

Web template system

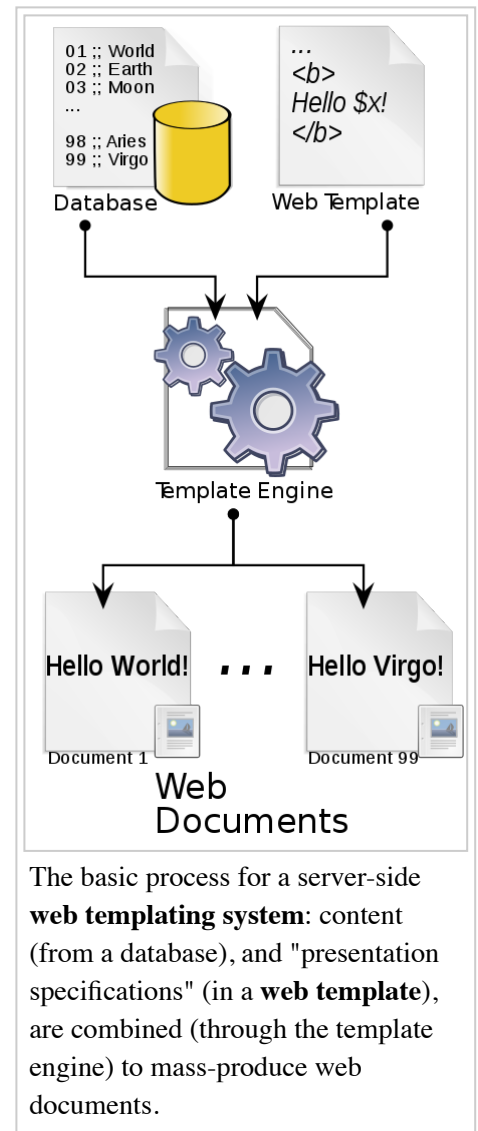
From Wikipedia, the free encyclopedia

A **web template system** uses a template processor to combine **web templates** to form finished web pages, possibly using some data source to customize the pages or present a large amount of content on similar-looking pages. It is a web publishing tool present in content management systems, web application frameworks, and HTML editors.

Web templates can be used like the template of a form letter to either generate a large number of "static" (unchanging) web pages in advance, or to produce "dynamic" web pages on demand.

Contents

- 1 Overview
- 2 Motivations and typical uses
 - 2.1 Applications
 - 2.2 Mass-production
 - 2.3 Style standardization
 - 2.4 Separation of concerns
 - 2.5 Flexible presentation
 - 2.6 Reusability
- 3 Example
- 4 Kinds of template systems
 - 4.1 Static site generators
 - 4.2 Server-side systems
 - 4.3 Edge-side systems
 - 4.4 Client-side systems
 - 4.5 Distributed systems
- 5 See also
- 6 References
- 7 External links



Overview

A *web template system* is composed of the following:

- A template engine: the primary processing element of the system;^[1]
- *Content resource*: any of various kinds of input data streams, such as from a relational database, XML files, LDAP directory, and other kinds of local or networked data;
- *Template resource*: *web templates* specified according to a template language;

The template and content resources are processed and combined by the template engine to mass-produce web documents. For purposes of this article, web documents include any of various output formats for transmission over the web via HTTP, or another Internet protocol.

Motivations and typical uses

Applications

Web templates can be used by any individual or organization to set up their website. Once a template is purchased or downloaded, the user will replace all generic information included in the web template with their own personal, organizational or product information. Examples of common uses of Templates are listed below:

- Display personal information or daily activities as in a blog.
- Sell products online.
- Display information about a company or organization.
- Display family history.
- Display a gallery of photos.
- Place music files such as MP3 files on-line for play through a web browser.
- Place videos online for public viewing.
- To set up a private login area online.

Mass-production

Various agencies and organizations use web template systems for mass-production of content when slower production alternatives prove unfeasible.

