## b:attr

The **b:attr** tag, or the *attribute* tag, is a tag for simplifying adding an attribute to the parent element, especially conditionally.

### Attributes

**name** - Name of the attribute.
**value** - Value of the attribute.
**cond** - Boolean expression for whether to append the attribute

### Example usage

```
<div>
  <b:attr name='data-foo'
          value='data:blog.blogId'
          cond='data:view.isHomepage' />
</div>
```

On homepages, this would produce

```
<div data-foo='123456789012'>
</div>
```

On other pages, this would produce

```
<div>
</div>
```

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

## b:class

The **b:class** tag, or the *class* tag, is a tag for simplifying adding a class to the parent element, especially conditionally.

### Attributes

**name** - Name of the class.
**cond** - Boolean expression for whether to add the class.

### Example usage

```
<div class='foo'>
  <b:class name='bar' cond='data:view.isHomepage' />
</div>
```

On homepages, this would produce

```
<div class='foo bar'>
</div>
```

On other pages, this would produce

```
<div class='foo'>
</div>
```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

## b:comment

The **b:comment** tag, or the *comment* tag, is a tag for adding a non-rendered comment to a template, or for adding a comment with dynamic content.

### Attributes

**render** - Whether to render the comment into the HTML (default: false)

### Example usage

**Non-rendered comments**

```
<b:comment>
This is an explanation.
It will not be in the rendered HTML.
</b:comment>
```

**Rendered comments**

```
<b:comment render='true'>
  This is<b:if cond='not data:view.isPost'> not</b:if> a post page.
</b:comment>
```

On a post, this would produce

```
<!-- This is a post page -->
```

On other pages, this would produce

```
<!-- This is not a post page -->
```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

## b:eval

The **b:eval** tag, or the *evaluate* tag, is a tag for evaluating the value of an expression. It differs from the **data** tag in that data tags emit a single value, whereas b:eval tags can combine or modify data values using an expression.

### Attributes

**expr** - Expression to evaluate.

### Example usage

**Resize an image with an operator call**
The following example emits the post title and body for each of the posts in the data:posts dictionary.

```
<style>
  .post-thumbnail {
    background-image: url(
        <b:eval expr='resizeImage(data:post.featuredImage, 385).cssEscaped' />);
  }
</style>
```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

## b:if

The **b:if** tag, or the *if* tag, is a tag for specifying a conditional output.

### Attributes

**cond** - Boolean expression for whether the b:if tag's descendants should be rendered.

b:if tags are combined with nested self-closing **b:elseif** and **b:else** tags.

## b:elseif

b:elseif tags are self-closing tags used within a b:if tag to separate alternative output for another conditional expression case.

### Attributes

**cond** - Boolean expression for whether the b:elseif tag's subsequent sibling nodes should be rendered.

## b:else

b:else tags are self-closing tags used within a b:if tag to separate the default output when the parent b:if tag's condition is false, and none of the other b:elseif conditions are true either.

### Example usage

The following example emits a different comment for some possible views.

```
<b:if cond='data:view.isSearch'>
  <!-- Search page -->
<b:elseif value='data:view.isMultipleItems' />
  <!-- Other stream page -->
<b:else />
  <!-- Not a multiple item page. -->
</b:if>
```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

## b:loop

The **b:loop** tag, or the *loop* tag, is a tag for iterating across a range of values and repeating the descendants of the b:loop tag for each iteration.

### Attributes

**values** - Values to iterate across. This can be an expression for a data array, e.g. data:posts, or it can be a specific number range, e.g. '1 to 12'.
**var** - Name of the variable which will hold the current value.
**index** - Name of the variable which will hold the index of the current iteration.
**reverse** - Boolean value for whether to iterate the values backward.

## Example usage

### A defined array

The following example emits the post title and body for each of the posts in the data:posts dictionary.

```
<b:loop var='post' values='data:posts' index='i'>
  <div>
    <h3>
      <b:eval expr='data:i + 1' />. <data:post.title />
    </h3>
    <div>
      <data:post.body />
    </div>
  </div>
</b:loop>
```

Example output might be:

```
<div>
  <h3>
    1. Second post ever
  </h3>
  <div>
    This is the second post.
  </div>
</div>
<div>
  <h3>
    2. Hello world
  </h3>
  <div>
    This is my first post, ever!
  </div>
</div>
```

### An arbitrary iteration

In another example, we produce a list of labels searches, by iterating across an inline-defined set of labels and producing links for searching for that label.

```
<ul>
  <b:loop var='label'
          values='["cat", "dog", "mouse", "hyena"]'>
    <li b:whitespace='remove'>
      <a href='data:blog.searchUrl params { label: data:label }' b:whitespace='remove'>
        <data:label />
      </a>
    </li>
  </b:loop>
</ul>
```

The resulting output, for myblog.blogspot.com, would be the following list.

```
<ul>
  <li><a href='http://myblog.blogspot.com/search?label=cat'>cat</a>
  <li><a href='http://myblog.blogspot.com/search?label=dog'>dog</a>
  <li><a href='http://myblog.blogspot.com/search?label=mouse'>mouse</a>
  <li><a href='http://myblog.blogspot.com/search?label=hyena'>hyena</a>
</ul>
```

### Tracking the index

In this example, we flag the last item in the list with the class 'last'.

```
<b:loop var='post' values='data:posts' index='i'>
  <div>
    <b:class name='last' cond='data:i == data:posts.size + 1' />
    <h3><data:post.title /></h3>
    <div>
      <data:post.body />
    </div>
  </div>
</b:loop>
```

Note that the `index` attribute is the zero-based offset into the loop.

