



# SelectOption Class

A `SelectOption` object specifies one of the possible values for a Visualforce `selectCheckboxes`, `selectList`, or `selectRadio` component.

## Namespace

### System

`SelectOption` consists of a label that is displayed to the end user, and a value that is returned to the controller if the option is selected. A `SelectOption` can also be displayed in a disabled state, so that a user cannot select it as an option, but can still view it.

## Instantiation

In a custom controller or controller extension, you can instantiate a `SelectOption` in one of the following ways:

- ```
SelectOption option = new SelectOption(value, label, isDisabled);
```

where *value* is the `String` that is returned to the controller if the option is selected by a user, *label* is the `String` that is displayed to the user as the option choice, and *isDisabled* is a `Boolean` that, if true, specifies that the user cannot select the option, but can still view it.

- ```
SelectOption option = new SelectOption(value, label);
```

where *value* is the `String` that is returned to the controller if the option is selected by a user, and *label* is the `String` that is displayed to the user as the option choice. Because a value for *isDisabled* is not specified, the user can both view and select the option.

## Example

The following example shows how a list of `SelectOptions` objects can be used to provide possible values for a `selectCheckboxes` component on a Visualforce page. In the following custom controller, the `getItems` method defines and returns the list of possible `SelectOption` objects:

```
public class sampleCon {  
  
    String[] countries = new String[]{};  
  
    public PageReference test() {  
        return null;  
    }  
  
    public List<SelectOption> getItems() {  
        List<SelectOption> options = new List<SelectOption>();  
        options.add(new SelectOption('US', 'US'));  
        options.add(new SelectOption('CANADA', 'Canada'));  
        options.add(new SelectOption('MEXICO', 'Mexico'));  
    }  
}
```



```

public void setCountries(String[] countries) {
    this.countries = countries;
}
}

```

In the following page markup, the `<apex:selectOptions>` tag uses the `getItems` method from the controller above to retrieve the list of possible values. Because `<apex:selectOptions>` is a child of the `<apex:selectCheckboxes>` tag, the options are displayed as checkboxes:

```

<apex:page controller="sampleCon">
    <apex:form>
        <apex:selectCheckboxes value="{!countries}">
            <apex:selectOptions value="{!items}" />
        </apex:selectCheckboxes><br/>
        <apex:commandButton value="Test" action="{!test}" rerender="out" status="status" />
    </apex:form>
    <apex:outputPanel id="out">
        <apex:actionstatus id="status" startText="testing...">
            <apex:facet name="stop">
                <apex:outputPanel>
                    <p>You have selected:</p>
                    <apex:dataList value="{!countries}" var="c">{!c}</apex:dataList>
                </apex:outputPanel>
            </apex:facet>
        </apex:actionstatus>
    </apex:outputPanel>
</apex:page>

```

- [SelectOption Constructors](#)
- [SelectOption Methods](#)

## SelectOption Constructors

The following are constructors for `SelectOption`.

- [SelectOption\(value, label\)](#)  
Creates a new instance of the `SelectOption` class using the specified value and label.
- [SelectOption\(value, label, isDisabled\)](#)  
Creates a new instance of the `SelectOption` class using the specified value, label, and disabled setting.

### SelectOption(value, label)

Creates a new instance of the `SelectOption` class using the specified value and label.

#### Signature

```
public SelectOption(String value, String label)
```

#### Parameters

##### *value*

Type: [String](#)

The string that is returned to the Visualforce controller if the option is selected by a user.

##### *label*

Type: [String](#)



Creates a new instance of the `SelectOption` class using the specified value, label, and disabled setting.

### Signature

```
public SelectOption(String value, String label, Boolean isDisabled)
```

### Parameters

#### *value*

Type: [String](#)

The string that is returned to the Visualforce controller if the option is selected by a user.

#### *label*

Type: [String](#)

The string that is displayed to the user as the option choice.

#### *isDisabled*

Type: [Boolean](#)

If set to true, the option can't be selected by the user but can still be viewed.

