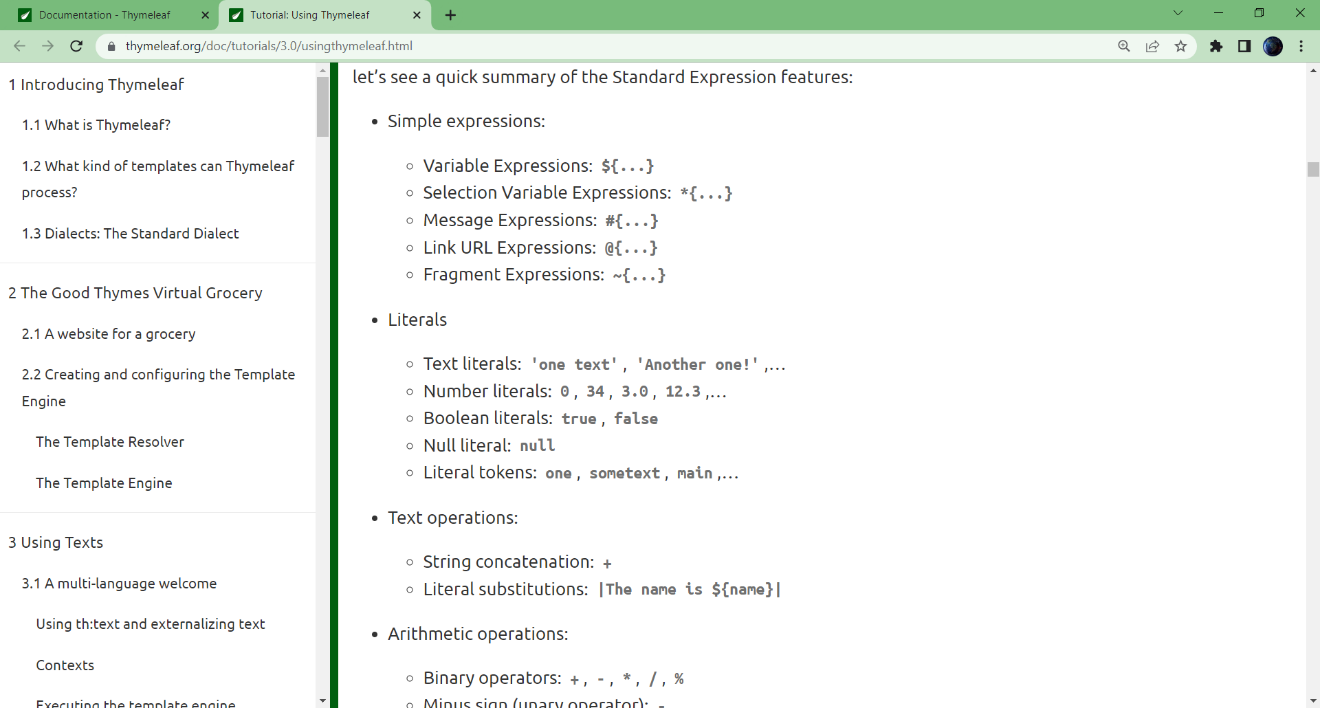
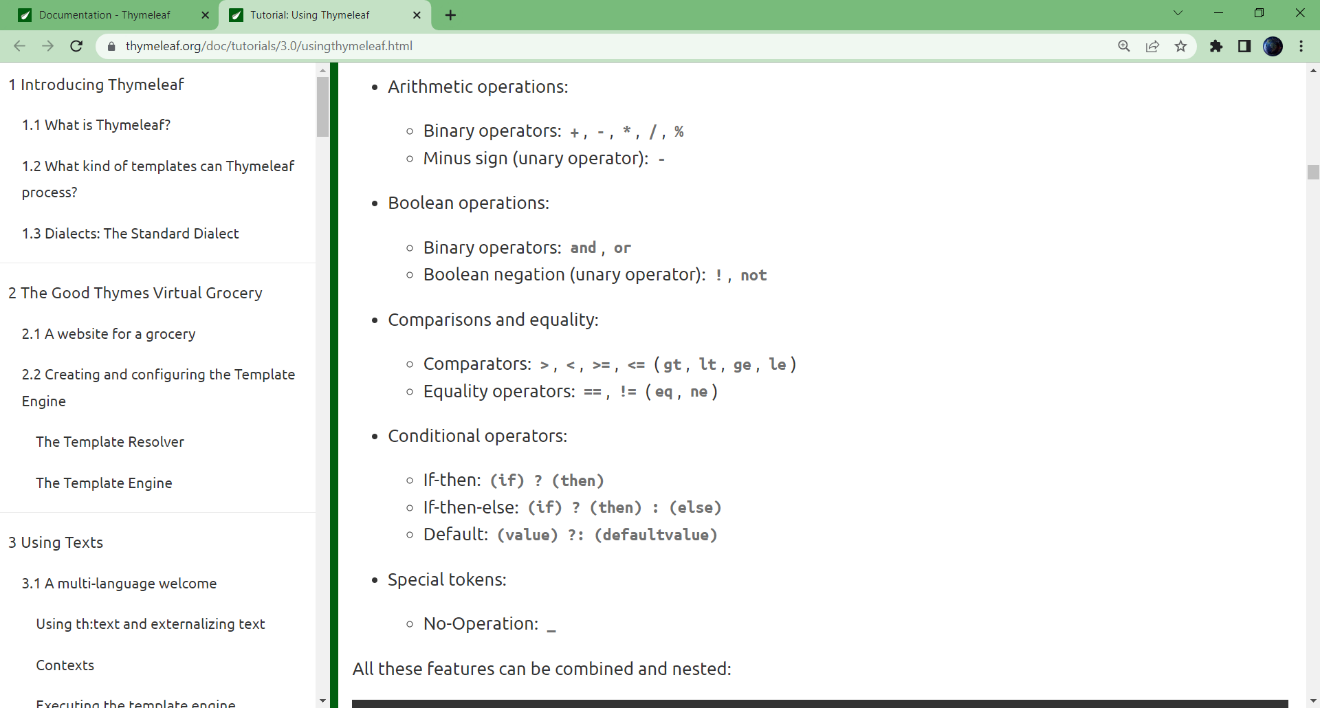
**Standard Expression Syntax:**