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

## b:message

The **b:message** tag, or the *message* tag, is a tag for rendering a translated message.

### Attributes

**name** - Name of the message to render.

While the output of a b:message tag can usually be achieved with an equivalent data tag, as demonstrated below, the b:message tag also supports parameterized messages (messages which have one or more variables in their content).

Messages which are translated are available in the global messages dictionary.

### b:param

b:message tags are combined with nested, self-closing b:param tags to specify the values of parameters when needed.

### Example usage

The following 2 tags would produce the same output, since the message *postAComment* has no parameters.

```
<b:message name='messages.postAComment' />
<data:messages.postAComment />
```

For an English blog, this produces 'Post a comment', while for a French blog, it produces 'Enregistrer un commentaire'

#### Parameterized messages

```
<b:message name='messages.numberOfComments'>
  <b:param name='numComments' expr:value='data:post.numberOfComments' />
</b:message>
```

On a French blog, for a post with 2 comments, this would produce '2 commentaires', while on an English blog it would produce '2 comments'.

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
///////////////////////////////////////////////////////////////

## b:switch

The **b:switch** tag, or the *switch* tag, is a tag for specifying the result for multiple possible values of a variable.

### Attributes

**var** - Variable value on which to switch.

b:switch tags are combined with nested self-closing **b:case** and **b:default** tags.

## b:case

b:case tags are used within a b:switch tag to separate the output for each value case of the variable in the parent b:switch tag.

### Attributes

**value** - Value which the variable has in order for this case to be rendered.

## b:default

b:default tags define the default output when none of the other b:case values are the value of the variable in the parent b:switch tag.

### Example usage

The following example emits a different greeting for each known language, defaulting to 'Hello world'.

```
<b:switch var='data:blog.locale.language'>
  <b:case value='fr' />
    <b:comment>French</b:comment>
    Bonjour monde
  <b:case value='it' />
    <b:comment>Italian</b:comment>
    Ciao mondo
  <b:case value='de' />
    <b:comment>German</b:comment>
    Hallo Welt
  <b:default />
    <b:comment>Default - English</b:comment>
    Hello world
</b:switch>
```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
///////////////////////////////////////////////////////////////

## b:tag

The **b:tag** tag is a tag for creating a tag or element which is conditionally the container for its children, or contains a dynamic tag name.

Any child elements of the **b:tag** element will be evaluated and rendered, regardless of whether the parent element is created. If the parent element is not created, the children will be appended to the **b:tag** element's *parent* element instead.

### Attributes

**name** - Name of the element that will be created, e.g. "div" or "span".
**cond** - Expression which, when false, causes the container element to be omitted.

**Note:** Attributes which are present on the **b:tag** element will be copied onto the resulting/created element, if created, except the attributes used by the **b:tag** element (*name* and *cond*).

## Example usage

**Conditional tag**
The following example emits an anchor tag around the blog title, but only if the current view is not the homepage.

```
<b:tag name='a'
       cond='data:view.url != data:blog.homepageUrl'
       expr:href='data:blog.homepageUrl'>
  <data:blog.title />
</b:tag>
```

**Dynamic tag names**
The following example emits different tags depending on whether the current view is mobile.

```
<b:tag expr:name='data:view.isMobile ? "select" : "ul"'
       cond='data:view.url != data:blog.homepageUrl'
       expr:href='data:blog.homepageUrl'>
  <b:loop var='item' values='data:items'>
    <b:tag expr:name='data:view.isMobile ? "option" : "li"'>
      <data:item.name />
    </b:tag>
  </b:loop>
</b:tag>
```

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

## b:with

The **b:with** tag, or the *variable* tag, is a tag for calculating a variable value before it needs to be used, whether it is being set up to be available for nested b:include tag calls, or is simply a short-name for an ugly expression we don't want to use inline.

## Attributes

**var** - Name of the variable.
**value** - Expression for the value the variable will take.

## Example usage

The following example overrides the Blog gadget's **main** includable to filter the lists of posts to exclude any posts labelled *hide-me*, by setting the data:posts variable's array to a filtered version of itself.

```
<widget id='Blog1' type='Blog'>
  <b:includable id='main'>
    <b:with var='posts'
            value='data:posts filter
                   (p => p.labels none
                        (l => l.name == "hide-me"))'>
      <b:include name='super.main' />
    </b:with>
  </b:includable>
</b:widget>
```

When the Blog gadget loads, any posts which are not labelled *hide-me* will still show, but all the posts which are labelled *hide-me* will not be rendered.

The next example re-uses a calculated title as the alt-text for an image as well as the content for an h1 tag.

```
<b:with var='title'
        value='data:post.title ?: data:view.title ?: "Default title"'>
  <a href='/foo'>
    <img src='/bar' expr:alt='data:title.escaped' />
    <h1><data:title></h1>
  </a>
  <b:include name='super.main' />
</b:with>
```

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
///////////////////////////////////////////////////////////////

## string

Strings are quote-wrapped sequences of characters.
**length**: Provides the number of characters in the string.
**size**: Same as length.
**escaped**: Provides the HTML escaped equivalent of the string. This is useful when the string may represent user input, to avoid HTML injection attacks
**jsEscaped**: Provides the JS escaped equivalent of the string. This is useful when the string may represent user input, to avoid JS errors.
**jsonEscaped**: Provides the JSON escaped equivalent of the string. This is useful when the string may represent user input, to avoid JSON errors.
**cssEscaped**: Provides the CSS escaped equivalent of the string.

