RichWare TTags class

Ver 2.02

The TTags class provides a standard for coding html in php. The developer can assign attributes, global attributes, events and data attribute. Once the attributes have been set there are functions that will create the html tags verifying the structure is correct.

Attributes:

Attribute	Description
accept	Elements: <form>, <input> Notes: List of types the server accepts, typically a file type.
accept_charset	Elements: <form> Notes: List of supported charsets.
action	Elements: <form> Notes: The URI of a program that processes the information submitted via the form.
align	Elements: <applet>, <caption>, <col>, <colgroup>, <hr>, <iframe>, <img>, <table>, <tbody>, <td>, <tfoot> , <th>, <thead>, <tr> Notes: Specifies the horizontal alignment of the element.
alt	Elements: <applet>, <area>, <img>, <input> Notes: Alternative text in case an image can't be displayed.
aria_hidden	Elements: Indicates that the element and all of its descendants are not visible or perceivable to any user as implemented by the author. (https://developer.paciellogroup.com/blog/2012/05/html5-accessibility-chops-hidden-and-aria-hidden/)
async	Elements: <script> Notes: Indicates that the script should be executed asynchronously.
autocomplete	Elements: <form>, <input> Notes: Indicates whether controls in this form can by default have their values automatically completed by the browser.
autofocus	Elements: <button>, <input>, <keygen>, <select>, <textarea> Notes: The element should be automatically focused after the page loaded.
autoplay	Elements: <audio>, <video> Notes: The audio or video should play as soon as possible.
autosave	Elements: <input> Notes: Previous values should persist dropdowns of selectable values across page loads.
buffered	Elements: <audio>, <video> Notes: Contains the time range of already buffered media.
challenge	Elements: <keygen> Notes: A challenge string that is submitted along with the public key.
charset	Elements: <meta>, <script> Notes: Declares the character encoding of the page or script.
checked	Elements: <command>, <input> Notes: Indicates whether the element should be checked on page load.
cite	Elements: <blockquote>, <del>, <ins>, <q> Notes: Contains a URI which points to the source of the quote or change.
code	Elements: <applet> Notes: Specifies the URL of the applet's class file to be loaded and executed.
codebase	Elements: <applet> Notes: This attribute gives the absolute or relative URL of the directory where applets' .class files referenced by the code attribute are stored.
cols	Elements: <textarea> Notes: Defines the number of columns in a textarea.

Attribute	Description
colspan	Elements: <td>, <th> Notes: The colspan attribute defines the number of columns a cell should span.
content	Elements: <meta> Notes: A value associated with http-equiv or name depending on the context.
controls	Elements: <audio>, <video> Notes: Indicates whether the browser should show playback controls to the user.
coords	Elements: <area> Notes: A set of values specifying the coordinates of the hot-spot region.
crossorigin	Elements: <audio>, <img>, <link>, <script>, <video> Notes: How the element handles cross-origin requests
data	Elements: <object> Notes: Specifies the URL of the resource.
datetime	Elements: <del>, <ins>, <time> Notes: Indicates the date and time associated with the element.
default	Elements: <track> Notes: Indicates that the track should be enabled unless the user's preferences indicate something different.
defer	Elements: <script> Notes: Indicates that the script should be executed after the page has been parsed.
dirname	Elements: <input>, <textarea> Notes:
disabled	Elements: <button>, <command>, <fieldset>, <input>, <keygen>, <optgroup>, <option>, <select>, <textarea> Notes: Indicates whether the user can interact with the element.
download	Elements: <a>, <area> Notes: Indicates that the hyperlink is to be used for downloading a resource.
enctype	Elements: <form> Notes: Defines the content type of the form date when the method is POST.
for	Elements: <label>, <output> Notes: Describes elements which belongs to this one.
form	Elements: <button>, <fieldset>, <input>, <keygen>, <label>, <meter>, <object>, <output>, <progress>, <select>, <textarea> Notes: Indicates the form that is the owner of the element.
formaction	Elements: <input>, <button> Notes: Indicates the action of the element, overriding the action defined in the <form>.
formnovalidate	Elements: <input> When present, it specifies that the <input> element should not be validated when submitted.
headers	Elements: <td>, <th> Notes: IDs of the <th> elements which applies to this element.
height	Elements: <img> Notes: Specifies the height of an image
high	Elements: <meter> Notes: Indicates the lower bound of the upper range.
href	Elements: <a>, <area>, <base>, <link> Notes: The URL of a linked resource.
hreflang	Elements: <a>, <area>, <link> Notes: Specifies the language of the linked resource.
http_equiv	Elements: <meta> Notes:
icon	Elements: <command> Notes: Specifies a picture which represents the command.
ismap	Elements: <img> Notes: Indicates that the image is part of a server-side image map.
keytype	Elements: <keygen> Notes: Specifies the type of key generated.
kind	Elements: <track> Notes: Specifies the kind of text track.
label	Elements: <track> Notes: Specifies a user-readable title of the text track.
language	Elements: <script> Notes: Defines the script language used in the element.