**Literals:**

- Literals can be used in Thymeleaf syntax:

Ex:

*<p th:text="${user.isAdmin()} == false ? 'A' : (0.5 + 2)"></p>*

- Token: If text does not have whitespace and special characters, can remove '' surround it.

*<p th:text="content"></p>* instead of *<p th:text="'content'"></p>*

- No-operation Token (\_) means "ignore this Thymeleaf attributes"

Ex: *<p th:text="${user}?: \_"></p>* so if user is null, nothing in the body of <p>

**Messages:**

- A Key-Value item in properties file.

- Extract using #{<name>}

Ex: In the file home.properties located at the same folder as home.html, we have:

*home.welcome=Some unicode text here*

And in the home.html:

*<p th:utext="#{home.welcome}">The welcome message here will be overrided</p>*

- The Values can have params follow java.text.MessageFormat syntax. The params can be substituted by **Variables**.

**Variables**:

- An Object stored in the Context.

- Can get by ***${<name>}***

- Can get properties and method on that Object using '.'

- Properties, Array item or Map item can access using ['<property\_name>'] or [<index>]

Ex: *<p th:utext="${today}"></p>*

**Selection**:

- ***th:object***: Select a specific Object for extracting information in the scope

- Extract information from Object using \*{<name>}

Ex:

<div th:object="${user}">

<p th:text="\*{name}"></p>

<p th:text="\*{age}"></p>

<p th:text="${#object.address}"></p> // #object return the selected object.

</div>

*Same as*:

<div>

<p th:text="${user.name}"></p>

<p th:text="\*{user.age}"></p> // If no Object $ and \* are the same

</div>

**Link**:

- URL also has some variables and params set by Context.