### Operators

#### string1 == string2

(Also supported with the syntax eq)
Returns true if the first string is equal to the second (case-sensitive).

#### string1 `contains` string2

(Also supported with the syntax `in`)
Returns true if the first string contains the second (case-sensitive).

#### snippet(string, options)

Produces a short snippet from an HTML string.
**options**: Object specifying the snippeting options, which are:

- **links**: boolean for whether to preserve anchors/links in the snippet. Defaults to true.
- **linebreaks**: boolean for whether to preserve linebreaks (
  `tags`) in the snippet. Defaults to true.
- **ellipsis**: boolean for whether to append an ellipsis (…) to the end of the snippet. Defaults to true.
- **length**: Number specifying the maximum length of the snippet.

**Example usage**

```
<b:eval expr='snippet(data:post.body, { length: 150, links: false, linebreaks: f
```

## object

Objects are containers, which have named values of many items of data.

Objects can be constructed inline, using a squiggly-brace syntax similar to JSON.
**Example**: An inline object being passed to the string snippet operator.

```
<b:eval expr='snippet(data:post.body, { length: 150, links: false, linebreaks: f
```

## url

URL data is a special type of string data which has some extra metadata values for converting the URL into other forms.

**canonical**: Provides the canonical equivalent of the given URL. A canonical version of a URL ignores the country domain, and is used to uniquely identify this page, for example when sharing a URL to facebook.
**http**: Ensures that the scheme of the given URL is http.
**https**: Ensures that the scheme of the given URL is https.

### Operators

The following operators apply to the `url` data type. Note that these operators can also be used on an arbitrary string, provided that string represents a URL.

#### path(url, path)

Replaces the path component of the given url with the given path.
**url**: Url to replace the path of, e.g. data:blog.url. **path**: New path for the url.

#### params(url, params)

Replaces the query parameters of the given url with the given parameters.
**url**: Url to replace the path of, e.g. data:blog.url.
**params**: Object containing the new parameters for the url. e.g. `{ foo: "bar" }` would produce `?foo=bar`.

#### appendParams(url, params)

Appends (instead of replacing) extra query parameters for the given url.
**url**: Url to replace the path of, e.g. data:blog.url.
**params**: Object containing the extra parameters for the url. e.g. `{ foo: "bar" }` would add `?foo=bar`.

#### fragment(url, fragment)

Appends or replaces the given fragment for the given url (e.g. foo in my.url.com/page.html#foo)
**url**: Url to replace the path of, e.g. data:blog.url.
**fragment**: New fragment for the url. e.g. `"url"` would produce `#foo=bar`.

## image

Image objects are wrappers for URLs which allow the image to be resized.

**isResizable**: true if the image can be resized using the resizeImage operator.
**isYoutube**: true if the image is a youtube thumbnail.
**youtubeMaxResDefaultUrl**: URL of the maxresdefault.jpg youtube thumbnail, which exists for HD youtube videos.
**width**: Width of the image, if specified explicitly, or null.
**height**: Height of the image, if specified explicitly, or null.

### Operators

#### resizeImage(image, width [, ratio])

Resizes the given image url to the given width (and crops it to the given ratio, if present).
*Note*: If the image is smaller than the resize width, the image will *not* be stretched/upscaled.

**image**: Image, or url of the image, to be resized. e.g. data:post.featuredImage.
**width**: New width constraint for the image.
**ratio** (optional): New width:height ratio for the image to be cropped to.

#### sourceSet(image, widths [, ratio])

Produces a srcset attribute for the given image, resizing the given image to each of the given widths (and cropping them to the given ratio, if present).
*Note*: If the image is smaller than the resize width, the image will *not* be stretched/upscaled. This can be problematic when the requested size is larger than the image as the browser will scale it incorrectly.
**image**: Image, or url of the image, to produce a srcset for. e.g. data:post.featuredImage.
**width**: Array of new width constraints for the image.
**ratio** (optional): New width:height ratio for the image to be cropped to.

## array

A collection of data values.
**length**: Provides the number of entries in the array.
**size**: Same as length.
**last**: The last item in the array.
**first**: The first item in the array.
**empty**: Whether the array has no items.
**notEmpty**: Whether the array has at least one items.
**any**: same as notEmpty.

### Operators

The following operators apply to the *array* data type.

#### array `contains` item

Returns a boolean which is true if the given array contains the given item.
**Example usage**:

```
<b:if cond='[123, 456] contains data:post.id' /> [Special markup] </b:if>
```

#### array `take` number

(Also supported with the syntax `limit`)
Returns an array limited to `number` of the given array's items.

#### array `skip` number

(Also supported with the syntax `offset`)
Returns an array of the remaining items after skipping `number` of the given array's items.

#### lambda operators

Lambda operators are operators which apply a given function or condition to each item in an array. They are applied using the syntax

```
array [lambda operation] (x => [function involving x])
```

Where:

- x can be any variable name you like.
- `lambda operation` is an operation from the list below.
- `function involving x` is any expression involving x, where x is a variable representing an item in the array.

