

http://theeye.pe.kr	년월일	버전	파일 / 참고	작성자	페이지
	2007. 07. 11	1.1	Velocity Template Language.doc	아이	1(19)
제목					
Velocity Template Language.doc					

Velocity Template Language

2007. 07. 11

<http://theeye.pe.kr>

http://theeye.pe.kr	년월일	버전	파일 / 참고	페이지
	2007-07-11	1.1	Velocity Template Language.doc	2(19)

1. Table of Contents

1.	Table of Contents	2
2.	Revision History.....	3
3.	Summary	4
3.1.	About This guide	4
3.2.	What is Velocity?.....	4
3.3.	Why use Velocity?	4
4.	An Introduction	5
4.1.	Velocity Template Language (VTL)	5
4.2.	Using VTL on HTML document.....	5
4.3.	Hello Velocity World!	5
4.4.	Comments	5
5.	Reference.....	7
5.1.	Variables	7
5.2.	Properties	7
5.3.	Methods	8
5.4.	Property Lookup Rules	8
5.5.	Formal Reference Notation.....	8
5.6.	Quiet Reference Notation	9
6.	Getting Literal	10
6.1.	Currency	10
6.2.	Escaping Valid VTL References	10
7.	Directive.....	11
7.1.	set.....	11
7.2.	Literals	12
8.	Conditionals	14
8.1.	If / ElseIf / Else	14
8.2.	Relational and Logical Operators	14
9.	Loops.....	15
9.1.	Foreach Loop	15
10.	Include / Parse / Stop.....	16
10.1.	Include	16
10.2.	Parse.....	16
10.3.	Stop.....	17
11.	Velocity on Spring Framework.....	18
11.1.	ViewResolver.....	18
11.2.	VelocityConfig.....	18

http://theeye.pe.kr	년월일	버전	파일 / 참고	페이지
	2007-07-11	1.1	Velocity Template Language.doc	3(19)

2. Revision History

Version	Date Updated	Revision Author	Summary of Major Changes Made
1.0	2007-07-10	박우람	문서 신규 작성
1.1	2007-07-11	박우람	문서 추가 작성, Spring 설정 추가

http://theeye.pe.kr	년월일	버전	파일 / 참고	페이지
	2007-07-11	1.1	Velocity Template Language.doc	4(19)

3. Summary

3.1. About This guide

이 문서는 View 쪽에 사용되는 Velocity Template 의 정의와 사용법을 정의하기 위한 문서이다.

3.2. What is Velocity?

Velocity 는 Velocity Template Language 라고도 부르며 줄여서 VTL 이라고 부른다. Velocity 는 Java 기반의 매우 강력한 템플릿 언어이다. Model-View-Controller 로 분리되어 개발되는 MVC 모델의 개발에서 UI 디자이너와 개발자의 병렬적인 개발이 가능하게 되었다. 여기에 더하여 Velocity 는 웹 디자이너로 하여금 Java 코드를 전혀 모르더라도 손쉽게 View 를 제작할 수 있게 해준다.

3.3. Why use Velocity?

Velocity 를 사용함으로써 UI 디자이너의 경우 보기 좋은 사이트를 만드는 데만 집중하면 되고, 프로그래머는 잘 동작하는 코드를 만드는 데만 집중하면 된다. Velocity 는 웹 페이지와 Java 코드를 분리하여, 장기적인 측면에서 볼 때 웹사이트를 손쉽게 유지보수 할 수 있도록 하고, JSP 또는 PHP 를 대체 할 수 있는 방안을 제시한다.

http://theeye.pe.kr	년월일	버전	파일 / 참고	페이지
	2007-07-11	1.1	Velocity Template Language.doc	5(19)

4. An Introduction

4.1. Velocity Template Language (VTL)

Velocity Template Language (VTL)은 동적인 콘텐츠를 포함한 웹 페이지에서 매우 쉽고, 간단하고, 깔끔한 코드를 위한 방안을 제시한다. 약간의 경험이 있는 혹은 전혀 경험이 없는 웹 페이지 개발자라도 약간의 노력만으로 VTL을 웹사이트 개발에 적용할 수 있다.

4.2. Using VTL on HTML document

VTL은 HTML 문서에서 다음과 같은 문법으로 사용된다.

```
#set ( $a = "Velocity" )
```

위의 VTL 문법에서 알 수 있듯이 VTL 문장은 샵(#)으로 시작된다. 변수명의 경우 달러(\$)로 표시한다. 위와 같은 set 함수를 사용하여 a 라는 이름의 변수에 Velocity 라는 값의 문자열을 대입할 수 있다.

