

BLOGSENSE-WP POST TEMPLATING DOCUMENTATION & GUIDE

SPYNTAX SHORTCODE

The Shortcode

`[spyntax] INSERT CONTENT HERE [/spyntax]`

What is Spyntax?

Spyntax is a industry wide term for content coded to have mutiple versions of itself, if ran through a script that reads it.

An article that is coded in spyntax looks like this:

Create & schedule video posts from RSS feeds with Youtube or Hulu video inside. {You can|You'll be {in a position|able|ready } to|You {in all probability|most likely|perhaps|possibly} can|You may|You'll|It is easy to|You could|Possible|One can|You can still|You will|One could|You can actually} use this module to create large niche video {web sites|websites|web pages|internet sites} covering almost any topic.

And will output like this:

#1

Create & schedule video posts from RSS feeds with Youtube or Hulu video inside. One could use this module to create large niche video web pages covering almost any topic.

#2

Create & schedule video posts from RSS feeds with Youtube or Hulu video inside. You can still use this module to create large niche video web pages covering almost any topic.

When hooking spyntax please place your spyntaxed articles into the appropriate tags like so:

[spyntax]*Create & schedule video posts from RSS feeds with Youtube or Hulu video inside. {You can|You'll be {in a position|able|ready } to|You {in all probability|most likely|perhaps|possibly} can|You may|You'll|It is easy to|You could|Possible|One can|You can still|You will|One could|You can actually} use this module to create large niche video {web sites|websites|web pages|internet sites} covering almost any topic.***[/spyntax]**

CUSTOM IMAGES SHORTCODE

BlogSense creates image pools from a handful of provided image urls, and then for each hook, selects an image from that pool at random. There are three source options for images; a local folder, Flickr.com, and Picasa.com.

Properties you can edit per hook:

1. Image Floating
CSS Classes
Max-Width
Max-Height

Images from a Local Server Folder

1. Place image files into a sub-folder and drop that sub-folder into your `/my-images/` ftp directory. BlogSense will in-source all of those photos, giving you a second round of selecting where you can deselect photos you do not want to use for this particular header/footer profile.
2. Place the correct relative path into the setting that asks for it. Usually you will just have to add your subfolder's name onto the already entered path to you `/my-images/` directory.
3. Press "Import Images". Use your mouse to deselect or select images to add to your random image pool. Press "Hook" to hook content.

Hooked content will look like this:

```
[hook type="image" pool="http://www.blogsense-wp.com/testblog/auto/my-images/dogs/worlds-strongest-dog.jpg,http://www.blogsense-wp.com/testblog/auto/my-images/dogs/93037486_94f5373edb_o.jpg,http://www.blogsense-wp.com/testblog/auto/my-images/dogs/514439499_ff22ed3a31_o.jpg" css="myclass" max_width="400" max_height="600" float="right"]
```

When posting items, BlogSense will detect this syntax and turn it into an image with the appropriate settings. Only one random image will be selected from the url pool .

Images from Flickr & Picasa

Hooking images from Flickr & Picasa works in the same exact way as the above mentioned. BlogSense uses the xml api of Flickr and Picasa to pull images from customized searches, then allows you to browse through them and further select

which ones you are wanting to use.

All images will be saved to server is same server image hosting is set to on.

CUSTOM YOUTUBE SHORTCODE

Settings:

1. CSS Class (Will add a class to the div that surrounds the description if enabled.
Video Width (Width of embedded video)
Video Height (Height of embedded video)
Include Description (Option to include a description of the video or only source the video alone.
Max Results (Results to pull from youtube api)

This content hook creates a pool of random youtube videos from youtube, via a keyword search, and then provides you with the ability to slim down your selection by clicking on the videos to disable selected and enabled selected. One video will be randomly selected from pool when article containing the hook is published.

Hooked content will look like this:

```
[hook type="video" pool="http://www.youtube.com/watch?v=hINrmhSVm08&feature=youtube_gdata,http://www.youtube.com/watch?v=J_q3TqokA_4&feature=youtube_gdata,http://www.youtube.com/watch?v=QWJDOX5cA9Y&feature=youtube_gdata" css="vid" width="425" height="385" descriptions="on"]
```

When posting items, BlogSense will detect this syntax and turn it into an video with settings applied. Only one random video will be selected from the url pool.

TOKENS

Tokens are signals to BlogSense to try and pull content automatically from a source. Each token has 1-4 unique parameters that pass information along to BlogSense.

Using the title of the post, a keyword phrase is generated. This happens by removing common words such as (and or the a of etc.) and the using the remaining string as a search query.

In some cases (such as the youtube token, a user can manually enter in a keyword to search).

[GoogleIMG] Token

example : {googleimg:%title_filtered%:1:275px:300px:first}

example : {googleimg:keyword here:1:275px:300px:random}

Parameter 2 – Keywords to search

Parameter 3 – # of images to pull.

Parameter 4 – max width of image.

Parameter 5 – max height of image.

Parameter 6 - Sort nature. Options are 'first' to return best result and 'random' to choose an image randomly from the results.

[Amazon] Token

example: {amazon:4:content:%title_filtered%}

Parameter 1 – token id.

Parameter 2 – number of items to display

Parameter 3 – mode. (options are 1. **content** 2. **widget**)

Parameter 4 – The keywords

note: content mode will look for items, and use the top rated review for content. Widget mode will use images and titles as relevant product links.

[Youtube] Token

example : {youtube:1}

example : {youtube:1:keyword here:exclude keywords here}

Parameter 2 – number of videos to display

Parameter 3 – keywords to search.

Parameter 4 – (optional) exclude keywords... searches title. Please separate multiple keywords with commas.

