



BASIC THYMELEAF


SLIDE 3.1

PHẦN I: CƠ BẢN

 GIỚI THIỆU THYMELEAF

 BIỂU THỨC THYMELEAF (EXPRESSION)

 ĐỊNH DẠNG DỮ LIỆU (FORMATING DATA)

 TOÁN TỬ (OPERATORS)

PHẦN II: CHỈ THỊ THYMELEAF (DIRECTIVES)

 CHỈ THỊ RENDER NỘI DUNG

 CHỈ THỊ RENDER THUỘC TÍNH

 CHỈ THỊ HỖ TRỢ

 CHỈ THỊ PHÂN LƯỒNG (FOLLOW CONTROL)





INTRODUCTION

- ❑ Thymeleaf là template engine (một động cơ tạo mẫu) chạy phía server có khả năng xử lý việc tạo ra HTML, XML, CSS, JS và TEXT.
- ❑ Thymeleaf dựa vào các nguyên mẫu (HTML, CSS...) và bổ sung các quy luật cần thiết để sinh ra template hoàn thiện với dữ liệu động.
- ❑ So với JSP thì Thymeleaf render đơn giản hơn đối với lập trình viên và render cũng nhanh hơn.
- ❑ Thư viện phụ thuộc

```
<dependency>
```

```
    <groupId>org.springframework.boot</groupId>
```

```
    <artifactId>spring-boot-starter-thymeleaf</artifactId>
```

```
</dependency>
```

QUICK START WITH THYMELEAF

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <title th:text="${title}">Page Title</title>
</head>
<body>
  <h1 th:text="${subject}">Page Subject</h1>
  
</body>
</html>
```

- ❑ Namespace: th
- ❑ Directives
 - ❖ th:text
 - ❖ th:src
- ❑ Expressions
 - ❖ \${...}
 - ❖ @{|...|}
- ❑ Model attribute (variable)
 - ❖ title
 - ❖ subject
 - ❖ photo

- ❑ Giả sử giá trị của các biến trong Model là
 - ❖ **title**: FPT Polytechnic
 - ❖ **subject**: Spring Boot MVC
 - ❖ **photo**: trump.png
- ❑ Thì kết quả response về trình duyệt sẽ là

```
<!DOCTYPE html>
<html>
<head>
  <title>FPT Polytechnic</title>
</head>
<body>
  <h1>Spring Boot MVC</h1>
  
