

Activity 1: Create and Update Pages Based on Conditions

Objective:

Use conditionals in Velocity to display a message based on the current day of the week.

Steps:

1. Get the current date:

```
velocity  
  
#set($currentDate = $datetool.get("EEEE", $datetool.getDate()))
```

2. Use #if, #elseif, #else blocks:

```
velocity  
  
#if($currentDate == "Monday")  
    Start of the Week!  
#elseif($currentDate == "Friday")  
    Almost Weekend!  
#else  
    Have a productive day!  
#end
```

Expected Outcome:

Learners understand how to use conditionals with dynamic data in Velocity.

 Activity 2: List All Pages in a Space**Objective:**

Use loops to retrieve and display all page names from a specific space.

Steps:

1. Set a target space:

```
velocity
#set($space = "Sandbox")
```

2. Use \$xwiki.getDocumentsName(\$space) to retrieve documents.

```
velocity
#set($docs = $xwiki.getDocumentsName($space))
#foreach($docName in $docs)
    * $docName<br/>
#end
```

Expected Outcome:

Learners understand how to iterate through a collection and access document names.

Activity 3: Display Document Titles with Links

Objective:

List documents in a space and link each title to its page.

Steps:

1. Retrieve document names:

```
velocity
#set($docs = $xwiki.getDocumentsName("Sandbox"))
```

2. Build links with \$xwiki.getDocument():

```
velocity
#foreach($docName in $docs)
    #set($doc = $xwiki.getDocument("${docName}"))
    * [$doc.displayTitle] ($doc.getURL())<br/>
#end
```

Expected Outcome:

Learners build real navigation lists using document objects.

Activity 4: Display Logged-in User Info

Objective:

Show current user's first name, last name, and profile link.

Steps:

1. Get current user:

```
velocity
#set($user = $xwiki.getUser($xcontext.user))
```

2. Display user details:

```
velocity
* Username: $xcontext.user <br/>
* First Name: $user.getValue("first_name") <br/>
* Last Name: $user.getValue("last_name") <br/>
* [Profile] ($user.getURL())
```

Expected Outcome:

Learners understand how to access and display user properties.

Activity 5: Add Content to a Page Based on Tags

Objective:

Show custom content if a page has certain tags.

Steps:

1. Get the current document:

```
velocity

#set($doc = $xwiki.getDocument($doc.fullName))
#set($tags = $doc.getTags())
```

2. Check tags and conditionally show a message:

```
velocity

#if($tags.contains("important"))
    <b>This page contains important content!</b>
#end
```

Expected Outcome:

Learners will work with tags and conditional logic in Velocity.

Activity 6: Generate a Table of Contents from Child Pages

Objective:

List child pages of a parent and build a dynamic TOC.

Steps:

1. Set parent reference:

```
velocity
#set($parent = "Sandbox.WebHome")
#set($children = $xwiki.getChildren($parent))
```

2. Loop and format as a TOC:

```
velocity
<ul>
#foreach($child in $children)
  <li><a href="$child.getURL()">$child.displayTitle</a></li>
#end
</ul>
```

Expected Outcome:

Learners create hierarchical navigation based on document structure.

Activity 7: Render XWiki Object Properties Dynamically

Objective:

Display custom object properties (e.g., from a class) in a formatted block.

Steps:

1. Get the current document:

```
velocity
#set($doc = $xwiki.getDocument($doc.fullName))
```

2. Access object fields:

```
velocity
#set($obj = $doc.getObject("MySpace.MyClass"))
#if($obj)
  * Name: $obj.getProperty("name").value <br/>
  * Age: $obj.getProperty("age").value <br/>
#end
```

Expected Outcome:

Learners interact with structured object data stored in pages.

Activity 8: Create a Velocity Macro for Reuse

Objective:

Define and use a Velocity macro that displays a user profile card.

Steps:

1. Define macro:

```
velocity

#macro(userCard $username)
    #set($user = $xwiki.getDocument("XWiki.${username}"))
    <div class="card">
        <b>$user.displayTitle</b><br/>
        Email: $user.getValue("email")
    </div>
#end
```

2. Call macro:

```
velocity

#userCard("Admin")
```

Expected Outcome:

Learners write reusable macros with parameters.

Activity 9: Display Recently Modified Pages

Objective:

Fetch and display a list of recently updated pages in a space.

Steps:

1. Use \$services.query API:

```
velocity

#set($query = "from doc where doc.space = 'Sandbox' order by doc.date
desc")
#set($results = $services.query.hql($query).setLimit(5).execute())
```

2. Show results:

```
velocity

#foreach($result in $results)
    #set($doc = $xwiki.getDocument($result))
    * <a href="$doc.getURL()">$doc.displayTitle</a> -
    $datetool.format("yyyy-MM-dd", $doc.date)<br/>
#end
```

Expected Outcome:

Learners perform dynamic HQL queries via Velocity.

 Activity 10: Conditional Rendering Based on User Rights**Objective:**

Show or hide content depending on user's admin status.

Steps:

1. Use `#if` with `$xwiki.hasAdminRights()`:

```
velocity

#if($xwiki.hasAdminRights())
    <div class="admin-box">Welcome, Administrator!</div>
#else
    <div class="user-box">Welcome, User!</div>
#end
```

Expected Outcome:

Learners apply security-aware logic in UIs using Velocity.