Attribute	Description
list	Elements: <input> Notes: Identifies a list of pre-defined options to suggest to the user.
loop	Elements: <audio>, <bgsound>, <marquee>, <video> Notes: Indicates whether the media should start playing from the start when it's finished.
low	Elements: <meter> Notes: Indicates the upper bound of the lower range.
manifest	Elements: <html> Notes: Specifies the URL of the document's cache manifest.
max	Elements: <input>, <meter>, <progress> Notes: Indicates the maximum value allowed.
maxlength	Elements: <input>, <textarea> Notes: Defines the maximum number of characters allowed in the element.
media	Elements: <a>, <area>, <link>, <source>, <style> Notes: Specifies a hint of the media for which the linked resource was designed.
method	Elements: <form> Notes: Defines which HTTP method to use when submitting the form. Can be GET (default) or POST.
min	Elements: <input>, <meter> Notes: Indicates the minimum value allowed.
minlength	Elements: <input>, <textarea> Notes: Defines the minimum number of characters allowed in the element.
multiple	Elements: <input>, <select> Notes: Indicates whether multiple values can be entered in an input of the type email or file.
muted	Elements: <audio>, <video> Notes: Indicates whether the audio will be initially silenced on page load.
name	Elements: <button>, <form>, <fieldset>, <iframe>, <input>, <keygen>, <object>, <output>, <select>, <textarea>, <map>, <meta>, <param> Notes: Name of the element. For example used by the server to identify the fields in form submits.
novalidate	Elements: <form> Notes: This attribute indicates that the form shouldn't be validated when submitted.
open	Elements: <details> Notes: Indicates whether the details will be shown on page load.
optimum	Elements: <meter> Notes: Indicates the optimal numeric value.
pattern	Elements: <input> Notes: Defines a regular expression which the element's value will be validated against.
ping	Elements: <a>, <area> Notes:
placeholder	Elements: <input>, <textarea> Notes: Provides a hint to the user of what can be entered in the field.
poster	Elements: <video> Notes: A URL indicating a poster frame to show until the user plays or seeks.
preload	Elements: <audio>, <video> Notes: Indicates whether the whole resource, parts of it or nothing should be preloaded.
radiogroup	Elements: <command> Notes:
readonly	Elements: <input>, <textarea> Notes: Indicates whether the element can be edited.
rel	Elements: <a>, <area>, <link> Notes: Specifies the relationship of the target object to the link object.
required	Elements: <input>, <select>, <textarea> Notes: Indicates whether this element is required to fill out or not.
reversed	Elements: <ol> Notes: Indicates whether the list should be displayed in a descending order instead of a ascending.