</body>
</html>
```

❑ Cú pháp:

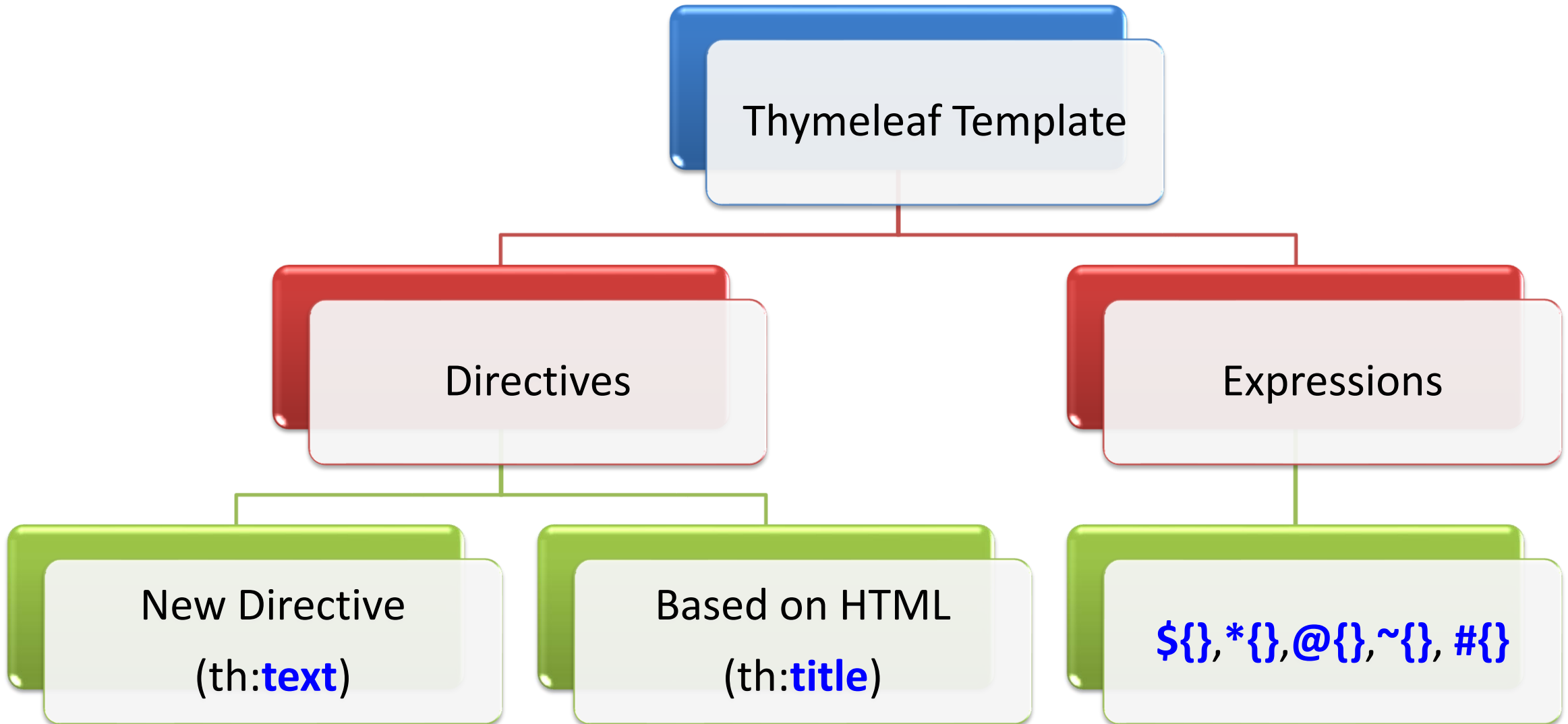
- ❖ `th:directive="expression"`

❑ Hai loại directive

- ❖ Mới do thymleaf định nghĩa: `th:text` vì text không phải thuộc tính của thẻ nào trong HTML
- ❖ Dựa trên nguyên mẫu của HTML: `th:src` vì `src` là thuộc tính của thẻ `img` trong HTML.

❑ Biểu thức

- ❖ `${}`
- ❖ `@{}`
- ❖ ...



DIRECTIVES & PRECEEDING (THỨ TỰ ƯU TIÊN)

TT	HÀNH ĐỘNG	CÁC THUỘC TÍNH
1	Bao hàm các fragment	th: insert , th: replace
2	Vòng lặp	th: each
3	Đánh giá điều kiện	th: if , th: unless , th: switch , th: case
4	Định nghĩa biến cục bộ	th: object , th: with
5	Hiệu chỉnh thuộc tính	th: attr , th: attrprepend , th: attrappend
6	Hiệu chỉnh thuộc tính HTML	th: value , th: href , th: src ...
7	Thay đổi nội dung thẻ	th: text , th: utext
8	Khai báo fragment	th: fragment
9	Xóa fragment	th: remove

- ❑ **`${expr}`** - Biểu thức biến (variable expression)
 - ❖ Truy xuất các biến trong Model, session, application, param...
- ❑ **`*{expr}`** - Biểu thức lựa chọn (selection expression):
 - ❖ Truy xuất các thuộc tính của bean được chọn bởi directive th:object
- ❑ **`#{expr}`** - Biểu thức thông điệp (message expression):
 - ❖ Truy xuất các message khai báo trong tài nguyên đa ngôn ngữ
- ❑ **`@{expr}`** - Biểu thức đường dẫn (URL expression):
 - ❖ Truy xuất các biến và tạo ra đường dẫn
- ❑ **`~{expr}`** - Biểu thức phân đoạn (fragment expression):
 - ❖ Truy xuất các phân đoạn template được khai báo trong file template nào đó



EXPRESSIONS

`${}` - SIMPLE VARIABLE EXPRESSION

@Controller