### Lambda Operations

- **any**: Returns a boolean for whether any item in the array returns true for the given function.
- **all**: Returns a boolean for whether all the items in the array returns true for the given function.
- **none**: Returns a boolean for whether none of the items in the array returns true for the given function.
- **count**: Returns a number for the count of the items in the array that return true for the given function.
- **map**: Returns an array of the results for applying the given function to each of the array items.
- **filter**: Returns an array of the items which return true for the given function.

## date

Dates are timestamp objects.
**year**: Number representing the year component of the date.
**month**: Number representing the month of the year (1 for Jan, 2 for Feb, etc).
**day**: Number representing the day of the month.
**iso8601**: ISO8601 standard formatting for the date.

### Operators

### format(date, format)

Formats the given date to the given format string, using the blog's selected language.
**date**: Date to be formatted, e.g. data:post.date
**format**: ICU format string for the date to be formatted to, in the blog's language. e.g. "MMM dd". **Example**: A custom Post timestamp byline (full example).

```
<b:eval expr='data:post.date format "MMM dd" ' />
```

## message

Messages are strings that are translated to the blog's locale.
Some messages have parameters, such as a the author name, that need to be passed in using the tag.
Example message with parameters:

```
<b:message name='messages.byAuthor'>
<b:param name='authorName' expr:value='data:post.author.name' />
</b:message>
```

**reader**: Translates the message into the readers locale instead of the blog's locale.

## locale

Locales are identifiers for a language and variant used by a blog.

**name**: Display name of the locale, e.g. "English".
**language**: Code representing the language, e.g. "en" for English.
**country**: The country of the language, e.g. "AU" for Australia.
**variant**: The variant of the language, e.g. "TH" for Thai digits when using locale th-TH-TH.
**script**: The script of the language, e.g. "Arab" for Arabic.
**languageDirection**: Either "LTR" (for left-to-right e.g. English), or "RTL" (for right-to-left e.g. Arabic).

Note that the blog's locale can be overridden using the **hl** parameter, e.g. ?hl=en will force English.
This can be leveraged to try and support multiple languages (Example instructions).

## data:view

| | | |
|---|---|---|
| view. archive | object | Fields relating to the archive for the current view. |
| view. description | string | Description of the current view. |
| view. featuredImage | image | Featured image for the current view. |
| view. isArchive | boolean | Whether the current view is an Archive feed. |
| view. isError | boolean | True if the current view is an error page. |
| view. isHomepage | boolean | True if the current view is the homepage of the blog. |
| view. isLayoutMode | boolean | True if the view is the Layout mode for rearranging widgets in the template. |
| view. isMultipleItems | boolean | True if the current view has multiple posts, e.g. the homepage or a search page. |
| view. isPage | boolean | True if the current view is for a single page item. |
| view. isPage | boolean | True if the current view is a page item. |
| view. isPost | boolean | True if the current view is for a single post. |
| view. isPost | boolean | True if the current view is a post page. |
| view. isPreview | boolean | Whether the current view is a preview. |
| view. isPreview | boolean | True if the current view is a preview. |
| view. isSearch | boolean | True if the current view is for a query or label filter search. |
| view. isSingleItem | boolean | True if the current view is a single post or page. |
| view. pageId | number | Id of the page in the current view, if the view is a page item. |
| view. postId | number | Id of the post for the current view, if the current view is a post. |
| view. search | object | Data related to the search parameters of the current view. |
| view. title | string | Title of the current view. |
| view. type | string | Type of the current view, either "item" or "feed". |
| view. url | url | The url of the current view. |
| view.archive. day | number | Day of the current archive feed view, e.g. 15. |
| view.archive. month | number | Month of the current archive feed view, e.g. 4 |
| view.archive. rangeMessage | string | Message describing the current archive view's date range. |
| view.archive. year | number | Year of the current archive view, e.g. 2016 |
| view.search. label | string | Label filter for the current view's search. |
| view.search. query | string | Query filter for the current view's search. |
| view.search. resultsMessage | string | Localized message about the search results being shown. |
| view.search.resultsMessageHtml | string | HTML version of resultsMessage . |

## data:blog

| | | |
|---|---|---|
| blog. adultContent | boolean | Whether the blog is contains adult content. |
| blog. blogId | string | Id of the blog. |
| blog. homepageUrl | url | Homepage url of the Blog. |
| blog. isPrivate | boolean | True if the blog is configured to be private, with a reader whitelist. |
| blog. languageDirection | string | The direction of the language used by the blog, either RTL or LTR. |
| blog. locale | locale | Locale code of the blog, representing the language that the blog will render its messages in. |
| blog. searchUrl | url | Url for executing a search for the blog. |

| | | |
|---|---|---|
| blog. title | string | Title of the Blog. |

## data:skin

| | | |
|---|---|---|
| skin. vars | array | Variables of the skin rendered for the current view, if the current view is a post. |