4.3. Hello Velocity World!

하나의 변수에 값을 대입하고, 그 값을 출력하는 간단한 예제이다.

```
<html>
<body>
#set ( $foo = "Velocity" )
Hello $foo World!
</body>
</html>
```

위의 HTML 문서의 실행 결과로는 "Hello Velocity World!" 가 출력될 것이다.

4.4. Comments

VTL에서는 다음과 같은 문법으로 주석을 달수 있다. HTML 상의 주석(<!-- -->)과 달리 HTML 문서에서 소스보기를 하여도 보이지 않는다. 프로그래머들간의 의사소통을 위한 주석으로 사용될 수 있다.

```
## This is a single line comment.
```

샵(#)을 두개 붙임으로써 주석의 역할을 하게 된다. 여러 줄의 주석이 필요할 경우 다음과 같이

http://theeye.pe.kr	년월일	버전	파일 / 참고	페이지
	2007-07-11	1.1	Velocity Template Language.doc	6(19)

작성하면 된다.

```

#*
    Thus begins a multi-line comment. Online visitors won't
    see this text because the Velocity Templating Engine will ignore it.
*#

```

다음의 약간의 예제로 VTL 주석을 완벽히 이해할 수 있다.

```

This Text is visible. ## This Text is not.
This Text is visible.
This Text is visible. #* This Text is not visible
This Text is not visible *# This Text is visible.

```

http://theeye.pe.kr	년월일	버전	파일 / 참고	페이지
	2007-07-11	1.1	Velocity Template Language.doc	7(19)

5. Reference

5.1. Variables

VTL 의 변수명 명명에는 다음과 같은 문자를 사용할수 있다.

- 알파벳 (a - z, A - Z)
- 숫자 (0 - 9)
- 하이픈 (“-“)
- 언더바 (“_”)

변수명으로 지정하기 위해 변수명 앞에 달러(\$) VTL 구분자를 붙여주어야 한다. 변수명은 반드시 알파벳 (a - z, A - Z)로 시작하여야 한다.

다음과 같은 변수명이 사용 가능하다.

```
$foo
$mudSlinger
$mud-slinger
$mud_slinger
$mudSlinger1
```

VTL 이 사용된 HTML 문서 안에서 사용할 수 있는 변수로는 문서내부에서 선언된 변수이거나 혹은 Java 코드 안에서 선언된 변수 모두 같은 이름으로 사용할 수 있다.

5.2. Properties

VTL 은 자유롭게 자바 객체의 Property 에 접근할 수 있다. 객체와 하위 Property 를 표시하기 위해 구분자 마침표(.)를 사용한다. 다음의 예를 보자.

```
$customer.Address
$purchase.Total
```

두가지 예제중에 첫번째 예제 한가지만 가지고 이야기를 해보자. 예제를 보면 두가지를 알 수 있다. 첫번째로 customer 는 HashTable 이나 혹은 어떤 데이터를 가지고 있는 객체이며 또한 Address 라는 키를 가지고 그 안에 들어있는 값을 가져오려고 한다는 것을 알 수 있다. 하지만 Address 객체에 직접적으로 접근하는 것이 아니라는 것을 알아야 한다. 위의 예제는 getAddress 함수를 호출한다. 결과적으로 \$customer.getAddress()를 수행하게 된다.

http://theeye.pe.kr	년월일	버전	파일 / 참고	페이지
	2007-07-11	1.1	Velocity Template Language.doc	8(19)

5.3. Methods

Method 는 VTL 에서 효과적으로 자바 객체에 접근할 수 있는 방법이다. Property 와 매우 같은 형태를 갖지만 뒤에 (“(“와 (“)”)가 붙는다.

```
$customer.getAddress()
$purchase.getTotal()
$page.setTitle( "My Home Page" )
$person.setAttribute( ["Strange", "Weird", "Excited"] )
```

\$customer.getAddress() 와 \$purchase.getTotal()을 통해 객체의 특정 변수값을 가져올수 있음을 알 수 있다. 또한 \$page.setTitle("My Home Page")와 같이 Setter Method 를 통해 값을 지정할수도 있다. 뿐만 아니라 \$person.setAttribute(["Strange", "Weird", "Excited"])와 같이 배열도 사용할 수 있다.

5.4. Property Lookup Rules

다음은 Property 별 실제 호출 Method 명 목록이다. 순서대로 Lookup 하게 된다.