```
public class StudentController {  
    @RequestMapping("/poly/student")  
    public String detail(Model model) {  
        model.addAttribute("fullname", "Nguyễn Văn Tèo");  
        model.addAttribute("mark", 7.5);  
        model.addAttribute("gender", true);  
        model.addAttribute("birthday", new Date());  
        model.addAttribute("photo", "photo.png");  
        return "/demo/student";  
    }  
}
```

```
<h2>Thông tin sinh viên</h2>  
<ul>  
    <li>Nguyễn Văn Tèo</li>  
    <li>true</li>  
    <li>Mon Dec 09 07:56:55 ICT 2024</li>  
    <li>7.5</li>  
    <li>photo.png</li>  
</ul>
```

```
<h2>Thông tin sinh viên</h2>  
<ul>  
    <li th:text="${fullname}>Fullname</li>  
    <li th:text="${gender}>Gender</li>  
    <li th:text="${birthday}>Birthday</li>  
    <li th:text="${mark}>Everage Mark</li>  
    <li th:text="${photo}>Photo</li>  
</ul>
```

`${}` - BEAN PROPERTY VARIABLE EXPRESSION

```
@RequestMapping("/poly/student")
public String detail(Model model) {
    Student bean = Student.builder()
        .fullname("Nguyễn Văn Tèo")
        .mark(7.5).gender(true).build();
    model.addAttribute("student", bean);
    return "/demo/student";
}
```

```
@NoArgsConstructor
@AllArgsConstructor
@Data
@Builder
public class Student {
    private String fullname;
    private boolean gender;
    @Default
    private Date birthday = new Date();
    @Default
    String photo = "photo.png";
    private double mark;
}
```

```
<h2>Thông tin sinh viên</h2>
<ul>
    <li th:text="${student.fullname}">Fullname</li>
    <li th:text="${student.gender}">Gender</li>
    <li th:text="${student.birthday}">Birthday</li>
    <li th:text="${student.mark}">Everage Mark</li>
    <li th:text="${student.photo}">Photo</li>
</ul>
```

```
<h2>Thông tin sinh viên</h2>  
<ul th:object="$student">  
  <li th:text="*{fullname}">Fullname</li>  
  <li th:text="*{gender}">Gender</li>  
  <li th:text="*{birthday}">Birthday</li>  
  <li th:text="*{mark}">Everage Mark</li>  
  <li th:text="*{photo}">Photo</li>  
</ul>
```

- ❑ `th:object="$student"`: chọn đối tượng
- ❑ `*{fullname}`: truy xuất thuộc tính fullname (`getFullname()`) của đối tượng đã được chọn

<h2>URL Expression</h2>

<ul th:object="{student}">

<li th:text="@{/photos/{x}(x={photo})}">Photo

<li th:text="@{'/photos/' + {photo}}">Photo

<li th:text="@{||/photos/{photo}||}">Photo

<li th:text="@{/photos(x={photo})}">Photo

<li th:text="@{'/photos?x=' + {photo}}">Photo

<li th:text="@{||/photos?x={photo}||}">Photo

<h2>URL Expression</h2>

/photos/photo.png

/photos/photo.png

/photos/photo.png

/photos?x=photo.png

/photos?x=photo.png

/photos?x=photo.png

❑ Khuyến nghị:

❖ @{|/photos/{photo}'|} => /photos/photo.png

❖ @{|Ảnh: {photo}|} => Ảnh: photo.png

<h2>Thông tin sinh viên</h2>

<ul th:object="{student}">

<li th:text="{#strings.capitalizeWords(fullname)}">Fullname

<li th:text="{gender ? 'Male' : 'Female'}">Gender

<li th:text="{#dates.format(birthday, 'MMM dd, yyyy')}">Birthday

<li th:text="{#numbers.formatDecimal(mark, 0, 2)}">Everage Mark

<li th:text="@{/photos/{photo}}">Photo

☐ Viết hoa đầu mỗi từ

❖ **#strings.capitalizeWords**(fullname)

☐ 2 chữ số lẻ

❖ **#numbers.formatDecimal**(mark, 0, 2)

❖ **#numbers.formatDecimal**(mark, 'POINT', 0, 'COMMA', 2)

☐ Định dạng thời gian

❖ **#dates.format**(birthday, 'MMM-dd, yyyy')

