

Zope Page Templates

- Using Templates, Scripts, DTML, and SQL
- METAL In-depth
- Future of ZPT: Libraries
- Future of ZPT: TERSE

Using Templates, Scripts, DTML, and SQL

- DTML and SQL Methods were designed to work with DTML, using namespaces and implicit parameter passing.
- Scripts and Templates were designed to work with Python, using explicit parameter passing.

Calling with Python

Python expressions can always be used:

- DTML has 'expr' and dtml-let.
- ZPT has “python:” expression type.

```
<dtml-var expr="x()">
```

```
<dtml-let v="x()">
```

```
<dtml-var v>
```

```
</dtml-let>
```

```
tal:replace="python:x()"
```

Calling with Python

- Scripts must be passed arguments that match their parameter lists.
- Templates may be passed any arguments, which are accessed through the 'options' mapping.
- SQL Methods may be called in three different ways:
 - 1) With no arguments
 - 2) With a dictionary
 - 3) With keyword arguments

Calling DTML with Python

- DTML requires the REQUEST object and the context object as its first two arguments. Any number of keyword arguments may also be used, to add values to the DTML namespace.

Calling with DTML Magic

- Calling an object by name from DTML invokes namespace magic.
 - 1) Other DTML inherits the entire namespace.
 - 2) Scripts can bind the namespace to a variable.
 - 3) SQL looks for parameter values in the namespace.

Calling with a TALES Path

- DTML and SQL Methods, Scripts, and Templates named by Paths are called with no arguments.
- A DTML Method places the calling template's variables on its namespace.
- A SQL Method looks in the REQUEST for its argument values.

METAL In-depth

- METAL allows you to define and use macros.
- Macros can have slots.

METAL Macros

- A METAL Macro is a well-formed piece of an HTML or XML document.
- Macros can be expanded at design-time, unlike `tal:replace="structure"`.
- TAL statements don't know or care if they are in a Macro. To change the behavior of TAL in a Macro, define variables in an element surrounding the Macro.

METAL Slots

- Slots are replaceable sub-elements of Macros.
- If not replaced, the contents of a slot definition remain in place.

Macros within Macros

- A Macro can be defined even within another Macro's definition or in a filled slot.
- A Macro's definition can use other Macros.

Future of ZPT: Libraries

- Templates with complex formatting or data processing can require a lot of Python.
- Embedding logic in templates is bad.
- Lots of little helper Scripts is bad.
- Controlling a Template with a Script is not so bad, but somewhat awkward.

Libraries to the Rescue

- A Library is a collection of Python functions that is managed as a single object.
- Library functions can easily call each other, and can share persistent, private data.
- TALEX expressions can be used to automatically provide context variables for all functions in a Library.

Future of ZPT: TERSE

- TAL is great for template design, if you're intimately familiar with HTML.
- TAL is limited by XML syntax rules, and the structure of documents.
- Certain patterns recur often, and should be abstracted.

TERSE is like CSS

- A **Template Rule Set** is a list of selector/rule pairs.
- Use CSS-style selectors to match document elements.
- Apply TAL-like operations to the selected elements.
- Just as stylesheets cooperate with “style” attributes embedded in documents, rule sets cooperate with TAL attributes.

TERSE is free of TAL's syntax

- TERSE rule sets are plain text documents, no subject to the limitations that XML attribute syntax places on TAL statements.
- Operations within a rule are applied in the order in which they are written, and different kinds of operation can be mixed together.
- A rule can specify operations to be performed before or after TAL statements attached to the selected element.

TERSE can do more than TAL

In addition to standard TAL operations such as variable definition, content replacement, and repetition, TERSE rules can:

- Insert text before or after an element.
- Search and replace text contained in an element.
- Change the tag-type of an element.
- Use simple conditional logic to decide which operations within a rule to apply.

TAL is still Useful

- While TERSE is a powerful tool, it is easier to use TAL for simple templating tasks.
- TERSE hasn't been implemented yet :-)