- \$customer.address
 1. getAddress()
 2. getAddress()
 3. get("address")
 4. isAddress()

- \$customer.Address
 1. getAddress()
 2. getAddress()
 3. get("Address")
 4. isAddress()

5.5. Formal Reference Notation

다음과 같이 중괄호 (“{””)를 사용하여 Variable, Property, Method 를 안전하게 사용할 수 있다.

```
${mudslinger}
${customer.Address}
${purchase.getTotal()}
```

\$vice 라는 변수에 특정 값이 들어있고 이 값의 출력을 원할 때 다음과 같은 방법으로는 정상적으로 값을 출력할 수가 없다.

http://theeye.pe.kr	년월일	버전	파일 / 참고	페이지
	2007-07-11	1.1	Velocity Template Language.doc	9(19)

Jack is a \$vicemaniac

하지만 중괄호를 사용하여 정상적으로 출력할 수 있다.

Jack is a \${vice}mania

5.6. Quiet Reference Notation

VTL 에서 정의되지 않은 Variable, Property, Method 를 사용하려고 할 때, 그것이 공백으로 처리되기를 바란다면 다음과 같이 느낌표(!)를 붙이면 된다.

```
<input type="text" name="email" value="$!email" />
```

좀더 안전하게 사용하기 위해 다음과 같이 사용해도 된다.

```
<input type="text" name="email" value="$!{email}" />
```

http://theeye.pe.kr	년월일	버전	파일 / 참고	페이지
	2007-07-11	1.1	Velocity Template Language.doc	10(19)

6. Getting Literal

6.1. Currency

VTL 은 작동하기 위해 \$나 #와 같은 특별한 문자들을 사용한다. 하지만 이와 같은 점이 새로운 문제점을 발생시킬 소지가 있다. 물론 해결방법이 있다.

다음의 문장은 \$가 사용되었지만 전혀 문제가 되지 않고 정상적으로 출력이 된다.

```
I bought a 4 lb. sack of potatoes at the farmer's market for only $2.50!
```

\$로 시작을 하였지만 소문자나 대문자의 알파벳으로 시작하지 않고 숫자로 시작되었다. 이것은 일반적인 문자열로 인식되어 그대로 출력이 된다.

6.2. Escaping Valid VTL References

\$로 시작하는 문자열을 일반 문자열로 출력하기 위해서는 역슬래쉬("\")를 사용하면 된다.

```
#set ( $email = "foo" )
$email
\$email
\\$email
\\\email
```

위의 실행 결과물은 다음과 같다.

```
foo
$email
\foo
\$email
```

만약에 선언된 적이 없는 Reference 일 때에는 그대로 출력된다.

```
$email
\$email
\\$email
\\\email
```

http://theeye.pe.kr	년월일	버전	파일 / 참고	페이지
	2007-07-11	1.1	Velocity Template Language.doc	11(19)

7. Directive

7.1. set

#set 지시자는 Variable, Property 등에 값을 세팅 할 때 사용한다. 그렇게 만들어진 변수를 다른 변수의 값으로 지정할 수도 있다. 다음과 같이 사용 가능하다.

```
#set ( $primate = "monkey" )
#set ( $customer.Behavior = $primate )
```

이퀄(“=”)을 기준으로 좌측에는 Variable, Property 명이 명시되어야 하며, 우측으로는 다음과 같은 타입의 값이 들어갈 수 있다.

- Variable
- String
- Property
- Method
- Number
- ArrayList
- Map

다음은 사용 가능한 예제들이다.

```
#set( $monkey = $bill ) ## variable reference
#set( $monkey.Friend = "monica" ) ## string literal
#set( $monkey.Blame = $whitehouse.Leak ) ## property reference
#set( $monkey.Plan = $spindocter.weave($web) ) ## method reference
#set( $monkey.Number = 123 ) ##number literal
#set( $monkey.Say = ["Not", $my, "fault"] ) ## ArrayList
#set( $monkey.Map = { "banana" : "good", "roast beef" : "bad" } ) ## Map
```

ArrayList 의 예제인 \$monkey.Say 는 \$monkey.Say.get(0)과 같은 식으로 배열의 바로 필요한 열의 값을 가져올수 있다. Map 의 예제인 \$monkey.Map 은 \$monkey.Map.get(“banana”) 혹은 \$monkey.Map.banana 와 같은 방식으로 값을 가져올 수 있다.