## data:messages

| | | |
|---|---|---|
| messages. adsGoHere | message | "Ads go here", translated into the language used by the blog. |
| messages. archive | message | "Archive", translated into the language used by the blog. |
| messages. comments | message | "Comments" translated into the language used by the blog. |
| messages. edit | message | "Edit" translated into the language used by the blog. |
| messages. emailAddress | message | "Email address" translated into the language used by the blog. |
| messages.getEmailNotifications | message | "Get email notifications" translated into the language used by the blog. |
| messages. gotIt | message | "Got it" translated into the language used by the blog. |
| messages. hidden | message | "Hidden" translated into the language used by the blog. |
| messages. home | message | "Home" translated into the language used by the blog. |
| messages. image | message | "Image" translated into the language used by the blog. |
| messages.joinTheConversation | message | "Join the conversation" translated into the language used by the blog. |
| messages. keepReading | message | "Keep reading" translated into the language used by the blog. |
| messages. labels | message | "Labels" translated into the language used by the blog. |
| messages. latestPosts | message | "Latest posts" translated into the language used by the blog. |
| messages. learnMore | message | "Learn more" translated into the language used by the blog. |
| messages. loadMorePosts | message | "Load more posts" translated into the language used by the blog. |
| messages. myFavoriteSites | message | "My favorite sites" translated into the language used by the blog. |
| messages. myPhoto | message | "My photo" translated into the language used by the blog. |
| messages. newer | message | "Newer" translated into the language used by the blog. |
| messages. newerPosts | message | "Newer posts" translated into the language used by the blog. |
| messages. newest | message | "Newest" translated into the language used by the blog. |
| messages. noResultsFound | message | "No results found" translated into the language used by the blog. |
| messages. noTitle | message | "No title" translated into the language used by the blog. |

| | | |
|---|---|---|
| messages.numberOfComments | message | "[Number] comments", where number is a parameter passed in, translated into the language used by the blog. |

## data:widgets

| | | |
|---|---|---|
| widgets. AdSense | array | An array of all of the AdSense widgets which are visible in the current page. |
| widgets. Attribution | array | An array of all of the Attribution widgets which are visible in the current page. |
| widgets. Blog | array | An array of all of the Blog widgets which are visible in the current page. |
| widgets. BlogArchive | array | An array of all of the BlogArchive widgets which are visible in the current page. |
| widgets. BlogList | array | An array of all of the BlogList widgets which are visible in the current page. |
| widgets. BlogSearch | array | An array of all of the BlogSearch widgets which are visible in the current page. |
| widgets. BloggerButton | array | An array of all of the BloggerButton widgets which are visible in the current page. |
| widgets. ContactForm | array | An array of all of the ContactForm widgets which are visible in the current page. |
| widgets. CustomSearch | array | An array of all of the CustomSearch widgets which are visible in the current page. |
| widgets. FeaturedPost | array | An array of all of the FeaturedPost widgets which are visible in the current page. |
| widgets. Feed | array | An array of all of the Feed widgets which are visible in the current page. |
| widgets. FollowByEmail | array | An array of all of the FollowByEmail widgets which are visible in the current page. |
| widgets. Followers | array | An array of all of the Followers widgets which are visible in the current page. |
| widgets. Gadget | array | An array of all of the Gadget (third-party) widgets which are visible in the current page. |
| widgets. HTML | array | An array of all of the HTML widgets which are visible in the current page. |
| widgets. Header | array | An array of all of the Header widgets which are visible in the current page. |
| widgets. Image | array | An array of all of the Image widgets which are visible in the current page. |
| widgets. Label | array | An array of all of the Label widgets which are visible in the current page. |
| widgets. LinkList | array | An array of all of the LinkList widgets which are visible in the current page. |
| widgets. Navbar | array | An array of all of the Navbar widgets which are visible in the current page. |
| widgets. PageList | array | An array of all of the PageList widgets which are visible in the current page. |
| widgets. PlusBadge | array | An array of all of the PlusBadge widgets which are visible in the current page. |
| widgets. PlusFollowers | array | An array of all of the PlusFollowers widgets which are visible in the current page. |
| widgets. PlusOne | array | An array of all of the PlusOne widgets which are visible in the current page. |
| widgets. Poll | array | An array of all of the Poll widgets which are visible in the current page. |
| widgets. PopularPosts | array | An array of all of the PopularPosts widgets which are visible in the current page. |

| | | |
|---|---|---|
| widgets. Profile | array | An array of all of the Profile widgets which are visible in the current page. |
| widgets. ReportAbuse | array | An array of all of the ReportAbuse widgets which are visible in the current page. |
| widgets. Slideshow | array | An array of all of the Slideshow widgets which are visible in the current page. |
| widgets. Stats | array | An array of all of the Stats widgets which are visible in the current page. |
| widgets. Subscribe | array | An array of all of the Subscribe widgets which are visible in the current page. |
| widgets. Text | array | An array of all of the Text widgets which are visible in the current page. |
| widgets. TextList | array | An array of all of the TextList widgets which are visible in the current page. |
| widgets. Translate | array | An array of all of the Translate widgets which are visible in the current page. |

