque anterior

);

$form->renderForm();