이퀄(“=”)의 우측에는 다음과 같이 간단한 산술 연산을 사용할 수 있다.

http://theeye.pe.kr	년월일	버전	파일 / 참고	페이지
	2007-07-11	1.1	Velocity Template Language.doc	12(19)

```
#set( $value = $foo + 1 )
#set( $value = $bar - 1 )
#set( $value = $foo * $bar )
#set( $value = $foo / $bar )
```

#set 으로 같은 변수의 값을 재지정하면 가장 최종적으로 지정한 값이 들어가게 된다.

```
#set( $result = $query.criteria("name") )
The result of the first query is $result

#set( $result = $query.criteria("address") )
The result of the second query is $result
```

만약 \$query.criteria("name") 가 “bill”이라는 문자열을 리턴하고 \$query.criteria("address")는 null 을 리턴한다면 \$result 는 계속해서 “bill”이 유지된다.

```
The result of the first query is bill

The result of the second query is bill
```

7.2. Literals

#set 지시자를 사용시에 큰따옴표(“”)안의 내용은 파싱되어 그 결과물이 출력되게 된다.

```
#set( $directoryRoot = "www" )
#set( $templateName = "index.vm" )
#set( $template = "$directoryRoot/$templateName" )
$template
```

위의 결과는 “www/index.vm”이라고 출력될것이다.

큰 따옴표(“”)대신에 작은 따옴표(‘’)를 사용한다면 안의 내용은 파싱되지 않고 그대로 출력된다.

http://theeye.pe.kr	년월일	버전	파일 / 참고	페이지
	2007-07-11	1.1	Velocity Template Language.doc	13(19)

```
#set( $foo = "bar" )
$foo
#set( $blargh = '$foo' )
$blargh
```

위의 실행결과는 “bar” 두번이 아닌, “bar”와 “\$foo”가 될것이다.
 기본적으로 작은따옴표(‘)는 파싱되지 않고 내용이 그대로 출력되지만 velocity.properties 안의 stringliterals.interpolate=false 값을 바꿈으로서 파싱이 되도록 설정할수 있다.

또한 #literal 을 사용하면 파싱없이 문자열 그대로를 출력할 수도 있다.

```
#literal()
#foreach ($woogie in $boogie)
  nothing will happen to $woogie
#end
#end
```

#literal()과 #end 사이의 모든 내용은 다음과 같이 그대로 출력된다.

```
#foreach ($woogie in $boogie)
  nothing will happen to $woogie
#end
```

http://theeye.pe.kr	년월일	버전	파일 / 참고	페이지
	2007-07-11	1.1	Velocity Template Language.doc	14(19)

8. Conditionals

8.1. If / Elself / Else

#if 지시자는 어떤 문장의 참거짓을 구분하여 특정한 처리를 할 때 사용한다. 다음의 예제를 보자.

```
#if( $foo )
    <strong>Velocity!</strong>
#end
```

위의 경우에는 “\$foo”가 True/False 값을 갖는 Boolean 형태일때 True 의 경우 실행이 된다. 또는 “\$foo”가 Boolean 형이 아닐때는 null 이 아니면 참이 된다.

8.2. Relational and Logical Operators

다음과 같이 #if 와 #elseif 도 같이 사용할 수 있다.

```
#if( $foo < 10 )
    <strong>Go North</strong>
#elseif( $foo == 10 )
    <strong>Go East</strong>
#elseif( $bar == 6 )
    <strong>Go South</strong>
#else
    <strong>Go West</strong>
#end
```

위의 예제에서 알 수 있듯이 <, >, == 등의 논리 연산뿐만 아니라 AND(&&), OR(||), NOT(!) 같은 연산도 사용할 수 있다.

http://theeye.pe.kr	년월일	버전	파일 / 참고	페이지
	2007-07-11	1.1	Velocity Template Language.doc	15(19)

9. Loops

9.1. Foreach Loop

#foreach 지시자는 ArrayList 같은 배열형 객체의 엘리먼트들을 사용하는데 용이하다.

```
<ul>
#foreach( $product in $allProducts )
  <li>$product</li>
#end
</ul>
```

“\$allProducts” 리스트에서 첫번째 값부터 시작하여 끝날때까지 타겟 변수인 “\$product”에 대입하게 된다. “\$allProducts”는 Vector, HashTable, Array 의 변수형이 사용될 수 있다.

HashTable 의 경우 다음과 같은 방법으로 값을 출력할 수 있다.

```
<ul>
#foreach( $key in $allProducts.keySet() )
  <li>Key: $key -> Value: $allProducts.get($key)</li>
#end
</ul>
```

http://theeye.pe.kr	년월일	버전	파일 / 참고	페이지
	2007-07-11	1.1	Velocity Template Language.doc	16(19)

10. Include / Parse / Stop

10.1. Include

`#include` 지시자는 로컬 서버안의 파일을 추가로 붙이는데 사용된다. 보안상의 문제로 `TEMPLATE_ROOT` 설정 디렉토리 이하의 파일만 인클루드가 가능하다. 해당 파일 안에 VTL 문법이 있더라도 파싱 되지 않고 문자열로 인식하게 된다.

```
#include( "one.txt" )
```

