

Notes on developing a Blogger Template.

The "Blog" widget appears to have a default "main" includable. The template I'm referencing only defined an includable called "post", but as far as I can tell the global default includable is always named "main". This indicates to me that there is a main already included for "Blog" widgets if you don't override it with one of your own, which I'm going to try to do.

Comments will be removed if they're inside of `<b:section />` nodes but outside of `<b:includable />`. However, the `<b:widget />`'s can *only* contain `<b:includable />`'s, so you can't put comments in them. Example:

```
<b:section class='header' id='header' maxwidgets='1' showaddelement='no'>
<!-- CORRECT -->
<b:widget id='Header1' locked='true' title='My Blog (Header)' type='Header'>
<!-- WRONG -->
<b:includable id='main'>
</b:includable>
</b:section>
```

Blogger automatically inserts stylesheet references (`<link rel='stylesheet' type='text/css' href='...' />`) in the `<head />` part of the document. These certainly must slow down the loading of the blog, especially on dial-up. I wonder if there's anyway to avoid having them loaded at all.

Single quotes (apostrophe characters) are used around all attribute values inside `<b:includable />` blocks, even if you coded them with double-quotes.

Similar to the above note, all attributes within tags are alphabetically sorted.

Extraneous whitespace (including carriage returns) is removed from `<b:includable />` sections when they're rendered, so it's okay to use heavy amounts of it for indentation and separation.

`<b:section />` and `<b:widget />` tags turn into `<div>`'s. The "id" attribute is kept the same, but the CSS "class" attribute becomes "*name* section" for `<b:section />` and "*widget Type*" for `<b:widget />`. Since this can't be overridden, I think it's best just to go with it and have the CSS utilize these classes rather than defining my own (less tag clutter).

`<p>` is unreliable as a way to add styling to *just the body* of a post, because a `<center>` tag will reset it on most browsers (e.g. end the paragraph tag, because `<center>` shouldn't actually appear within `<p>`). Anyway, I'm using it anyway simply as a delimiter for when CSS is not present; the post still looks okay. Post body styling will be present in the "post" class and overridden in underlying tags like `<h#>` (title), `<var>` (date), `<address>` (author/tags), `<dl>` (comments) etc.

Accessing a non-existent `<data:... />` item will result in a page error, but the page that comes back won't actually tell you what the problem was.

--

No `elseif` / `elif`, there is only "if" and "else". This means that trying to switch between more than two outcomes causes a lot of messy nesting.

Any attribute value can be built by concatenating data values using the "expr" prefix. For example:

```
<tag expr:attrname='data:post.foo + "literal" + data:post.bar' />
```

List subscripts, e.g. accessing an item using an explicit index, doesn't appear to be possible in Blogger conditions. For example, I'd like to be able to check the first label of a post like so:

```
<b:if cond='data:post.labels[0] == "example">  
  Do something specific when the first label of a post is "example".  
</b:if>
```

If you're looking to lock posts into a single category, however, it is possible to do this by checking the last label, like so:

```
<b:loop var='label' values='data:post.labels'><b:if  
  cond='data:label.isLast'>  
<b:if cond='data:label.name == "video"'>  
  [[cat:video]]  
</b:if>  
<b:if cond='data:label.name == "walk"'>  
  [[<a expr:href='data:label.url + ".jpg"'>cat:walk</a>]]  
</b:if>  
</b:if></b:loop>  
<b:if cond='data:post.labels == 0'>  
  [[cat:generic]]  
</b:if>
```

Blarg.

:EOF