

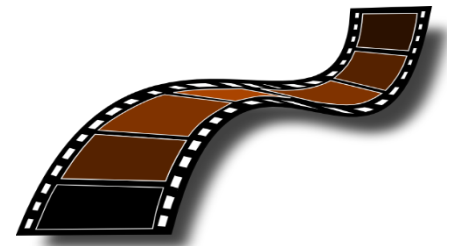
# Custom Product Selectors



Paul D. Sheriff

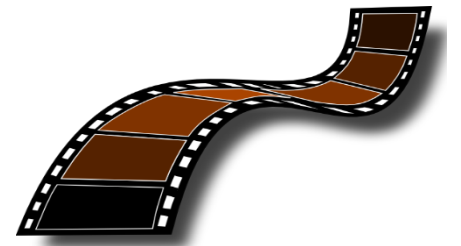
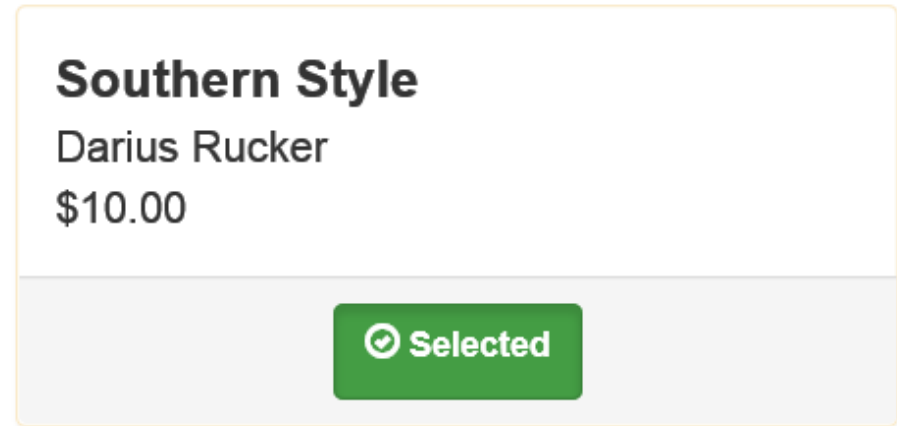
[Psheriff@pdsa.com](mailto:Psheriff@pdsa.com), [www.pdsa.com](http://www.pdsa.com), @pdsainc

# Reasons to Watch This Module



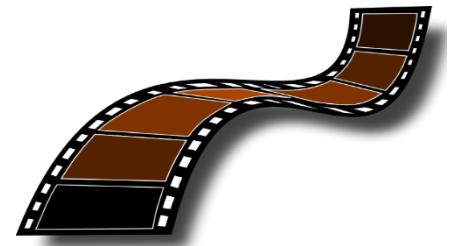
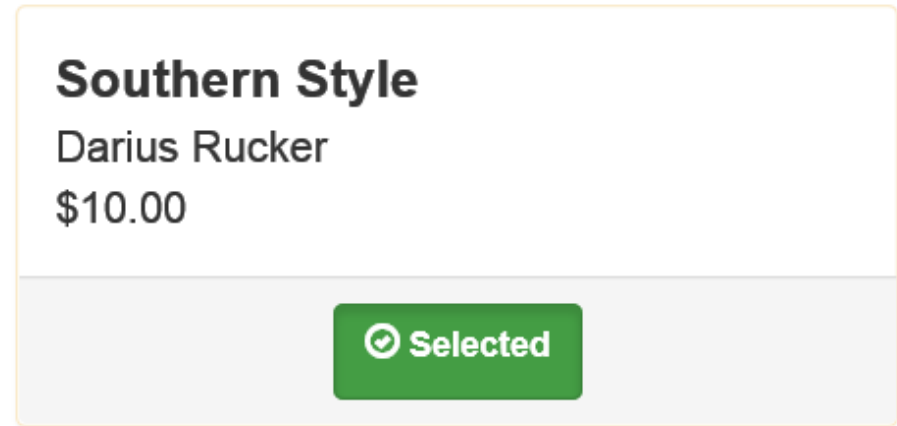
# Reasons to Watch This Module

- Build custom product selection boxes



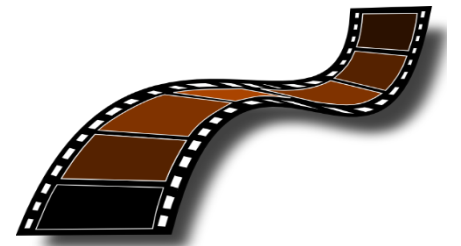
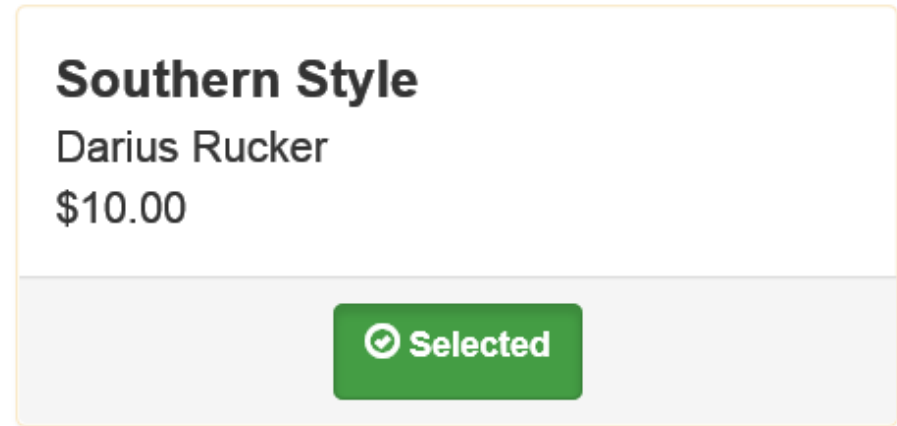
# Reasons to Watch This Module

