

Genshi Template Basics Cheatsheet

Format:

Genshi templates uses XML attributes to conquer the world. Here is an example:

```
<html xmlns:py="http://genshi.edgewall.org/" xmlns="http://www.w3.org/1999/xhtml"
xmlns:xi="http://www.w3.org/2001/XInclude">
<head>
  <title py:content="'New Title'">this will be replaced</title>
</head><body>
  this is the body
</body></html>
```

Notice two things. First is the XML Namespaces (the **xmlns:** parts). Those are required to use Genshi Templates. Include them and don't ask questions. The other is the **py:content** in the title tag. **py:content** is our example Genshi XML attribute. To see what it does, look below. But notice that the string in double-quotes is also surrounded in single-quotes. This is because all values for Genshi must be valid Python expressions. In other words, everything inside quotes must be valid Python code. **'New Title'** is a Python string whereas just **New Title** would raise an error. Not everything must be in an attribute. Python commands inside `${command}` will also be evaluated.

Conditionals

- **py:if="condition"** – if *condition* evaluates to **true** then the element and its contents will be rendered
- **py:choose="a-value": py:when="b-value"** – like a C switch statement. *a-value* is matched to whichever child node matches with *b-value* specified. Example:

```
<div py:choose="isEnabled">
  <h1>This part will always render because it doesn't have a py:when directive.</h1>
  <div py:when="True">You are enabled</div>
  <div py:when="False">You are not enabled</div>
</div>
```

Looping

- **py:for="item in itemlist"** – loop over a list. In this case, use the 'item' variable to access the list item. Example:

```
<ul py:for="name in AgentNames">
  <li>${name}</li>
</ul>
```

Structure Manipulation

- **py:content=" 'Text' "** – set the innerXML equal to the content specified. These two example tags yield the same output:

```
<title py:content="'New Title'">this will be replaced</title>
<title>${'New Title'}</title>
```

- **py:replace=" 'Text' "** – replace the tag and its contents with the content specified. Example:

```
<div><span py:replace="'Some text'">this will be replaced</span></div>
```

Renders as:

```
<div>Some Text</div>
```

- **py:strip="True"** – Removes the top-level element including this tag. This is conditional based on the value specified. Use this to hold other directives that are necessary for processing but should not directly render markup. Example:

```
<div py:strip="True"><span>Some Text</span></div>
```

Renders as:

```
<span>Some Text</span>
```

- **py:attrs="{ 'class': 'center' }"** – takes a Python dictionary and sets additional attributes for that tag:

```
<div py:attrs="{ 'class': 'center' }">Some Text</div>
```

Renders as:

```
<div class="center">Some Text</div>
```

Other Directives

- Specify a variable: `<div py:with="x=11">${x}</div>`
- XML comment that will be rendered to output: `<!-- This is a comment -->`
- XML comment that will **not** be rendered to output: `<!--! This is also a comment -->`
- Including other templates: `<xi:include href="path/to/file.html" />`
- Python code inside a page (a.k.a. Python Server Pages): `<?python import os ?>`
- More info at: <http://genshi.edgewall.org/wiki/Documentation/xml-templates.html>