Attribute	Description
rows	Elements: <textarea> Notes: Defines the number of rows in a text area.
rowspan	Elements: <td>, <th> Notes: Defines the number of rows a table cell should span over.
sandbox	Elements: <iframe> Notes:
scope	Elements: <th> Notes:
scoped	Elements: <style> Notes:
seamless	Elements: <iframe> Notes:
selected	Elements: <option> Notes: Defines a value which will be selected on page load.
shape	Elements: <a>, <area> Notes:
size	Elements: <input>, <select> Notes: Defines the width of the element (in pixels). If the element's type attribute is text or password then it's the number of characters.
sizes	Elements: <link>, <img>, <source> Notes:
span	Elements: <col>, <colgroup> Notes:
src	Elements: <audio>, <embed>, <iframe>, <img>, <input>, <script>, <source>, <track>, <video> Notes: The URL of the embeddable content.
srcdoc	Elements: <iframe> Notes:
srclang	Elements: <track> Notes:
srcset	Elements: <img> Notes:
start	Elements: <ol> Notes: Defines the first number if other than 1.
step	Elements: <input> Notes:
summary	Elements: <table> Notes:
target	Elements: <a>, <area>, <base>, <form> Notes:
type	Elements: <button>, <input>, <command>, <embed>, <object>, <script>, <source>, <style>, <menu> Notes: Defines the type of the element.
usemap	Elements: <img>, <input>, <object> Notes:
value	Elements: <button>, <option>, <input>, <li>, <meter>, <progress>, <param> Notes: Defines a default value which will be displayed in the element on page load.
width	Elements: <img>
wrap	Elements: <textarea> Notes: Indicates whether the text should be wrapped.

Global Attributes:

Attribute	Description
accesskey	Elements: Global attribute Notes: Defines a keyboard shortcut to activate or add focus to the element.
class	Elements: Global attribute Notes: Often used with CSS to style elements with common properties.
contenteditable	Elements: Global attribute Notes: Indicates whether the element's content is editable.
contextmenu	Elements: Global attribute Notes: Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Elements: Global attribute Notes: Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Elements: Global attribute Notes: Defines whether the element can be dragged.

Attribute	Description
dropzone	Elements: Global attribute Notes: Indicates that the element accept the dropping of content on it.
hidden	Elements: Global attribute Notes: Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
id	Elements: Global attribute Notes: Often used with CSS to style a specific element. The value of this attribute must be unique.
itemid	https://www.w3.org/TR/microdata/#encoding-microdata
itemprop	Elements: Global attribute Notes: https://www.w3.org/TR/microdata/#encoding-microdata
itemref	https://www.w3.org/TR/microdata/#encoding-microdata
itemscope	https://www.w3.org/TR/microdata/#encoding-microdata
itemtype	https://www.w3.org/TR/microdata/#encoding-microdata
lang	Elements: Global attribute Notes: Defines the language used in the element.
slot	Elements: Global attribute Notes: Assigns a slot in a shadow DOM shadow tree to an element.
spellcheck	Elements: Global attribute Notes: Indicates whether spell checking is allowed for the element.
style	Elements: Global attribute Notes: Defines CSS styles which will override styles previously set.
tabindex	Elements: Global attribute Notes: Overrides the browser's default tab order and follows the one specified instead.
title	Elements: Global attribute Notes: Text to be displayed in a tooltip when hovering over the element.

Events:

Event	Description
onafterprint	Script to be run after the document is printed
onbeforeprint	Script to be run before the document is printed
onbeforeunload	Script to be run when the document is about to be unloaded
onerror	Script to be run when an error occurs
onhashchange	Script to be run when there has been changes to the anchor part of the a URL
onload	Fires after the page is finished loading
onmessage	Script to be run when the message is triggered
onoffline	Script to be run when the browser starts to work offline
ononline	Script to be run when the browser starts to work online
onpagehide	Script to be run when a user navigates away from a page
onpageshow	Script to be run when a user navigates to a page
onpopstate	Script to be run when the window's history changes
onresize	Fires when the browser window is resized
onstorage	Script to be run when a Web Storage area is updated
onunload	Fires once a page has unloaded (or the browser window has been closed)
onblur	Fires the moment that the element loses focus
onchange	Fires the moment when the value of the element is changed
oncontextmenu	Script to be run when a context menu is triggered