- Build custom product selection boxes
  - Use Bootstrap panel classes



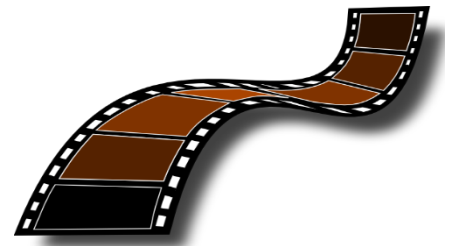
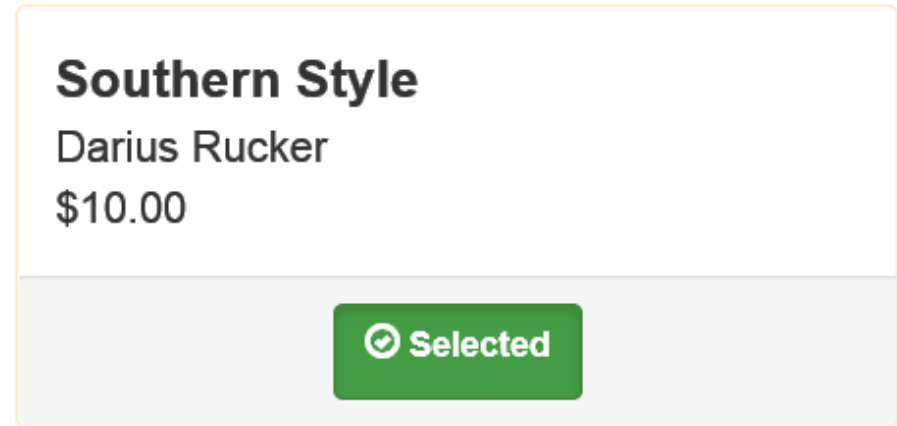
# Reasons to Watch This Module

- Build custom product selection boxes
  - Use Bootstrap panel classes
  - Add custom styles



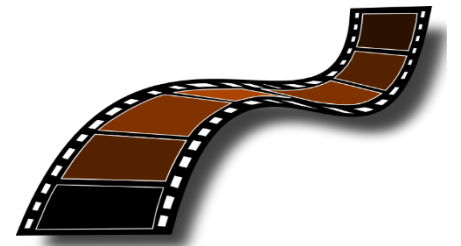
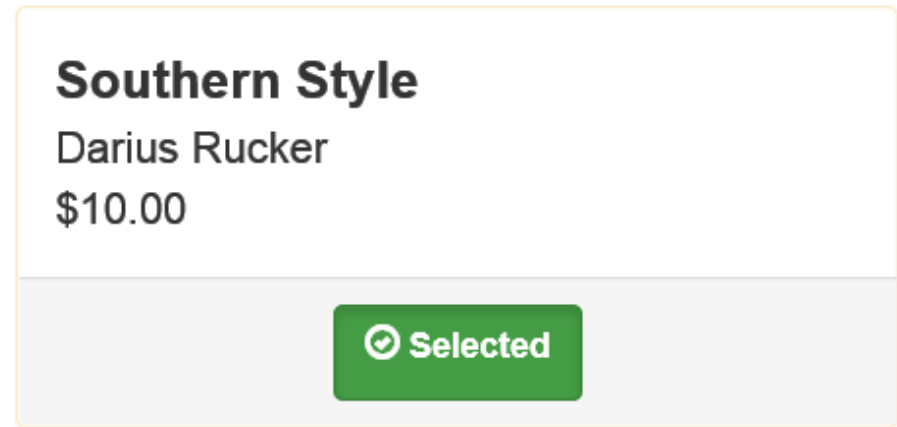
# Reasons to Watch This Module

- Build custom product selection boxes
  - Use Bootstrap panel classes
  - Add custom styles
- Add check boxes in buttons



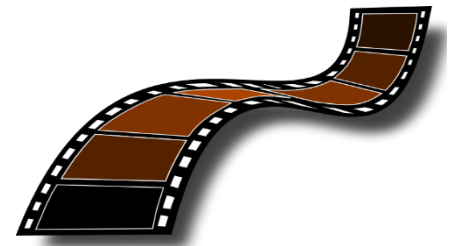
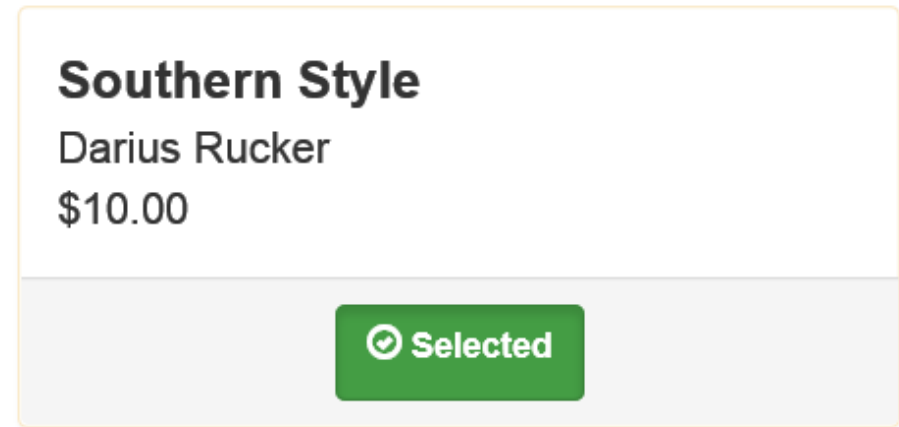
# Reasons to Watch This Module