For an introductory overview, take a news website as an example. Consider a "static website", where all web pages are static, built by a web designer. It would be very repetitive work to change individual pages as often as the news changes. A typical strategy to automate the web designer's "repetitive work" using Templates could be as follows:

1. choose a *web template system* to maintain the website;
2. group *news items* with different presentation needs;
3. specify the "presentation standards" through web templates, for each group of news;
4. specify a *content resource* to generate or update the content of each *news item*.

Style standardization

Separation of concerns

A common goal among experienced web developers is to develop and deploy applications that are flexible and easily maintainable. An important consideration in reaching this goal is the separation of business logic from presentation logic.^[2] Developers use web template systems (with varying degrees of success) to maintain this separation.^[2]

For the web designer, when each web page comes from a web template, they can think about a modular web page structured with components that can be modified independently of each other. These components may include a header, footer, global navigation bar (GNB), local navigation bar and content such as articles, images, videos etc.

For programmers the template language offers a more restricted logic, only for *presentation* adaptations and decisions, not for complex (business model) algorithms.

For other members of the "site team", a *template system* frees webmasters to focus on technical maintenance, content suppliers to focus on content, and gives all of them more reliability.

Moreover, it has the following advantages to its use:

- *Ease of design change*: presentation variations on templates are "content invariant", meaning a web designer can update the presentation without wider infrastructural preoccupations. Example of this types of websites^[3]
- *Ease of interface localization*: menus and other presentation standards are easy to make uniform, for users browsing on the site. Using Breadcrumb (navigation) makes any website more user friendly and flexible.

[4]

- *Possibility to work separately* on design and code by different people at the same time. It can be perform while all the codes in a templates are clean design and every block or section of the websites are write with individual commenting system.
- *Responsive web design* is now a mandatory factors for any website. Everything must be perform without any change in responsive design. Bootstrap based website remove the headache of responsive website design.^[5]
- *Ease of documentation* a handy documentation saves more time to understand the whole template and also accelerate the modification process. Professional website designers highly emphasize documentation.

One difficulty in evaluating separation of concerns is the lack of well-defined formalisms to measure when and how well it is actually met.^[2] There are, however, fairly standard heuristics that have been borrowed from the domain of software engineering. These include 'inheritance' (based on principles of object-oriented programming); and 'templating and generative programming', (consistent with the principles of MVC separation).^[6] The precise difference between the various guidelines is subject to some debate, and some aspects of the different guidelines share a degree of similarity.^[7]

Flexible presentation

One major rationale behind "effective separation" is the need for maximum flexibility in the code and resources dedicated to the presentation logic.^[6] Client demands, changing customer preferences and desire to present a "fresh face" for pre-existing content often result in the need to dramatically modify the public appearance of web content while disrupting the underlying infrastructure as little as possible.

The distinction between "presentation" (front end) and "business logic" (infrastructure) is usually an important one, because:

- the presentation source code language may differ from other code assets
- the production process for the application may require the work to be done at separate times and locations
- different workers have different skill sets, and presentation skills do not always coincide with skills for coding business logic
- code assets are easier to maintain and more readable when disparate components are kept separate and loosely coupled^[6]

Reusability

Not all potential users of web templates have the willingness and ability to hire developers to design a system for their needs. Additionally, some may wish to use the web but have limited or no technical proficiency. For these reasons, a number of developers and vendors have released web templates specifically for reuse by non-technical people. Although web template reusability is also important for even highly skilled and technically experienced developers, it is *especially* critical to those who rely on simplicity and "ready-made" web solutions.

Such "ready-made" web templates are sometimes free, and easily made by an individual domestically. However, specialized web templates are sometimes sold online. Although there are numerous commercial sites that offer web templates for a licensing fee, there are also free and "open-source" sources as well.

Example

With the model typically held in a relational database, the remaining components of the MVC architecture are the control and view. In the simplest of systems these two are not separated. However, adapting the separation of concerns principle one can completely decouple the relationship.