- An "need to be computed" URL place on @{<url>}

- params on URL place on (key=${<value>},...) separated by ','

- relative url start with '/' will be append a base\_url to ***context-root***

- Use @{~/relative\_path...} to indicate this relative\_path should be append a base\_url to ***Server-root***.

Ex:

*<a href="details.html" th:href="@{/order/{orderId}/details(orderId=${o.id})}">view</a>*

will produce

*[base\_url]/order/3/details?orderId=5*

with variables in Context: orderId=3, o.id=5

**Some Thymeleaf attributes:**

- th:text : Escaped text. Hello <b>World</b> -> Hello &lt;b&gt;World&lt;/b&gt;

- th:utxet : Unescaped text. Hello <b>World</b> -> Hello <b>World</b>

- th:attr : Set attributes for template, separates many by ','

<input type="submit" th:attr="value=${message.displayText}"/>

-> <input type="submit" value="Subscribe!"/>

- th:<attr> : Set value for attributes <attr> of current tag. Instead of using ***th:attr***

Ex: th:href, th:value, th:action, th:for, th:label,....

- th:<attr1>-<attr2> : Set same value for 2 attributes.

Ex: th:alt-title

- th:attrappend, th:attrprepend : Append (Prepend) value for sepecific attributes

Ex class="btn" th:attrappend="class=${button.status.css}" -> class="btn warning"

- th:classappend, th:styleappend

**Iterations**:

- Syntax: th:each="item : ${list\_items}"

- item just used in sub-tag of current tag

- Can use with ***th:block*** (Not a actual Tag, just for using Thymeleaf attribute like th:each)

- ***Current Item status***: declare after "item" and before ":" (or default name is ***itemStat*** if "item" is the name of iterator). This variable has some properties: *index* (0,1,...), *count* (1,2,...), *size* (size of list), *current* (same as item), *even*, *odd*, *first*, *last* (boolean)

**Conditions**:

- Syntax: ***th:if="<condition expression>"*** or ***th:unless*** for ***th:if="not <condidtion>"***.

- The current tag (and sub-tags) will be remove if the condition false and otherwise

- Switch-case: ***th:switch="..."*** in tag and ***th:case="..."*** in sub-tags, use ***th:case="\*"*** for default case.

**Fragments**:

- Declare a Tag (and subtags) as a Fragment (footer, header, sidebar,...) use:

th:fragment="<name>"

- In the Tag using that fragment, using ***th:insert*** or ***th:replace***:

th:insert="~{<file\_declare\_fragment>::<fragment\_name>}"

Or, replace using <fragment\_name> with selector like "#element\_id" or "p div.footer"

- th:insert: Insert Fragment (The container tag and all subtags) inside current Tag

- th:replace: Replace Body of current tag with The Fragment

- th:include: Insert Fragment (The subtags only) inside current Tag

- ***Parameters for Fragment***:

Declare with : th:fragment="<name>(<param,>)"

Calling : th:insert="...::<name>(<param=value,>)"

param name can be removed if passing arguments in order

param value can be a fragment selector ~{... :: ...} so can created a inheritance layout

**Remove**:

- If want Thymeleaf remove a Tag when it processing. th:remove

- ="all" : Current Tag + All Subtags

- ="body" : All Subtags only

- ="tag" : Current Tag only

- ="all-but-first" : All Subtags except the first children (Remain: Current + First Child)

- ="none" : Nothing (For dynamic valuation)

**Local variables:**

- Declare with: th:with="<name>=<value>,..."

- Can be in scope of container Tag.

**Inlining**:

th:text -> [[<expression>]]

th:utext -> [(<expression>)]

**Form on Spring**

* Using ***th:object="${…}"*** for form object. The variable must not be a property of an object.
* Using ***th:field="\*{…}"*** for binding the form-field with specific property of the form object.
* th:field can be apply to <input>, <select>, <textarea>,…