- Build custom product selection boxes
  - Use Bootstrap panel classes
  - Add custom styles
- Add check boxes in buttons
- Use jQuery to calculate total



# Reasons to Watch This Module

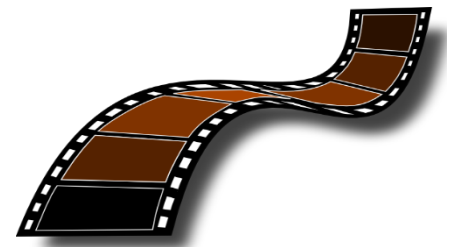
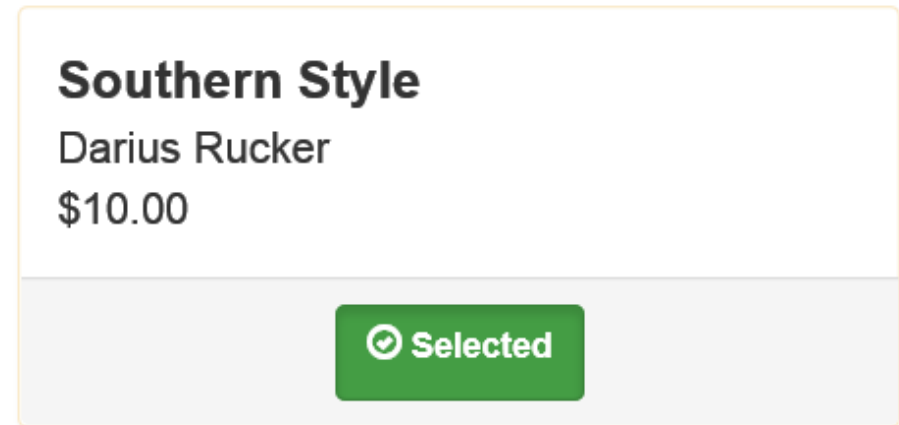
- Build custom product selection boxes
  - Use Bootstrap panel classes
  - Add custom styles
- Add check boxes in buttons
- Use jQuery to calculate total
- Use JavaScript objects for generic code





# Reasons to Watch This Module

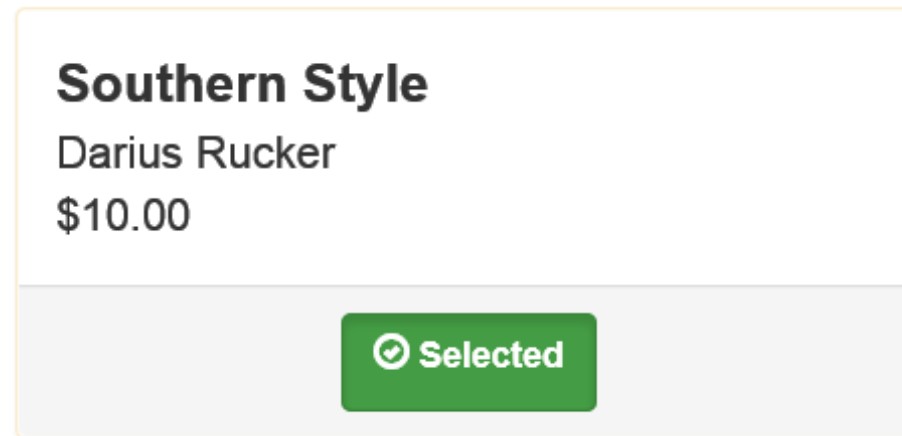
- Build custom product selection boxes
  - Use Bootstrap panel classes
  - Add custom styles
- Add check boxes in buttons
- Use jQuery to calculate total
- Use JavaScript objects for generic code
- Let's get started...