For example, the view template may look like this:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head><title>Sites</title></head>
<body><h1 data-xp="title"><!-- placeholder --></h1></body>
</html>
```

Then the control template will load the view and then can use xpath addressing to insert components from a database, for instance:

```
<?php
$doc = new DOMDocument;
$doc->preserveWhiteSpace = false;
$doc->Load('view.html');
$titleNode = $doc->createTextNode("Like this");
$xpath = new DOMXPath($doc);
$xpath->registerNamespace("h", "http://www.w3.org/1999/xhtml");
$query="//h:*[@data-xp='title']/comment()";
$entries = $xpath->query($query);
foreach ($entries as $entry) { $entry->parentNode->replaceChild($titleNode, $entry); }
echo $doc->saveXML();
?>
```

Kinds of template systems

A web browser and web server are a client–server architecture; often a web cache is also used to improve performance. Overall, there are five types of templating systems, classified based on when assembly happens - placeholders are substituted for variable information (such as names, addresses, and product info) and sub-templates:

- Server-side - run-time substitution happens on the web server
- Client-side - run-time substitution happens in the web browser
- Edge-side - run-time substitution happens on a proxy between web server and browser
- Outside server - static web pages are produced offline and uploaded to the web server; no run-time substitution
- Distributed - run-time substitution happens on multiple servers

Template languages may be:

- Embedded or event-driven.
- Simple, iterable, programmable, or complex.
- Defined by a consortium, privately defined, or de facto defined by an open implementation. Ownership influences the stability and credibility of a specification. However, in most jurisdictions, language specification cannot be copyrighted, so control is seldom absolute.

The source code of the template engine can be proprietary or open source.

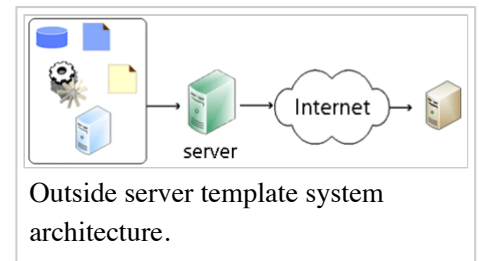
Many template systems are a component of a larger programming platform or framework. They are referred to as the "platform's template system". Some template systems have the option of substituting a different template language or engine.

Programming languages such as Perl, Ruby, C, and Java support template processing either natively, or through add-on libraries and modules. JavaServer Pages (JSP), PHP, and Active Server Pages (ASP with VBScript, JScript or other languages) are examples, themselves, of web template engines. These technologies are typically used in server-side templating systems, but could be adapted for use on a "edge-side" proxy or for static page generation.

Static site generators

HTML editors often use web template systems to produce only static web pages. These can be viewed as a ready-made web design, used to mass-produce "cookie-cutter" websites for rapid deployment. They also commonly include themes in place of CSS styles. In general, the template language is to be used only with the editor's software.^[8]

FrontPage and Dreamweaver were once the most popular editors with template sub-system. A Flash web template uses Macromedia Flash to create visually interactive sites.



System label/name	Platform/editor	Notes
Dreamweaver	Macromedia	HTML authoring. Embedded iterable language.
Contribute	Macromedia	Client authoring.
Flash	Macromedia	Flash authoring.
FrontPage	Microsoft	HTML authoring. Embedded iterable language.
Nvu	Linux/Nvu	HTML authoring.
Website Meta Language	Unix-like	

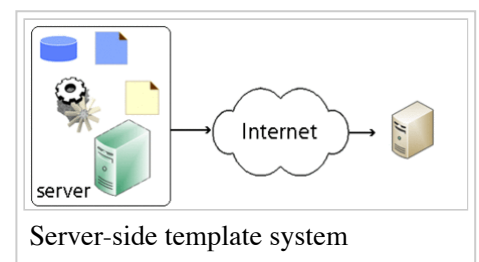
