

# select

Please make sure you have read the [Tag Syntax](#) document and understand how tag attribute syntax works.

## Description

Render an HTML input tag of type select.

## Parameters

### Dynamic Attributes Allowed:

true

### Name Required Default Evaluated Type Description

accesskey	false	false	String	Set the html accesskey attribute on rendered html element
class	false	false	String	The css class to use for element - it's an alias of cssClass attribute.
cssClass	false	false	String	The css class to use for element
cssErrorClass	false	false	String	The css error class to use for element
cssErrorStyle	false	false	String	The css error style definitions for element to use
cssStyle	false	false	String	The css style definitions for element to use
disabled	false	false	String	Set the html disabled attribute on rendered html element
emptyOption	false	false	Boolean	Whether or not to add an empty (--) option after the header option
errorPosition	false	false	String	Define error position of form element (top bottom)
headerKey	false	false	String	Key for first item in list. Must not be empty! '-1' and '' is correct, '' is bad.
headerValue	false	false	String	Value expression for first item in list
id	false	false	String	HTML id attribute
javascriptTooltip	false	false	Boolean	Use JavaScript to generate tooltips
key	false	false	String	Set the key (name, value, label) for this particular component
label	false	false	String	Label expression used for rendering an element specific label
labelSeparator	false	false	String	String that will be appended to the label
labelPosition	false	false	String	Define label position of form element (top/left)
list	true	false	String	Iterable source to populate from. If the list is a Map (key, value), the Map key will become the option 'value' parameter and the Map value will become the option body.
listClass	false	false	String	Property of list objects to get css class from
listCssStyle	false	false	String	Property of list objects to get css style from
listKey	false	false	String	Property of list objects to get field value from
listLabelKey	false	false	String	Property of list objects to be used to lookup for localised version of field label
listTitle	false	false	String	Property of list objects to get title from
listValue	false	false	String	Property of list objects to get field content from
listValueKey	false	false	String	Property of list objects to get field value label from
multiple	false	false	Boolean	Creates a multiple select. The tag will pre-select multiple values if the values are passed as an Array or a Collection(of appropriate types) via the value attribute. If one of the keys equals one of the values in the Collection or Array it will be selected
name	false	false	String	The name to set for element
onblur	false	false	String	Set the html onblur attribute on rendered html element
onchange	false	false	String	Set the html onchange attribute on rendered html element
onclick	false	false	String	Set the html onclick attribute on rendered html element
ondblclick	false	false	String	Set the html ondblclick attribute on rendered html element
onfocus	false	false	String	Set the html onfocus attribute on rendered html element
onkeydown	false	false	String	Set the html onkeydown attribute on rendered html element

onkeypress	false	false	String	Set the html onkeypress attribute on rendered html element
onkeyup	false	false	String	Set the html onkeyup attribute on rendered html element
onmousedown	false	false	String	Set the html onmousedown attribute on rendered html element
onmousemove	false	false	String	Set the html onmousemove attribute on rendered html element
onmouseout	false	false	String	Set the html onmouseout attribute on rendered html element
onmouseover	false	false	String	Set the html onmouseover attribute on rendered html element
onmouseup	false	false	String	Set the html onmouseup attribute on rendered html element
onselect	false	false	String	Set the html onselect attribute on rendered html element
required	false	false	Boolean	If set to true, the rendered element will indicate that input is required
requiredPosition	false	false	String	Define required position of required form element (left right)
size	false	false	Integer	Size of the element box (# of elements to show)
style	false	false	String	The css style definitions for element to use - it's an alias of cssStyle attribute.
tabindex	false	false	String	Set the html tabindex attribute on rendered html element
template	false	false	String	The template (other than default) to use for rendering the element
templateDir	false	false	String	The template directory.
theme	false	false	String	The theme (other than default) to use for rendering the element
title	false	false	String	Set the html title attribute on rendered html element
tooltip	false	false	String	Set the tooltip of this particular component
tooltipConfig	false	false	String	Deprecated. Use individual tooltip configuration attributes instead.
tooltipClass	StrutsTT	false	String	CSS class applied to JavaScript tooltips
tooltipDelay	Classic	false	String	Delay in milliseconds, before showing JavaScript tooltips
tooltipIconPath	Classic	false	String	Icon path used for image that will have the tooltip
value	false	false	String	Preset the value of input element.

## Examples

Error rendering macro 'code': Invalid value specified for parameter 'java.lang.NullPointerException'

Note: For any of the tags that use lists (select probably being the most ubiquitous), which uses the OGNL list notation (see the "months" example above), it should be noted that the map key created (in the months example, the '01', '02', etc.) is typed. '1' is a char, '01' is a String, "1" is a String. This is important since if the value returned by your "value" attribute is NOT the same type as the key in the "list" attribute, they WILL NOT MATCH, even though their String values may be equivalent. If they don't match, nothing in your list will be auto-selected.

Error rendering macro 'code': Invalid value specified for parameter 'java.lang.NullPointerException'

```
<s:select label="Pets"
  name="petIds"
  list="petDao.pets"
  listKey="id"
  listValue="name"
  multiple="true"
  size="3"
  required="true"
  value="{petDao.pets.{id}}"
/>

<s:select label="Months"
  name="months"
  headerKey="-1" headerValue="Select Month"
  list="#{'01':'Jan', '02':'Feb', [...]}"
  value="selectedMonth"
  required="true"
/>

// The month id (01, 02, ...) returned by the getSelectedMonth() call
// against the stack will be auto-selected
```