# Use Panel Classes

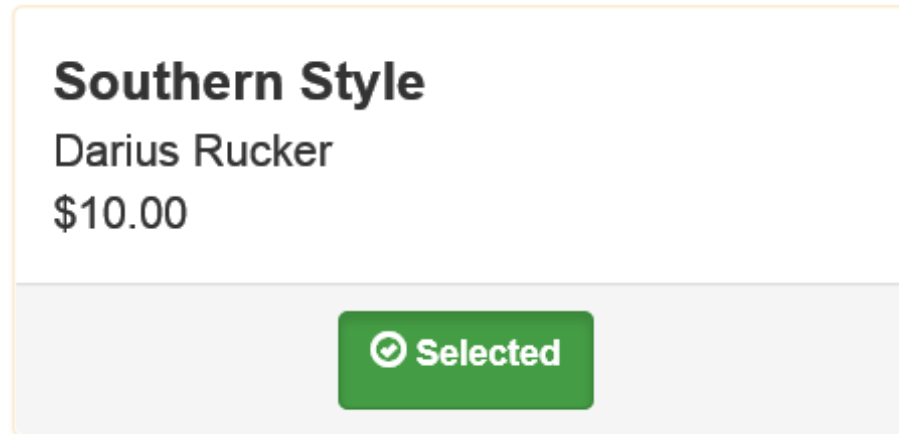
# Use Panel Classes

- Start with built-in bootstrap panel classes



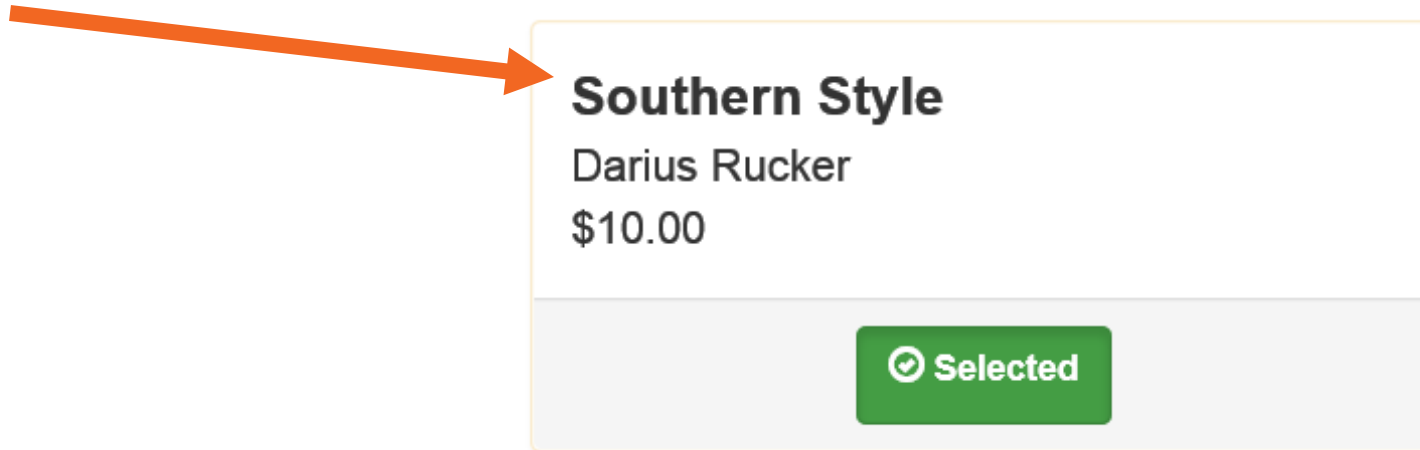
# Use Panel Classes

- Start with built-in bootstrap panel classes
- Add styles for each area of the selector



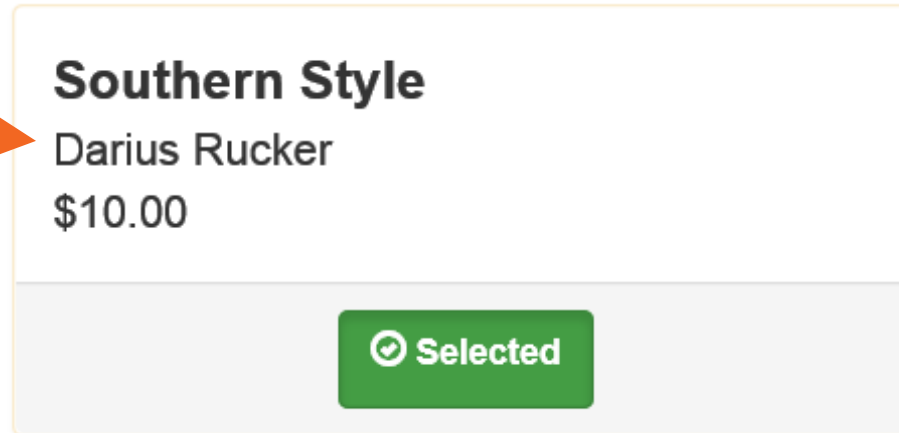
# Use Panel Classes

- Start with built-in bootstrap panel classes
- Add styles for each area of the selector
  - title



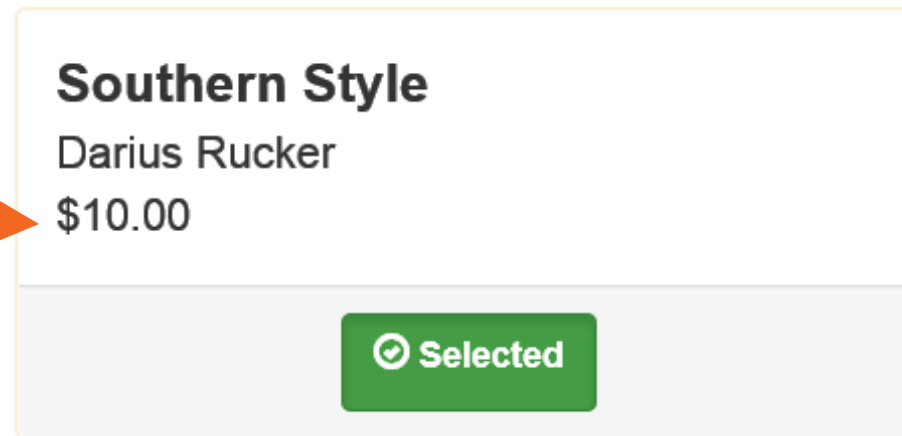
# Use Panel Classes

- Start with built-in bootstrap panel classes
- Add styles for each area of the selector
  - title
  - desc