Many *server-side template systems* have the option to publish the output pages on the server, where the published pages will be static. It is a common feature on content management systems, like Vignette. But this does not have to be considered an out-server generation.

In the majority of the cases, this "publish option" doesn't interfere with the *template system*, and it can be made by external software, as Wget.

Server-side systems

Server-side dynamic pages began to be generated by templates with pre-existent software adapted for this task. This early software was the preprocessors and macro languages, adapted for the web use, running on CGI. Next, a simple but relevant technology was the direct execution made on extension modules, started with SSI.

Many *template systems* are typically used as *server-side template systems*:



System label/name	Platform/framework	Notes
CheetahTemplate	Python	Public. Embedded complex language.
Django	Python	Use the "Django template language".
FreeMarker	Java	Public.
Facelets	Java EE	Public. Part of JavaServer Faces
Genshi	Python	Public
Haml	Ruby or Other	Public.
Hamlets	Java	Public.
Jinja2	Python	Public. Embedded complex language.
Kid	Python	
Lasso	LassoSoft, LLC (http://www.lassosoft.com/)	Proprietary. Interpreted Programming Language and Server
Mustache	ActionScript, C++, Clojure, CoffeeScript, ColdFusion, D, Erlang, Fantom, Go, Java, server-side JavaScript, Lua, .NET, Objective-C, ooc, ^[9] Perl, PHP, Python, Ruby, Scala, Tcl	Public.
Basic Server Side Includes (SSI)	The basic directives fix a "standard".	Embedded simple language, if exclude exec directive.
Smarty	PHP	Public. Embedded complex language.
Template Toolkit	Perl	Public. Embedded complex language.
Template Attribute Language (TAL)	Zope, Python, Java, Perl, PHP, XSLT	Public; a.k.a. Zope Page Templates (ZPT); see also TAL Expression Syntax (TALES), Macro Expansion TAL (METAL)
Tiles	Java	Public. Supports multiple template languages (JSP, Velocity, Freemarker, Mustache) from various frameworks (servlet, portlets, struts, spring).
Thymeleaf	Java	Public.
Topsite	Python	Public. <i>"As of 2008-02-20, this project is no longer under active development."</i> ^[10]
PHPLib	PHPLib	Public. Embedded iterable language.
WebMacro	Java	Public. Embedded iterable language.
WebObjects	Java	Use the WebObjects Builder as engine.
Velocity (Jakarta/Apache)	Java	Public. Use VTL - Velocity Template Language (http://velocity.apache.org/engine/devel/vtl-reference-guide.html).
Vignette	Proprietary.	Commercial solution. Embedded complex language.
VlibTemplate	PHP	Public.

System label/name	Platform/framework	Notes
XSLT (standard language)	Any with an XSLT parser	Standard. Event-driven programmable language.
XQuery (standard language)	Any with an XQuery parser	Standard. Embedded programmable language.

Technically, the methodology of embedding programming languages within HTML (or XML, etc.), used in many "server-side included script languages" are also templates. All of them are Embedded complex languages.

System label/name	Notes
Active Server Pages (ASP)	Proprietary (Microsoft platform). See also: VBScript, Javascript, PerlScript, etc. extensions for ASP.
eRuby	Public (Ruby).
ColdFusion Markup Language (CFM)	Public (Lucee, Railo, OpenBD). Proprietary (Adobe ColdFusion).
JavaServer Pages (JSP)	Public, Java platform.
Active Perl	Public.
PHP	Public.
OpenACS	Public (Tcl).

There are also preprocessors used as server-side template engines. Examples:

Preprocessor	Notes
C preprocessor	Public. Embedded iterable language. ^[11]
M4	Public. Embedded complex language.

Edge-side systems