## SelectOption Methods

The following are methods for `SelectOption`. All are instance methods.

- [getDisabled\(\)](#)  
Returns the current value of the `SelectOption` object's `isDisabled` attribute.
- [getEscapeItem\(\)](#)  
Returns the current value of the `SelectOption` object's `itemEscaped` attribute.
- [getLabel\(\)](#)  
Returns the option label that is displayed to the user.
- [getValue\(\)](#)  
Returns the option value that is returned to the controller if a user selects the option.
- [setDisabled\(isDisabled\)](#)  
Sets the value of the `SelectOption` object's `isDisabled` attribute.
- [setEscapeItem\(itemsEscaped\)](#)  
Sets the value of the `SelectOption` object's `itemEscaped` attribute.
- [setLabel\(label\)](#)  
Sets the value of the option label that is displayed to the user.
- [setValue\(value\)](#)  
Sets the value of the option value that is returned to the controller if a user selects the option.

### getDisabled()

Returns the current value of the `SelectOption` object's `isDisabled` attribute.

### Signature

```
public Boolean getDisabled()
```

### Return Value

Type: [Boolean](#)

### Usage



Returns the current value of the SelectOption object's `itemEscaped` attribute.

#### Signature

```
public Boolean getEscapeItem()
```

#### Return Value

Type: [Boolean](#)

#### Usage

If `itemEscaped` is set to `true`, sensitive HTML and XML characters are escaped in the HTML output generated by this component. If `itemEscaped` is set to `false`, items are rendered as written.

### getLabel()

Returns the option label that is displayed to the user.

#### Signature

```
public String getLabel()
```

#### Return Value

Type: [String](#)

### getValue()

Returns the option value that is returned to the controller if a user selects the option.

#### Signature

```
public String getValue()
```

#### Return Value

Type: [String](#)

### setDisabled(isDisabled)

Sets the value of the SelectOption object's `isDisabled` attribute.

#### Signature

```
public Void setDisabled(Boolean isDisabled)
```

#### Parameters

##### *isDisabled*

Type: [Boolean](#)

#### Return Value

Type: Void

#### Usage

If `isDisabled` is set to `true`, the user can view the option, but cannot select it. If `isDisabled` is set to `false`, the user can both view and select the option.



```
public Void setEscapeItem(Boolean itemsEscaped)
```

#### Parameters

##### *itemsEscaped*

Type: [Boolean](#)

#### Return Value

Type: Void

#### Usage

If `itemEscaped` is set to `true`, sensitive HTML and XML characters are escaped in the HTML output generated by this component. If `itemEscaped` is set to `false`, items are rendered as written.

## setLabel(label)

Sets the value of the option label that is displayed to the user.

#### Signature

```
public Void setLabel(String label)
```

#### Parameters

##### *label*

Type: [String](#)

#### Return Value

Type: Void

## setValue(value)

Sets the value of the option value that is returned to the controller if a user selects the option.

#### Signature

```
public Void setValue(String value)
```

#### Parameters

##### *value*

Type: [String](#)

#### Return Value

Type: Void

#### DID THIS ARTICLE SOLVE YOUR ISSUE?

Let us know so we can improve!

[Share your feedback](#)



MuleSoft  
Tableau  
Commerce Cloud  
Lightning Design System  
Einstein  
Quip

Component Library  
APIs  
Trailhead  
Sample Apps  
Podcasts  
AppExchange

Events and Calendar  
Partner Community  
Blog  
Salesforce Admins  
Salesforce Architects

© Copyright 2025 Salesforce, Inc. [All rights reserved.](#) Various trademarks held by their respective owners. Salesforce, Inc.  
Salesforce Tower, 415 Mission Street, 3rd Floor, San Francisco, CA 94105, United States

[Privacy Information](#) [Terms of Service](#) [Legal](#) [Use of Cookies](#) [Trust](#) [Cookie Preferences](#)

[Your Privacy Choices](#) [Responsible Disclosure](#) [Contact](#)