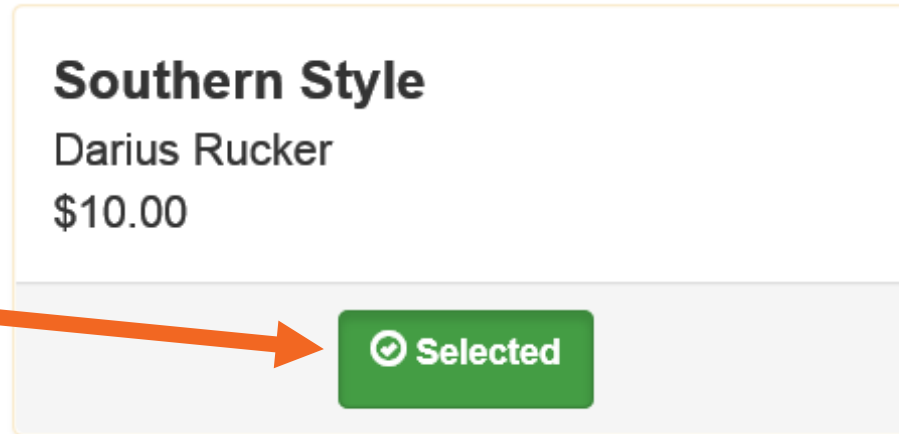
# Use Panel Classes

- Start with built-in bootstrap panel classes
- Add styles for each area of the selector
  - title
  - desc
  - price



# Use Panel Classes

- Start with built-in bootstrap panel classes
- Add styles for each area of the selector
  - title
  - desc
  - price
- Button group
  - In panel-footer





# Use Panel Classes

# Use Panel Classes

- Button group uses check box covered in last module

```
<div class="btn-group" data-toggle="buttons">  
  <label class="btn btn-success active">  
    <span class="glyphicon glyphicon-ok-circle"></span>  
    <input id="chk1" type="checkbox" checked="checked" />  
    <label for="chk1">Selected</label>  
  </label>  
</div>
```

# Use Panel Classes

- Button group uses check box covered in last module

```
<div class="btn-group" data-toggle="buttons">  
  <label class="btn btn-success active">  
    <span class="glyphicon glyphicon-ok-circle"></span>  
    <input id="chk1" type="checkbox" checked="checked" />  
    <label for="chk1">Selected</label>  
  </label>  
</div>
```

# Use Panel Classes

# Use Panel Classes

- jQuery is used to toggle buttons and change text

```
if ($(this).prop('checked')) {  
    $(this).prev().removeClass('glyphicon-unchecked')  
        .addClass('glyphicon-ok-circle');  
    $(this).parent().removeClass('btn-primary')  
        .addClass('btn-success');  
    $(this).next().text("Selected");  
} else ...
```

# Use Panel Classes

- jQuery is used to toggle buttons and change text
- Respond to change event on check boxes

```
if ($(this).prop('checked')) {  
    $(this).prev().removeClass('glyphicon-unchecked')  
        .addClass('glyphicon-ok-circle');  
    $(this).parent().removeClass('btn-primary')  
        .addClass('btn-success');  
    $(this).next().text("Selected");  
} else ...
```

# Use Panel Classes

- jQuery is used to toggle buttons and change text
- Respond to change event on check boxes
- Toggle glyphs and text based on checked/unchecked

```
if ($(this).prop('checked')) {  
    $(this).prev().removeClass('glyphicon-unchecked')  
        .addClass('glyphicon-ok-circle');  
    $(this).parent().removeClass('btn-primary')  
        .addClass('btn-success');  
    $(this).next().text("Selected");  
} else ...
```

# Use Panel Classes

- jQuery is used to toggle buttons and change text
- Respond to change event on check boxes
- Toggle glyphs and text based on checked/unchecked

```
if ($(this).prop('checked')) {  
    $(this).prev().removeClass('glyphicon-unchecked')  
                .addClass('glyphicon-ok-circle');  
    $(this).parent().removeClass('btn-primary')  
                    .addClass('btn-success');  
    $(this).next().text("Selected");  
} else ...
```



# Use Panel Classes

- jQuery is used to toggle buttons and change text
- Respond to change event on check boxes
- Toggle glyphs and text based on checked/unchecked

```
if ($(this).prop('checked')) {  
    $(this).prev().removeClass('glyphicon-unchecked')  
        .addClass('glyphicon-ok-circle');  
    $(this).parent().removeClass('btn-primary')  
        .addClass('btn-success');  
    $(this).next().text("Selected");  
} else ...
```

A single product selector



# Generic Check Boxes

# Generic Check Boxes

- Too much hard-coding in last sample

# Generic Check Boxes

- Too much hard-coding in last sample
- We need to remove

# Generic Check Boxes

- Too much hard-coding in last sample
- We need to remove
  - 'glyphicon'

# Generic Check Boxes

- Too much hard-coding in last sample
- We need to remove
  - 'glyphicon'
  - 'btn-success'

# Generic Check Boxes

- Too much hard-coding in last sample
- We need to remove
  - 'glyphicon'
  - 'btn-success'
  - 'btn-primary'



# Generic Check Boxes

- Too much hard-coding in last sample
- We need to remove
  - 'glyphicon'
  - 'btn-success'
  - 'btn-primary'
  - 'Selected'

# Generic Check Boxes

- Too much hard-coding in last sample
- We need to remove
  - 'glyphicon'
  - 'btn-success'
  - 'btn-primary'
  - 'Selected'
  - 'Not Selected'

# Generic Check Boxes

# Generic Check Boxes

- Create JavaScript object for check box options

```
var checkOptions = {  
  id: "",  
  checkedGlyph: "glyphicon-ok-circle",  
  uncheckedGlyph: "glyphicon-unchecked",  
  checkedBtnClass: "btn-success",  
  uncheckedBtnClass: "btn-primary",  
  checkedText: "Selected",  
  uncheckedText: "Not Selected"  
};
```