[Ebay] Token

This token creates an ebay javascript widget. It may be better to use shortcodes associated with a third party application such as phpBay.

example: {ebay:5336369034:5:400px}

Parameter 2 – Ebay Campaign ID found [here](#):

Parameter 3 – Number of items to display on widget.

Parameter 4 – Width of widget.

[Flickr] Token

This token returns broad results. If your goal is quality its recommended that you use the images import feature.

example: {flickr:1:275px:300px}

Parameter 2 – Number of images.

Parameter 3 – Max width of images.

Parameter 4 – Max height of images

[RSS] Token

This token selects an RSS item from an RSS feed at random and returns the content in a format defined by the RSS Token Template.

example:

{rss:http://www.blogsense-wp.com/news/rss/:1:<div><p>
<i><u><table><a>}

Parameter 2 – URL of RSS Feed.

Parameter 3 – Number of RSS Items to select at random and add to post.

Parameter 4 – HTML tags NOT to be stripped out. See PHP's [striptags\(\)](#) function for a list of applicable parameters.

[IFNOIMAGE] Token

This shortcode was created to help generate image content when no other images were found in the original post.

example:

[IFNOIMAGE]{googleimg:%title_filtered%:1:275px:300px:first}[IFNOIMAGE]

[INSERT] Token

This token which encloses the content in [insert][insert] tags will allow the user to insert content into the middle of the main content block. The main content block refers to the content provided by the primary source of the campaign, typically represented by the %postbody% templating variable.

example: [insert:middle]content here[/insert]

Parameter 2 – This parameter has 3 options:

1) middle -will place content directly in middle of content.

2) #p - (replace # with a numeric value) – will place content after # occurrence of </p>

Numeric Value - A numeric value will place the content after Xth character. (x being the number)

More Examples:

- 1) `[insert:3p]content here[/insert]` – places content after 3rd paragraph.
`[insert:1000]content here[/insert]` -places content after 1000th character.

[DATE] TOKEN

This token uses PHP's `date()` function to generate a date for use inside our templates.

example: `[date format="F j, Y"]`
This will output: `March 10, 2001`

Please use <http://php.net/manual/en/function.date.php> to determine which format is best for you.

[SUBSTRING] Token

This token was added primarily so we would have a way to turn full content into partial content, but also makes use of the wide variety of string manipulations that PHP's `substr()` offers. Below are a couple examples of how you might use this token.

Param 1: start value
Param 2: `int(length)`.

Example Usages:

```
[substring:0:3]abcdef[/substring] // returns "abc"  
[substring:0:8]abcdef[/substring] // returns "abcdef"  
[substring:1:0]abcdef[/substring] // returns "bcdef"  
[substring:1:3]abcdef[/substring] // returns "bcd"
```

```
[substring:-1:1]abcdef[/substring] // returns "f"  
[substring:-2:2]abcdef[/substring] // returns "ef"  
[substring:-3:3]abcdef[/substring] // returns "def"
```

More information about `substr()` at <http://php.net/manual/en/function.substr.php>

Advanced Token Arrangements: [rand] token

To attach multiple tokens in one profile please use the following formula.
`[rand]{token:1}/{token:2}[/rand]`

POSTBODY VARIABLES

When creating or editing a campaign; depending on which module you are working in you will find that you have a handful of variables you can use. These variables represent some other form of content. Here is a list variables you will find within BlogSense:

- %title%
 - *Original title of content*
- %link%
 - *Original link of content; If Amazon this link will be affiliate link.*
- %postbody%
 - *(RSS, Sources, Drop Posting.) Content body pulled from source.*
- %campaign_name%
 - *Name of created campaign. (user defined)*
- %tag_title%
 - *Uses Yahoo Terms API to discover a tag based on Title's content.*
- %tag_postbody%
 - *Uses Yahoo Terms API to discover a tag based on Post Body's content.*
- %video_embed%
 - *Youtube Only. Object Embed code of video.*
- %video_thumbnail%
 - *Youtube Only . SRC URL of thumbnail associated with video.*
- %video_description%
 - *Youtube only . Text description associated with video. Please not that due to Youtube API Nature, no live links are included within the description.*
- %amazon_price%
 - *Amazon only. Price of item if available. "price n/a" will appear in stead if no price is available.*
- %amazon_small_image_url%
 - *Amazon only. URL returned Smallest available rendition of main product image.*
- %amazon_medium_image_url%
 - *Amazon only. URL returned. Medium sized rendition of product image.*
- %amazon_large_image_url%
 - *Amazon only. URL returned. Largest available image size of product image.*
- %amazon_product_description%
 - *Amazon only. Combination of Product and Amazon reviews.*
- %amazon_product_features%
 - *Amazon only. Feature list combined into a ul/li html format.*
- %amazon_brand_name%
 - *Amazon only. Attempts to find the brand name of product and return it.*
- %amazon_model%
 - *Amazon only. Attempt to find a model number of the product and return it.*
- %amazon_customer_review_content_1%

- *Amazon only. First customer review if available.*
- *%amazon_customer_review_content_2%*
 - *Amazon only. Second customer review if available.*
- *%amazon_customer_review_content_3%*
 - *Amazon only. Third customer review if available.*
- *%amazon_customer_review_name_1%*
 - *Amazon only. Name details of first customer review.*
- *%amazon_customer_review_name_2%*
 - *Amazon only. Name details of second customer review.*
- *%amazon_customer_review_name_3%*
 - *Amazon only. Name details of third customer review.*
- *%amazon_buyitnow_button%*
 - *Amazon only. URL to default 'buy it now' button.*

This button is located ../../(back one directory)/images/btn_a_view_item.gif