Edge-Side template and inclusion systems. “Edge-side” refers to web servers that reside in the space between the client (browser) and the originating server. They are often referred to as “reverse-proxy” servers. These servers are generally tasked with reducing the load and traffic on originating servers by caching content such as images and page fragments, and delivering this to the browser in an efficient manner.

Basic Edge Side Includes (ESI) is an SSI-like language. ESI has been implemented for content delivery networks. The ESI template language may also be implemented in web browsers using JavaScript and Ajax, or via a browser "plug-in".

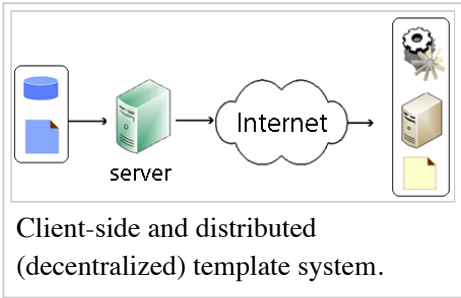
Client-side systems

Many web browsers can apply an XSLT stylesheet to XML data that transforms the data into an XHTML document, thereby providing template functionality in the browser itself.

Other systems implement template functionality in the browser using JavaScript or another client-side scripting language, including:

- Mustache

Distributed systems



The most simple form is transclusions (HTML frames). In other cases dynamic web pages are needed.

Examples:

- Ajax
- Rich Internet application

See also

Concepts:

- Boilerplate code
- Bytecode
- Comparison of web template engines
- Layout engine
- Text substitution macros
- Preprocessor
- Template processor
- Template (file format)
- Transclusion
- Virtual machine

Standards:

- UIML (User Interface Markup Language)
- XSLT (Extensible Stylesheet Language Transformations)

Softwares:

- CodeCharge Studio
- Jekyll

References

1. "Template engine". [phpwact.org wiki](http://phpwact.org/wiki). Archived from the original on December 4, 2012. Retrieved 7 January 2013.
2. Parr, Terence John (2004). *Enforcing strict model-view separation in template engines*. Proceedings of the 13th international conference on World Wide Web. ISBN 1-58113-844-X.
3. Template:Envato
4. [1] (<https://www.smashingmagazine.com/2009/03/breadcrumbs-in-web-design-examples-and-best-practices>)
5. [2] (<http://designbootstrap.net/>)
6. Paragon Corporation (2003-07-19). "Separation of Business Logic from Presentation Logic in Web Applications".
7. MVC vs OOP (<http://c2.com/cgi/wiki?MvcIsNotObjectOriented>)
8. MacDonald, Matthew (2015). *Creating a Website: The Missing Manual*. Chapter 8 > Putting the Same Content on Multiple Pages > Web Templates > Note box: O'Reilly Media, Inc. ISBN 9781491936177. Retrieved 19 January 2016.
9. "[{{mustache}}](http://jekyllrb.com/)". Retrieved 15 October 2013.
10. jodyburns. "Topsite Templating System". Retrieved 15 October 2013.
11. Website, Templates (2013). *Wordpress Website Templates*.

External links

- JavaScript template libraries (<http://psionides.eu/2009/06/15/javascript-template-libraries/>) comparison from 2009
- Enforcing Strict Model-View Separation in Template Engines (<http://www.cs.usfca.edu/~parrr/papers/mvc.templates.pdf>)
- A Double-Model Approach to Achieve Effective Model-View Separation in Template Based Web Applications (<http://www.unirioja.es/cu/fgarcia/JST-ICWE2007.pdf>)
- A PHP template engine comparison with graphic charts (<http://www.phpcomparison.net/>)
- Comparisons/benchmarks of some Python template-engines and some generic thoughts about template-engines (<http://www.simple-is-better.org/template/>)
- web-mode.el is an emacs major for editing web templates (<http://web-mode.org/>)

Retrieved from "https://en.wikipedia.org/w/index.php?title=Web_template_system&oldid=776112971"

Categories: [Scripting languages](#) | [Template engines](#) | [Web design](#)

- This page was last modified on 19 April 2017, at 01:43.
- Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.