또는 다음과 같이 여러 개의 파일을 인클루드 할 수 있다. 콤마(,)로 구분한다.

```
#include( "one.gif","two.txt","three.htm" )
```

인클루드할 파일명이 동적으로 변할 수 있는 상황이라면 변수명을 적어주게 되면 해당 파일을 인클루드 하게 된다.

```
#include( "greetings.txt", $seasonalstock )
```

10.2. Parse

`#parse` 지시자는 `Include` 와 동일하지만 Velocity Template Engine 이 파싱을 하여 그 결과물을 출력한다는 점이 다릅니다. 마찬가지로 `TEMPLATE_ROOT` 안의 파일만을 `Parse` 할 수 있다.

```
Count down.
#set( $count = 8 )
#parse( "parsefoo.vm" )
All done with dofoo.vm!
```

위와 같은 내용이 있을 때 “`parsefoo.vm`” 파일의 내용은 다음과 같다.

http://theeye.pe.kr	년월일	버전	파일 / 참고	페이지
	2007-07-11	1.1	Velocity Template Language.doc	17(19)

```

$count
#set( $count = $count - 1 )
#if( $count > 0 )
    #parse( "parsefoo.vm" )
#else
    All done with parsefoo.vm!
#end

```

위의 결과는 “Count down”이 출력되고 “\$count” 변수에 8 이 대입된다. 이후 “parsefoo.vm”을 읽어들이어 현재 “\$count”의 값을 출력하고(처음에는 8) 값을 -1 시킨다. “\$count”가 0 이 될때까지 계속해서 값을 줄여나가며 스스로를 읽어들이다가 0 이 되면 “All donw with parsefoo.vm!”을 두번 출력하고 종료하게 된다. 여기서 알수 있듯이 #parse 로 읽어들이는 파일 안에서도 바깥의 변수에 접근하고 수정할 수 있다.

10.3. Stop

#stop 지시자는 현재의 줄에서 템플릿 파싱을 중단한다. 디버깅시 유용하게 활용될수 있다.

```

#stop

```

http://theeye.pe.kr	년월일	버전	파일 / 참고	페이지
	2007-07-11	1.1	Velocity Template Language.doc	18(19)

11. Velocity on Spring Framework

11.1. ViewResolver

```
<bean id="viewResolver"
  class="org.springframework.web.servlet.view.velocity.VelocityViewResolver">
  <property name="cache" value="true"/>
  <property name="prefix" value="/WEB-INF/vm"/>
  <property name="suffix" value=".vm"/>
  <property name="contentType" value="text/html; charset=UTF-8"/>
</bean>
```

스프링 프레임워크에서 VTL 을 사용하기 위해서는 위와같이 ViewResolver 를 바꾸어 준다. VelocityViewResolver 클래스를 사용하고, cache 를 사용한다. 또한 contentType 을 이곳에서 강제적으로 지정하여 Response 의 헤더에 실려가도록 설정할 수 있다.

11.2. VelocityConfig

```
<bean id="velocityConfig"
  class="org.springframework.web.servlet.view.velocity.VelocityConfigurer">
  <property name="resourceLoaderPath" value="/WEB-INF/vm"/>
  <property name="velocityProperties">
    <props>
      <prop key="directive.foreach.counter.name">loopCounter</prop>
      <prop key="directive.foreach.counter.initial.value">0</prop>
      <prop key="input.encoding">UTF-8</prop>
      <prop key="output.encoding">UTF-8</prop>
    </props>
  </property>
</bean>
```

ViewResolver 설정 바로 밑에 위와 같은 VelocityConfig 를 설정해 주면 된다. resourceLoaderPath 는 VTL 페이지들이 있는곳의 경로를 지정하여준다. 이런 보안적인 제제가 필요없다면 "/"로 설정하면 된다.

밑의 directive.foreach.counter.name 와 directive.foreach.counter.initial.value 설정은 VTL 의 foreach 루프의 기본 루프 카운터가 1 부터 시작하는 것을 0 부터 시작하도록 수정하는 옵션이다.

http://theeye.pe.kr	년월일	버전	파일 / 참고	페이지
	2007-07-11	1.1	Velocity Template Language.doc	19(19)

기타 `velocity.properties` 에 설정되는 옵션들을 이곳에 설정할수 있다.

그 밑의 `input.encoding` 와 `output.encoding` 는 인풋과 아웃풋시의 인코딩 캐릭터셋을 지정하여 준다. 마찬가지로 UTF-8 프로젝트시 UTF-8 로 지정하면 된다.