## Blog Posts gadget

| | | |
|---|---|---|
| blogComment | message | Message for commenting action on the blog. |
| description | string | The description for the blog. |
| feedLinks | array | An array of links to the blog's feeds. |
| feedLinks[i]. feedType | string | The feed's type, e.g. "Atom". |
| feedLinks[i]. mimeType | string | Mime type of the feed, e.g. "application/atom+xml". |
| feedLinks[i]. name | string | The name of the feed, e.g. "Posts". |
| feedLinks[i]. url | string | URL of the feed link. |
| footerBylines | array | An array of the 3 footer bylines, shown after a post. |
| footerBylines[i]. items | array | The byline items in the byline region. |
| footerBylines[i].items[i].label | string | Label of the byline, shown before the byline content, e.g. "Posted by" for author. |
| footerBylines[i].items[i].name | string | Name of the byline. |
| footerBylines[i].regionName | string | Name of the footer byline region, e.g. "footer-2". There are 3 byline footer regions, but only one byline header region. |
| id | string | The id of the blog. Equivalent to data:blog.blogId. |
| messages | object | Object containing Blog Posts messages. |
| numPosts | number | Number of posts in the posts array. (Deprecated - can be found using the posts.size equivalent). |
| olderPageUrl | url | URL of the next page of (older) posts, used for pagination. |
| posts | array | An array of the Blog's posts to be displayed in the current view. |
| posts[i]. allowComments | boolean | Whether the post has been configured to allow comments. |
| posts[i].allowNewComments | boolean | Whether new comments are allowed to be made on the post. |
| posts[i]. author | object | Object containing fields relating to the author of the post. |
| posts[i].author. aboutMe | string | Contents of the 'About me' blurb on the post author's profile. |
| posts[i].author. name | string | Name of the author. |
| posts[i].author. profileUrl | string | URL of the profile of the post author. |
| posts[i]. body | string | The HTML content of the post. |

| | | |
|---|---|---|
| posts[i].commentFormIframeSrc | string | URL for the source of the comment iframe. |
| posts[i]. commentsUrl | url | URL linking to the comments on the post. |
| posts[i]. date | date | Date of the post. |
| posts[i]. emailPostUrl | string | URL for the *Email post* byline's share-by-email action. |
| posts[i].embedCommentForm | boolean | Whether the user has configured comments to be embedded in the page. |
| posts[i]. featuredImage | image | The featured image for the post. |
| posts[i]. hasJumpLink | boolean | Whether the post has a <!--more--> *jump-break* in it. |
| posts[i]. id | string | The ID of the post. |
| posts[i]. lastUpdated | date | Date that the post was last updated. |
| posts[i].numberOfComments | number | The number of comments that have been made on the post. |
| posts[i]. reactionsUrl | url | URL for the iframe of the reactions byline of the post. |
| posts[i]. title | string | Title of the post. |
| posts[i]. url | url | URL of the post. |
| title | string | Title of the blog. Equivalent to the global data:blog.title field. |

## Header gadget

| | | |
|---|---|---|
| backgroundPositionStyleStr | string | Position style ("left" or "right"), depending on whether the blog is RTL. |
| description | string | Description of the blog. |
| image | image | Background image for the Header gadget, if there is one. |
| imagePlacement | string | Placement of the background image, if present. One of BEHIND, REPLACE, BEFORE_DESCRIPTION. |
| title | string | Title of the blog. |

## Popular Posts gadget

| | | |
|---|---|---|
| postDisplay | object | Object containing the display settings for a post. |
| postDisplay.showFeaturedImage | boolean | Whether to show the featured image for the post. |
| postDisplay. showSnippet | boolean | Whether to show a snippet of the post body. |
| postDisplay. showTitle | boolean | Whether to show the title of the post. |
| posts | array | Array of objects containing the popular posts. |
| posts[i]. body | string | HTML of the content of the post. |
| posts[i]. commentsUrl | url | URL of the comments of the post. |
| posts[i]. date | date | Date that the post was written. |
| posts[i]. featuredImage | image | Featured image for the post. |
| posts[i]. id | string | The ID of the post. |
| posts[i]. lastUpdated | date | Date that the post was last updated. |
| posts[i].numberOfComments | number | The number of comments on the post. |
| posts[i]. title | string | Title of the post. |
| posts[i]. url | url | URL of the post. |

## Blog Archive gadget

| | | |
|---|---|---|
| data | array | Array containing objects which span the archive ranges. |
| data[i]. name | string | Name of the archive range, e.g. "July 2017". |
| data[i]. post-count | number | Number of posts in the archive range. |
| data[i]. url | url | URL for the archive range. |
| style | string | Style that the archive is rendered in - either FLAT, HIERARCHY or MENU. |
| title | string | Title of the Blog Archive gadget. |

## Labels gadget

| | | |
|---|---|---|
| display | string | Display style for the Labels gadget - either "list" or "cloud". |
| labels | array | Array containing objects for each of the labels. |
| labels[i]. count | number | Number of posts with the label. |
| labels[i]. name | string | Name of the label. |
| labels[i]. url | url | URL for viewing posts with the label (label search). |
| showFreqNumbers | boolean | Whether to show the number of posts with the label. |
| title | string | Title of the Labels gadget. |

## Featured Post gadget

| | | |
|---|---|---|
| postDisplay | object | Object containing the post display settings for the Featured Post gadget. |
| postDisplay.showFeaturedImage | boolean | Whether to show the featured image of the post. |
| postDisplay. showSnippet | boolean | Whether to show a snippet of the post body. |
| postDisplay. showTitle | boolean | Whether to show the title of the post. |
| posts | array | Array containing objects for the posts. |
| posts[i].author.featuredImage | image | Featured image for the post. |
| posts[i]. body | string | HTML content of the post's body. |
| posts[i]. commentsUrl | url | URL for the comments on the post. |
| posts[i]. date | date | Date that the post was written. |
| posts[i]. hasJumpLink | boolean | Whether the post contains the <!--more--> jump-link text. |
| posts[i]. id | string | ID of the post. |
| posts[i]. lastUpdated | date | Date that the post was last updated. |
| posts[i].numberOfComments | number | Number of comments on the post. |
| posts[i]. title | string | Title of the post. |
| posts[i]. url | url | URL of the post. |