```
<h2>Thông tin sinh viên</h2>
<ul>
  <li>Nguyễn Văn Tèo</li>
  <li>Male</li>
  <li>Dec 09, 2024</li>
  <li>7.50</li>
  <li>/photos/photo.png</li>
</ul>
```


ACCESSING SESSION, SERVLETCONTEXT & PARAMETER

```

@SessionAttributes("var1")
@Controller
public class MyController {
    @Autowired
    HttpSession session;
    @Autowired
    ServletContext servletContext;

    @RequestMapping("/my-url")
    public String myMethod(Model model) {
        model.addAttribute("var1", "Value 1");
        session.setAttribute("var2", "Value 2");
        servletContext.setAttribute("var3", "Value 3");
        return "/my-template";
    }
}

```

localhost:8080/my-url?p=Value 0

- Value 0
- Value 1
- Value 2
- Value 3

```

<ul>
  <li>Value 0</li>
  <li>Value 1</li>
  <li>Value 2</li>
  <li>Value 3</li>
</ul>

```

```

<ul>
  <li th:text="${param.p}"></li>
  <li th:text="${session.var1}"></li>
  <li th:text="${session.var2}"></li>
  <li th:text="${application.var3}"></li>
</ul>

```

❑ Xử lý chuỗi:

❖ +

❖ |<hỗn hợp chuỗi và biến>|

❑ Số học:

❖ +

❖ -

❖ *

❖ /

❖ %

❑ Điều kiện

❖ Condition ? trueValue : falseValue

❑ Logic:

❖ and

❖ or

❖ Not hoặc !

❑ So sánh:

❖ < hoặc lt

❖ <= hoặc le

❖ > hoặc gt

❖ >= hoặc ge

❖ == hoặc eq

❖ != hoặc ne



BASIC THYMELEAF

SLIDE 3.2



THYMELEAF DIRECTIVES

❑ Rendering content

- ❖ th:text, th:utext
- ❖ inline

❑ Rendering attribute

- ❖ th:<html-attr> (th:href, th:src...)
- ❖ th:attr, th:attrappend, th:attrprepend
- ❖ th:classappend, th:classprepend

❑ Controlling support

- ❖ th:block
- ❖ th:object
- ❖ th:with

❑ Follow Control

- ❖ th:if, th:unless
- ❖ th:switch
- ❖ th:each

❑ Data Binding

- ❖ th:field
- ❖ th:error

❑ Fragment

- ❖ th:fragment
- ❖ th:replace
- ❖ th:insert
- ❖ th:remove

```
@Controller
public class MyController {
    @RequestMapping("/my-url")
    public String method(Model model) {
        model.addAttribute("var", "FPT <b>Polytechnic</b>");
        return "/demo/my-template";
    }
}
```

HTML Encoded Data

- FPT Polytechnic
- Inline: FPT Polytechnic

HTML Raw Data

- FPT **Polytechnic**
- Inline: FPT **Polytechnic**

```
<h4>HTML Encoded Data</h4>
<ul>
    <li th:text="${var}"></li>
    <li>Inline: [[${var}]]</li>
</ul>
<h4>HTML Raw Data</h4>
<ul>
    <li th:utext="${var}"></li>
    <li>Inline: [(${var})]</li>
</ul>
```