# Generic Check Boxes

- Create JavaScript object for check box options
- Avoids hard coding

```
var checkOptions = {  
  id: "",  
  checkedGlyph: "glyphicon-ok-circle",  
  uncheckedGlyph: "glyphicon-unchecked",  
  checkedBtnClass: "btn-success",  
  uncheckedBtnClass: "btn-primary",  
  checkedText: "Selected",  
  uncheckedText: "Not Selected"  
};
```

# Generic Check Boxes

- Create JavaScript object for check box options
- Avoids hard coding
- Notice the property names

```
var checkOptions = {  
  id: "",  
  checkedGlyph: "glyphicon-ok-circle",  
  uncheckedGlyph: "glyphicon-unchecked",  
  checkedBtnClass: "btn-success",  
  uncheckedBtnClass: "btn-primary",  
  checkedText: "Selected",  
  uncheckedText: "Not Selected"  
};
```

# Generic Check Boxes

- Create JavaScript object for check box options
- Avoids hard coding
- Notice the property names
- Notice the default values

```
var checkOptions = {  
  id: "",  
  checkedGlyph: "glyphicon-ok-circle",  
  uncheckedGlyph: "glyphicon-unchecked",  
  checkedBtnClass: "btn-success",  
  uncheckedBtnClass: "btn-primary",  
  checkedText: "Selected",  
  uncheckedText: "Not Selected"  
};
```

# Generic Check Boxes



# Generic Check Boxes

- Defaults are set as shown in previous slide

# Generic Check Boxes

- Defaults are set as shown in previous slide
- Modify checkOptions properties to change check boxes

```
$(document).ready(function () {  
    checkOptions.id = "#productSelectors";  
    checkOptions.checkedText = "In Cart";  
});
```

# Generic Check Boxes

- Defaults are set as shown in previous slide
- Modify checkOptions properties to change check boxes

```
$(document).ready(function () {  
    checkOptions.id = "#productSelectors";  
    checkOptions.checkedText = "In Cart";  
});
```

# Generic Check Boxes

# Generic Check Boxes

- Create two functions

# Generic Check Boxes

- Create two functions
  - Replace code in if statement

# Generic Check Boxes

- Create two functions
  - Replace code in if statement
- Call one of the following based on checked property

# Generic Check Boxes

- Create two functions
  - Replace code in if statement
- Call one of the following based on checked property
  - setChecked

```
function setChecked(ctl) {  
    $(ctl).prev()  
        .removeClass(checkOptions.uncheckedGlyph)  
        .addClass(checkOptions.checkedGlyph);  
    $(ctl).parent()  
        .removeClass(checkOptions.uncheckedBtnClass)  
        .addClass(checkOptions.checkedBtnClass);  
    $($ (ctl).next()).text(checkOptions.checkedText);  
}
```



# Generic Check Boxes

- Create two functions
  - Replace code in if statement
- Call one of the following based on checked property
  - setChecked

```
function setChecked(ctl) {  
    $(ctl).prev()  
        .removeClass(checkOptions.uncheckedGlyph)  
        .addClass(checkOptions.checkedGlyph);  
    $(ctl).parent()  
        .removeClass(checkOptions.uncheckedBtnClass)  
        .addClass(checkOptions.checkedBtnClass);  
    $($ (ctl).next()).text(checkOptions.checkedText);  
}
```

# Generic Check Boxes

- Create two functions
  - Replace code in if statement
- Call one of the following based on checked property
  - setChecked
  - setUnchecked

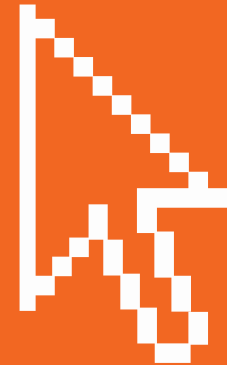
```
function setUnchecked(ctl) {  
    $(ctl).prev()  
        .removeClass(checkOptions.checkedGlyph)  
        .addClass(checkOptions.uncheckedGlyph);  
    $(ctl).parent()  
        .removeClass(checkOptions.checkedBtnClass)  
        .addClass(checkOptions.uncheckedBtnClass);  
    $($ (ctl).next()).text(checkOptions.uncheckedText);  
}
```

# Generic Check Boxes

- Create two functions
  - Replace code in if statement
- Call one of the following based on checked property
  - setChecked
  - setUnchecked
- These functions toggle the glyphs and text

```
function setUnchecked(ctl) {  
    $(ctl).prev()  
        .removeClass(checkOptions.checkedGlyph)  
        .addClass(checkOptions.uncheckedGlyph);  
    $(ctl).parent()  
        .removeClass(checkOptions.checkedBtnClass)  
        .addClass(checkOptions.uncheckedBtnClass);  
    $($ (ctl).next()).text(checkOptions.uncheckedText);  
}
```

Generic check boxes



# Calculate Totals

# Calculate Totals

- Keep running total of products selected

# Calculate Totals

- Keep running total of products selected
  - Add `<span id="total">`

Select Albums to Purchase

**Southern Style**  
Darius Rucker  
\$10.00

☒ Selected

**Tom Cat**  
Tom Scott & the L.A. Express  
\$7.00

☒ Selected

**Drunk Enough to Dance**  
Bowling for Soup  
\$9.00

☐ Not Selected

Total: \$17.00

# Calculate Totals

- Keep running total of products selected
  - Add `<span id="total">`
- Write jQuery

## Select Albums to Purchase

Total: \$17.00

### Southern Style

Darius Rucker  
\$10.00

☒ Selected

### Tom Cat

Tom Scott & the L.A. Express  
\$7.00

☒ Selected

### Drunk Enough to Dance

Bowling for Soup  
\$9.00

☐ Not Selected



# Calculate Totals

- Keep running total of products selected
  - Add `<span id="total">`
- Write jQuery
  - Connect to change event on check boxes