## Plus One gadget

| | | |
|---|---|---|
| annotation | string | Annotation of the Plus One gadget - either "bubble", "inline" or "none". |
| language | string | Language to display the Plus One gadget in. |
| size | string | Size of the Plus One gadget - either "small", "medium", "standard" or "tall". |

## Plus Followers gadget

| | | |
|---|---|---|
| height | number | Height of the Plus Followers display. |
| language | string | Language of the Plus Followers display. |
| profileUrl | string | URL of the Google+ profile, if the Plus Followers gadget is linked to one. |
| theme | string | Theme of the Plus Followers display - either DARK or LIGHT. |
| title | string | Title of the Plus Followers gadget. |
| widgetInstanceId | string | Instance ID of the Plus Followers gadget. |
| width | number | Width of the Plus Followers display. |

## Blogger Button gadget

| | | |
|---|---|---|
| source | string | Image source of the Blogger button to be shown. |

## Translate gadget

| | | |
|---|---|---|
| layout | string | Layout of the Translate gadget - either VERTICAL, HORIZONTAL or SIMPLE. |
| pageLanguage | string | Language of the page, e.g. "en" or "fr". Defaults to "auto" when no language is found. |
| title | string | Title of the Translate gadget. |

## Blog List gadget

| | | |
|---|---|---|
| items | array | Array of objects representing each of the blogs. |
| items[i]. blogIconUrl | string | Source URL for the icon of the blog. |
| items[i]. blogTitle | string | Title of the blog. |
| items[i]. blogUrl | string | The blog's homepage URL. |
| items[i]. itemSnippet | string | Snippet of the post's content for the first post of the blog. |
| items[i]. itemThumbnail | object | Object containing details for the thumbnail of the first post for the blog. |
| items[i].itemThumbnail.height | number | Height of the thumbnail of the first post for the blog. |
| items[i].itemThumbnail. url | string | URL of the thumbnail of the first post for the blog. |
| items[i].itemThumbnail. width | number | Width of the thumbnail of the first post for the blog. |
| items[i]. itemTitle | string | Title of the first post for the blog. |
| items[i]. itemUrl | string | URL of the first post for the blog. |
| items[i].timePeriodSinceLastUpdate | string | Translated message representing the amount of time since the most recent update to the blog. |
| numItemsToShow | number | The number of items to show in the Blog List. |
| showIcon | boolean | Whether to show each blog's icon. |
| showItemSnippet | boolean | Whether to show a snippet of the first item |

| | | |
|---|---|---|
| | | for each blog. |
| showItemThumbnail | boolean | Whether to show a thumbnail with the first item of each blog. |
| showItemTitle | boolean | Whether to show the title of the first post for each blog. |
| sortType | string | Type of sorting used by the Blog List gadget - either ALPHABETICAL or LAST_UPDATE_DESCENDING. |
| totalItems | number | Total number of items in the *items* array. |

## Contact gadget

| | | |
|---|---|---|
| blogId | string | ID of the blog. |
| submitUrl | string | URL for submitting the Contact form. |
| title | string | Title of the Contact Form gadget. |

**CONDITION - EST VRAI**

```
<b:if cond='data:view.isLayoutMode'>

</b:if>
```

**CONDITION - N'EST PAS VRAI**

```
<b:if cond='not data:view.isLayoutMode'>

</b:if>
```

**CONDITION - EST VRAI, SINON EST FAUX**

```
<b:if cond='data:view.isLayoutMode'>

  <b:else/>

</b:if>
```

```
<b:if cond='data:view.isSearch'>
  <!-- Search page -->
<b:elseif value='data:view.isMultipleItems' />
  <!-- Other stream page -->
<b:else />
  <!-- Not a multiple item page. -->
</b:if>
```

```
<b:if cond='condition_expression'>
    <!-- content when condition is satisfied -->
</b:if>
```

## List of Conditional tags for page types

## Archive page

```
<b:if cond='data:blog.pageType == "archive"'>
<!--archive_Page-->
</b:if>
```

## Error Page (404)

```
<b:if cond='data:blog.pageType == "error_page"'>
<!-- all error pages-->
</b:if>
```

## Index Page

```
<b:if cond='data:blog.pageType == "index"'>
<!-- all index pages -->
</b:if>
```

## Homepage

```
<b:if cond='data:blog.url == data:blog.homepageUrl'>
<!-- only homepage -->
</b:if>
```

## Item (post) pages

```
<b:if cond='data:blog.pageType == "item"'>
<!-- all item pages -->
</b:if>
```

## Specific Post by URL

```
<b:if cond='data:blog.url == data:blog.canonicalHomepageUrl + "2014/08/foo.html"'>
<!-- a item page from august 2014 with post-title 'foo'-->
</b:if>
```