```
<h4>HTML Encoded Data</h4>
<ul>
    <li>FPT &lt;b&gt;Polytechnic&lt;/b&gt;</li>
    <li>Inline: FPT &lt;b&gt;Polytechnic&lt;/b&gt;</li>
</ul>
<h4>HTML Raw Data</h4>
<ul>
    <li>FPT <b>Polytechnic</b></li>
    <li>Inline: FPT <b>Polytechnic</b></li>
</ul>
```

RENDER ATTRIBUTE DIRECTIVES

```
<h4>HTML Attributes</h4>
<a th:href="@{/images/${photo}}">
    
</a>
<h4>Create, Append, Prepend Attributes</h4>
<ul>
    <li title="${photo}"></li>
    <li th:attr="title=${photo}"></li>
    <li title="Nghiem" th:attrprepend="title=${ho}"></li>
    <li title="Nguyễn " th:attrappend="title=${ten}"></li>
</ul>
<h4>Append, Prepend Classes</h4>
<ul>
    <li class="btn " th:attrappend="class=${myclass}"></li>
    <li class="btn" th:classappend="${myclass}"></li>
</ul>
```

Model Attributes:

- photo: avatar.png
- ho: Nguyễn
- ten: Nghiệm
- myclass: btn-danger

```
<h4>HTML Attributes</h4>
<a href="/images/avatar.png">
    
</a>
<h4>Create, Append, Prepend Attributes</h4>
<ul>
    <li title="${photo}"></li>
    <li title="avatar.png"></li>
    <li title="Nguyễn Nghiệm"></li>
    <li title="Nguyễn Nghiệm"></li>
</ul>
<h4>Append, Prepend Classes</h4>
<ul>
    <li class="btn btn-danger"></li>
    <li class="btn btn-danger"></li>
</ul>
```

```
Student bean = Student.builder().fullname("Nguyễn Nghiệm").mark(9.8).build();
model.addAttribute("student", bean);
```

```
<h4>Student Info</h4>
<ul th:object="${student}">
    <li th:text="*{fullname}"></li>
    <li th:text="*{mark}"></li>
</ul>
<ul th:with="name=${student.fullname}, mark=${student.mark}">
    <li th:text="${name}"></li>
    <li th:text="${mark}"></li>
</ul>
<th:block th:if="${student.mark >= 9.5}">
    Congratulations, you are an <b>FPT Polytechnic Golden Bea!</b>
</th:block>
```

```
<h4>Student Info</h4>
<ul>
    <li>Nguyễn Nghiệm</li>
    <li>9.8</li>
</ul>
<ul>
    <li>Nguyễn Nghiệm</li>
    <li>9.8</li>
</ul>

    Congratulations, you are an <b>FPT Polytechnic Golden Bea!</b>
```



```
Student bean = Student.builder().fullname("Nguyễn Nghiệm").gender(true).mark(9.8).build();
model.addAttribute("student", bean);
```



```
<h4>Student Info</h4>
<ul th:object="$student">
    <li th:text="*{fullname}"></li>
    <li th:text="*{mark}"></li>
    <th:block th:switch="*{gender}">
        <li th:case="true">Male</li>
        <li th:case="*">Female</li>
    </th:block>
    <li th:if="*{mark lt 5}">Bad Student</li>
    <li th:if="*{mark ge 5 and mark lt 9.5}">Good Student</li>
    <li th:unless="*{mark lt 9.5}">Golden Bea</li>
</ul>
```

```
<h4>Student Info</h4>
<ul>
    <li>Nguyễn Nghiệm</li>
    <li>9.8</li>
    <li>Male</li>
    <li>Golden Bea</li>
</ul>
```

FOLLOW CONTROL DIRECTIVES – TH: EACH