## Select Albums to Purchase

Total: \$17.00

### Southern Style

Darius Rucker  
\$10.00

☒ Selected

### Tom Cat

Tom Scott & the L.A. Express  
\$7.00

☒ Selected

### Drunk Enough to Dance

Bowling for Soup  
\$9.00

☐ Not Selected

# Calculate Totals

- Keep running total of products selected
  - Add `<span id="total">`
- Write jQuery
  - Connect to change event on check boxes
  - Calculate total on change

## Select Albums to Purchase

Total: \$17.00

### Southern Style

Darius Rucker  
\$10.00

☒ Selected

### Tom Cat

Tom Scott & the L.A. Express  
\$7.00

☒ Selected

### Drunk Enough to Dance

Bowling for Soup  
\$9.00

☐ Not Selected

# Calculate Totals

- Keep running total of products selected
  - Add `<span id="total">`
- Write jQuery
  - Connect to change event on check boxes
  - Calculate total on change
  - Get price in each box

## Select Albums to Purchase

Total: \$17.00

### Southern Style

Darius Rucker  
\$10.00

☒ Selected

### Tom Cat

Tom Scott & the L.A. Express  
\$7.00

☒ Selected

### Drunk Enough to Dance

Bowling for Soup  
\$9.00

☐ Not Selected

# Calculate Totals

- Keep running total of products selected
  - Add `<span id="total">`
- Write jQuery
  - Connect to change event on check boxes
  - Calculate total on change
  - Get price in each box
    - Strip currency symbols

## Select Albums to Purchase

Total: \$17.00

### Southern Style

Darius Rucker  
\$10.00

☒ Selected

### Tom Cat

Tom Scott & the L.A. Express  
\$7.00

☒ Selected

### Drunk Enough to Dance

Bowling for Soup  
\$9.00

☐ Not Selected

# Calculate Totals

- Keep running total of products selected

- Add `<span id="total">`

- Write jQuery

- Connect to change event on check boxes
- Calculate total on change
- Get price in each box
  - Strip currency symbols
  - Add/subtract from total area

## Select Albums to Purchase

Total: \$17.00

### Southern Style

Darius Rucker  
\$10.00

☒ Selected

### Tom Cat

Tom Scott & the L.A. Express  
\$7.00

☒ Selected

### Drunk Enough to Dance

Bowling for Soup  
\$9.00

☐ Not Selected

# Calculate Totals

- Keep running total of products selected
  - Add `<span id="total">`
- Write jQuery
  - Connect to change event on check boxes
  - Calculate total on change
  - Get price in each box
    - Strip currency symbols
    - Add/subtract from total area
    - Format numbers as currency

## Select Albums to Purchase

Total: \$17.00

### Southern Style

Darius Rucker  
\$10.00

☒ Selected

### Tom Cat

Tom Scott & the L.A. Express  
\$7.00

☒ Selected

### Drunk Enough to Dance

Bowling for Soup  
\$9.00

☐ Not Selected

# Calculate Totals

# Calculate Totals

- stripCurrency function

```
function stripCurrency(value, symbol, separator) {  
    symbol = (typeof symbol == 'undefined' ? '$' : symbol);  
    separator = (typeof separator == 'undefined' ? ',' : separator);  
  
    value = value.replace(symbol, "")  
                .replace(separator, "")  
                .replace(" ", "");  
    return value;  
}
```



# Calculate Totals

- stripCurrency function

Optional parameters

```
function stripCurrency(value, symbol, separator) {  
    symbol = (typeof symbol == 'undefined' ? '$' : symbol);  
    separator = (typeof separator == 'undefined' ? ',' : separator);  
  
    value = value.replace(symbol, "")  
                .replace(separator, "")  
                .replace(" ", "");  
  
    return value;  
}
```

# Calculate Totals

- stripCurrency function

Optional parameters  
(defaults to U.S. format)

```
function stripCurrency(value, symbol, separator) {  
    symbol = (typeof symbol == 'undefined' ? '$' : symbol);  
    separator = (typeof separator == 'undefined' ? ',' : separator);  
  
    value = value.replace(symbol, "")  
                .replace(separator, "")  
                .replace(" ", "");  
  
    return value;  
}
```

# Calculate Totals

- stripCurrency function

```
function stripCurrency(value, symbol, separator) {  
    symbol = (typeof symbol == 'undefined' ? '$' : symbol);  
    separator = (typeof separator == 'undefined' ? ',' : separator);  
  
    value = value.replace(symbol, "")  
              .replace(separator, "")  
              .replace(" ", "");  
  
    return value;  
}
```



The diagram consists of an orange speech bubble pointing to the `value.replace(symbol, "")` line in the code. The text inside the bubble is "Strip currency symbol".