## Label page

```
<b:if cond='data:blog.searchLabel'>
<!-- all label pages -->
</b:if>
```

## Specific Label Page

```
<b:if cond='data:blog.searchLabel == "foo"'>
<!-- for label 'foo' -->
</b:if>
```

## Search page

```
<b:if cond='data:blog.searchQuery'>
<!-- all search pages -->
</b:if>
```

## Specific Search Query page

```
<b:if cond='data:blog.searchQuery == "foo"'>
<!-- for query 'foo' -->
</b:if>
```

## Static page

```
<b:if cond='data:blog.pageType == "static_page"'>
<!-- all static pages -->
</b:if>
```

## Specific Static page by URL

```
<b:if cond='data:blog.url == data:blog.canonicalHomepageUrl + "p/foo.html"'>
<!-- a specific static page with name 'foo' -->
</b:if>
```

## AND/OR/NOT

**AND**

```
<b:if cond='data:blog.pageType == "index"'>
  <b:if cond='data:blog.searchQuery'>
    <!--search_page AND index_page-->
  </b:if>
</b:if>
```

**OR**

```
<b:if cond='data:blog.url == data:blog.canonicalHomepageUrl + "p/foo.html"'>
  <!-- static_site foo OR static_site bar -->
      <b:else/>
<b:if cond='data:blog.url == data:blog.canonicalHomepageUrl + "p/bar.htm"'>
  <!-- static_site foo OR static_site bar -->
   </b:if>
</b:if>
```

**NOT**

```
<b:if cond='data:blog.pageType != "item"'>
  <!-- all pages except item pages -->
</b:if>

<b:if cond='data:blog.url != data:blog.homepageUrl'>
  <!-- all pages but NOT homepage -->
</b:if>
```

## How to Use conditional tags

To apply a conditional tag to some content, simply put the content inside the opening `<b:if cond…>` and the closing `</b:if>` like:

```
<b:if cond='data:blog.pageType == "item"'>
    # THIS CONTENT WILL BE EXECUTED IF CONDITION IS TRUE
</b:if>
```

In the example above, the content will only appear on post pages.

The content can be a `div`, a `section`, a `style` tag or another `conditional tag` etc.

If you want to specify a alternate content (when the condition is false), you need to insert a `<b:else/>` tag followed by the content, like this:

```
<b:if cond='data:blog.pageType == "item"'>
    # THIS CONTENT WILL BE EXECUTED IF CONDITION IS TRUE
    # i.e. if current page is post (item) page
<b:else/>
    # THIS CONTENT WILL BE EXECUTED IF CONDITION IS FALSE
    # i.e. if not post page
</b:if>
```

The `<b:else/>` also works like an `OR` operator as explained above.

## 1. Archive Page

Blogger pages added by Archive widget that normally has "_archive.html" ending in its URL belong to Archive page type.

```
<b:if cond='data:blog.pageType == &quot;archive&quot;'>
...CODE...
</b:if>
```

## 2. Index Page

Blogger pages with a list of posts are normally called Index pages. Eg - Homepage and Search result page.

```
<b:if cond='data:blog.pageType == &quot;index&quot;'>
...CODE...
</b:if>
```

Example - https://amprandom.blogspot.in/search/label/BloggerPages

## 3. Item Page

Blogger posts created by clicking new post button are called Item pages.

```
<b:if cond='data:blog.pageType == &quot;item&quot;'>
...CODE...
</b:if>
```

Conditional Tag for First Blogger post alone

```
<b:if cond='data:post.isFirstPost'>
...CODE...
</b:if>
```

Example - https://amprandom.blogspot.in/2017/01/ban-jallikattu-ban-pongal-ban-tamil.html

## 4. Static Page

Blogger pages created by clicking new page button are called Static pages.

```
<b:if cond='data:blog.pageType == &quot;static_page&quot;'>
...CODE...
```

```
</b:if>
```

Example - Using one such conditional Tag, I've converted one of my Static Pages into Home Page.
**HomePage URL:** https://amprandom.blogspot.in/
**Static Page URL:** https://amprandom.blogspot.in/p/random-tyms.html

## 5. Error Page

Blogger pages can be redirected to Error pages when directed to non-existent URLs.

```
<b:if cond='data:blog.pageType == &quot;error_page&quot;'>
...CODE...
</b:if>
```

Example -
**Correct Page:** https://amprandom.blogspot.in/2017/01/ban-jallikattu-ban-pongal-ban-tamil.html

**Error Page:** https://amprandom.blogspot.in/20177/01/ban-jallikattu-ban-pongal-ban-tamil.html

## 6. Label Page

Blogger label-based pages contain similar or related posted articles.

```
<b:if cond='data:blog.searchLabel'>
...CODE for all Label Pages...
</b:if>
```

Specify Label Name in the conditional statement.

```
<b:if cond='data:blog.searchLabel== &quot;LABEL_NAME&quot;'>
...CODE for specific Label Page...
</b:if>
```

Example - Label Name: FunPages
https://amprandom.blogspot.in/search/label/FunPages

## 7. Search Page

Blogger pages that provides a list of posts that matches the given query terms.

```
<b:if cond='data:blog.searchQuery'>
...CODE for all Search Pages...
</b:if>
```

Specify Search Query in the conditional statement.

```
<b:if cond='data:blog.searchQuery == &quot;SEARCH_QUERY&quot;'>
...CODE for specific Search Page...
</b:if>
```

Example - Search Query: Indian
https://amprandom.blogspot.in/search?q=indian