```
List<Student> students = List.of(
    Student.builder().fullname("Student 1").gender(true).mark(9.8).build(),
    Student.builder().fullname("Student 2").gender(false).mark(7).build(),
    Student.builder().fullname("Student 3").gender(true).mark(4).build(),
    Student.builder().fullname("Student 4").gender(true).mark(6).build(),
    Student.builder().fullname("Student 5").gender(false).mark(8).build());
model.addAttribute("list", students);
```

```
<h4>Student List</h4>
<th:block th:each="student : ${list}">
    <ul th:object="${student}">
        <li th:text="*{fullname}"></li>
        <li th:text="*{mark}"></li>
        <li th:text="*{gender ? 'Male' : 'Female'}"></li>
        <li th:if="*{mark ge 9.5}">Golden Bea</li>
    </ul>
</th:block>
```

Student List

- Student 1
- 9.8
- Male
- Golden Bea
- Student 2
- 7.0
- Female
- Student 3
- 4.0
- Male
- Student 4
- 6.0
- Male
- Student 5
- 8.0
- Female

FOLLOW CONTROL DIRECTIVES – TH:EACH ITEM STATUS INFO

```

<table border="1" style="width:100%">
  <thead>...</thead>
  <tbody>
    <tr th:each="student, status : ${list}">
      <td th:text="${student.fullname}"></td>
      <td th:text="${status.index}"></td>
      <td th:text="${status.count}"></td>
      <td th:text="${status.first}"></td>
      <td th:text="${status.last}"></td>
      <td th:text="${status.odd}">
      <td th:text="${status.even}">
    </tr>
  </tbody>
</table>
  
```

Fullname	Item Status Info					
	Index	Count	First	Last	Odd	Even
Student 1	0	1	true	false	true	false
Student 2	1	2	false	false	false	true
Student 3	2	3	false	false	true	false
Student 4	3	4	false	false	false	true
Student 5	4	5	false	true	true	false

```
<h2>Fill to Select Box</h2>
```

```
<select>
```

```
  <th:block th:each="sv, st : ${list}">
```

```
    <option th:value="${st.index}" th:text="${sv.fullname}"/>
```

```
  </th:block>
```

```
</select>
```

```
<h2>Fill to Select Box</h2>
```

```
<select>
```

```
  <option value="0">Student 1</option>
```

```
  <option value="1">Student 2</option>
```

```
  <option value="2">Student 3</option>
```

```
  <option value="3">Student 4</option>
```

```
  <option value="4">Student 5</option>
```

```
</select>
```

Fill to Select Box

Student 1 ▼

```
<h2>Create Radio buttons</h2>
```

```
<div>
```

```
  <th:block th:each="sv, st : ${list}">
```

```
    <input type="radio" name="student" th:value="${st.index}">
```

```
    <label th:text="${sv.fullname}"></label>
```

```
  </th:block>
```

```
</div>
```

```
<h2>Create Radio buttons</h2>
```

```
<div>
```

```
  <input type="radio" name="student" value="0">  
  <label>Student 1</label>
```

```
  <input type="radio" name="student" value="1">  
  <label>Student 2</label>
```

Create Radio buttons

☐ Student 1 ☐ Student 2 ☐ Student 3 ☐ Student 4 ☐ Student 5

✓ PHẦN I: CƠ BẢN

- ✓ GIỚI THIỆU THYMELEAF
- ✓ BIỂU THỨC THYMELEAF (EXPRESSION)
- ✓ ĐỊNH DẠNG DỮ LIỆU (FORMATING DATA)
- ✓ TOÁN TỬ (OPERATORS)

✓ PHẦN II: CHỈ THỊ THYMELEAF (DIRECTIVES)

- ✓ CHỈ THỊ RENDER NỘI DUNG
- ✓ CHỈ THỊ RENDER THUỘC TÍNH
- ✓ CHỈ THỊ HỖ TRỢ
- ✓ CHỈ THỊ PHÂN LƯỒNG (FOLLOW CONTROL)





FPT Education

FPT POLYTECHNIC

Thank you