onfocus	Fires the moment when the element gets focus
oninput	Script to be run when an element gets user input
oninvalid	Script to be run when an element is invalid
onreset	Fires when the Reset button in a form is clicked
onsearch	Fires when the user writes something in a search field (for <input="search">)
onselect	Fires after some text has been selected in an element
onsubmit	Fires when a form is submitted
onkeydown	Fires when a user is pressing a key
onkeypress	Fires when a user presses a key
onkeyup	Fires when a user releases a key
onclick	Fires on a mouse click on the element
ondblclick	Fires on a mouse double-click on the element
onmousedown	Fires when a mouse button is pressed down on an element
onmousemove	Fires when the mouse pointer is moving while it is over an element
onmouseout	Fires when the mouse pointer moves out of an element
onmouseover	Fires when the mouse pointer moves over an element
onmouseup	Fires when a mouse button is released over an element
onwheel	Fires when the mouse wheel rolls up or down over an element
ondrag	Script to be run when an element is dragged
ondragend	Script to be run at the end of a drag operation
ondragenter	Script to be run when an element has been dragged to a valid drop target
ondragleave	Script to be run when an element leaves a valid drop target
ondragover	Script to be run when an element is being dragged over a valid drop target
ondragstart	Script to be run at the start of a drag operation
ondrop	Script to be run when dragged element is being dropped
onscroll	Script to be run when an element's scrollbar is being scrolled
oncopy	Fires when the user copies the content of an element
oncut	Fires when the user cuts the content of an element
onpaste	Fires when the user pastes some content in an element
onabort	Script to be run on abort
oncanplay	Script to be run when a file is ready to start playing (when it has buffered enough to begin)
oncanplaythrough	Script to be run when a file can be played all the way to the end without pausing for buffering
oncuechange	Script to be run when the cue changes in a <track> element
ondurationchange	Script to be run when the length of the media changes
onemptied	Script to be run when something bad happens and the file is suddenly unavailable (like unexpectedly disconnects)
onended	Script to be run when the media has reach the end (a useful event for messages like "thanks for listening")
onloadeddata	Script to be run when media data is loaded
onloadedmetadata	Script to be run when meta data (like dimensions and duration) are loaded
onloadstart	Script to be run just as the file begins to load before anything is actually loaded
onpause	Script to be run when the media is paused either by the user or programmatically
onplay	Script to be run when the media is ready to start playing

onplaying	Script to be run when the media actually has started playing
onprogress	Script to be run when the browser is in the process of getting the media data
onratechange	Script to be run each time the playback rate changes (like when a user switches to a slow motion or fast forward mode)
onseeked	Script to be run when the seeking attribute is set to false indicating that seeking has ended
onseeking	Script to be run when the seeking attribute is set to true indicating that seeking is active
onstalled	Script to be run when the browser is unable to fetch the media data for whatever reason
onsuspend	Script to be run when fetching the media data is stopped before it is completely loaded for whatever reason
ontimeupdate	Script to be run when the playing position has changed (like when the user fast forwards to a different point in the media)
onvolumechange	Script to be run each time the volume is changed which (includes setting the volume to "mute")
onwaiting	Script to be run when the media has paused but is expected to resume (like when the media pauses to buffer more data)
onshow	Fires when a <menu> element is shown as a context menu
ontoggle	Fires when the user opens or closes the <details> element

Data Attributes:

Attribute	Description
data_multiple_caption	
data_tab	

Functions:

If function where *newline* is an option, a new line character (\n) will be appended to the tag.

Option	description
tagname	Name of the html tag. This is not verified or checked so any name can be entered. Invalid name will produce unpredictable results.
tagnames	This is a list of comma separated list of tagnames.
newline	If is ‘true’, a new line character (\n) will be appended.

Function	Description
End(*tagname*,*newline*)	Sets the end of tag, e.g. </*tagname*>
EndTags(*tagnames*,*newline*)	Performs the End() function for each name in the *tagnames list*.
Reset()	All attributes, global attributes, events and data attributes will be cleared.
Start(*tagname*,*newline*)	Creates an html tag for the *tagname* with all option that have been selected included with the tag.

Function	Description
StartTags(*tagnames,newline*)	Performs the Start() function for each name in the *tagnames list*. Please note, each attribute will be included for each tag name.
ToolTip(*tagname,tooltip class*)	Creates html as follows: <span class="*tooltip class*" data-help="*tagname*">?<</span>

Examples:

```
    $tags = new \RichWare\Tags\v2_02\TTags();
    $html = null;
    $iconFile = "c:\icons\somefile.png";
    $tags->Reset();
    $tags->src = $iconFile;
    $tags->alt = "File Upload";
    $tags->height = "42";
    $tags->width = "42";
    $html .= $tags->start('img');
    $tags->Reset();
    $html .= $tags->Start('br');
```

The resulting HTML:

```
<img alt="File Upload" height="42" src="c:\icons\somefile.png" width="42" >
<br />
```