# Calculate Totals

- stripCurrency function

```
function stripCurrency(value, symbol, separator) {  
    symbol = (typeof symbol == 'undefined' ? '$' : symbol);  
    separator = (typeof separator == 'undefined' ? ',' : separator);  
  
    value = value.replace(symbol, "")  
                .replace(separator, "")  
                .replace(" ", "");  
    return value;  
}
```

Strip separator

# Calculate Totals

- stripCurrency function

```
function stripCurrency(value, symbol, separator) {  
    symbol = (typeof symbol == 'undefined' ? '$' : symbol);  
    separator = (typeof separator == 'undefined' ? ',' : separator);  
  
    value = value.replace(symbol, "")  
                .replace(separator, "")  
                .replace(" ", "");  
    return value;  
}
```

Strip spaces

# Calculate Totals

# Calculate Totals

- formatCurrency function

```
function formatCurrency(value, decimals, decpoint, separator, symbol) {  
    decimals = (typeof decimals == 'undefined' ? 2 : decimals);  
    decpoint = (typeof decpoint == 'undefined' ? '.' : decpoint);  
    separator = (typeof separator == 'undefined' ? ',' : separator);  
    symbol = (typeof symbol == 'undefined' ? '$' : symbol);  
  
    var parts = value.toFixed(decimals).toString().split(decpoint);  
    parts[0] = parts[0].replace(/\B(?=(\d{3})+(?!\d))/g, separator);  
    return (symbol + parts.join(decpoint)).toLocaleString();  
}
```

# Calculate Totals

- formatCurrency function

Optional parameters

```
function formatCurrency(value, decimals, decpoint, separator, symbol) {  
    decimals = (typeof decimals == 'undefined' ? 2 : decimals);  
    decpoint = (typeof decpoint == 'undefined' ? '.' : decpoint);  
    separator = (typeof separator == 'undefined' ? ',' : separator);  
    symbol = (typeof symbol == 'undefined' ? '$' : symbol);  
  
    var parts = value.toFixed(decimals).toString().split(decpoint);  
    parts[0] = parts[0].replace(/\B(?=(\d{3})+(?!\d))/g, separator);  
    return (symbol + parts.join(decpoint)).toLocaleString();  
}
```



# Calculate Totals

- formatCurrency function

```
function formatCurrency(value, decimals, decpoint, separator, symbol) {
    decimals = (typeof decimals == 'undefined' ? 2 : decimals);
    decpoint = (typeof decpoint == 'undefined' ? '.' : decpoint);
    separator = (typeof separator == 'undefined' ? ',' : separator);
    symbol = (typeof symbol == 'undefined' ? '$' : symbol);

    var parts = value.toFixed(decimals).toString().split(decpoint);
    parts[0] = parts[0].replace(/\B(?=(\d{3})+(?!\d))/g, separator);
    return (symbol + parts.join(decpoint)).toLocaleString();
}
```

Optional parameters  
(defaults to U.S.  
format)

# Calculate Totals

- formatCurrency function

```
function formatCurrency(value, decimals, decpoint, separator, symbol) {  
    decimals = (typeof decimals == 'undefined' ? 2 : decimals);  
    decpoint = (typeof decpoint == 'undefined' ? '.' : decpoint);  
    separator = (typeof separator == 'undefined' ? ',' : separator);  
    symbol = (typeof symbol == 'undefined' ? '$' : symbol);  
  
    var parts = value.toFixed(decimals).toString().split(decpoint);  
    parts[0] = parts[0].replace(/\B(?=(\d{3})+(?!\d))/g, separator);  
    return (symbol + parts.join(decpoint)).toLocaleString();  
}
```

Break into array based on decimal point symbol

# Calculate Totals

- formatCurrency function

```
function formatCurrency(value, decimals, decpoint, separator, symbol) {  
    decimals = (typeof decimals == 'undefined' ? 2 : decimals);  
    decpoint = (typeof decpoint == 'undefined' ? '.' : decpoint);  
    separator = (typeof separator == 'undefined' ? ',' : separator);  
    symbol = (typeof symbol == 'undefined' ? '$' : symbol);  
  
    var parts = value.toFixed(decimals).toString().split(decpoint);  
    parts[0] = parts[0].replace(/\B(?=(\d{3})+(?!\d))/g, separator);  
    return (symbol + parts.join(decpoint)).toLocaleString();  
}
```

Create currency with  
separator character

# Calculate Totals

- formatCurrency function

```
function formatCurrency(value, decimals, decpoint, separator, symbol) {  
    decimals = (typeof decimals == 'undefined' ? 2 : decimals);  
    decpoint = (typeof decpoint == 'undefined' ? '.' : decpoint);  
    separator = (typeof separator == 'undefined' ? ',' : separator);  
    symbol = (typeof symbol == 'undefined' ? '$' : symbol);  
  
    var parts = value.toFixed(decimals).toString().split(decpoint);  
    parts[0] = parts[0].replace(/\B(?=(\d{3})+(?!\d))/g, separator);  
    return (symbol + parts.join(decpoint)).toLocaleString();  
}
```

Join currency part and  
number after decimal

Calculate total



# Final Sample

# Final Sample

- Put all CSS and jQuery into .css and .js files

# Final Sample

- Put all CSS and jQuery into .css and .js files
- Genericize total calculation function



# Final Sample

- Put all CSS and jQuery into .css and .js files
- Genericize total calculation function
  - Add JavaScript object

# Final Sample

- Put all CSS and jQuery into .css and .js files
- Genericize total calculation function
  - Add JavaScript object
  - Set defaults

```
var totalOptions = {  
  id: "#total",  
  priceClass: ".price",  
  priceContainerClass: ".panel"  
}
```

```
function calculateTotal(ctl) {  
  // Get the total amount  
  var total = $(totalOptions.id).text();  
  // Strip currency symbols and thousands separator  
  total = stripCurrency(total);  
  
  // Get the price from within this panel  
  var price = $(ctl).closest(totalOptions.priceContainerClass)  
    .find(totalOptions.priceClass).text();
```

# Final Sample

- Put all CSS and jQuery into .css and .js files
- Genericize total calculation function
  - Add JavaScript object
  - Set defaults

```
var totalOptions = {  
  id: "#total",  
  priceClass: ".price",  
  priceContainerClass: ".panel"  
}
```

```
function calculateTotal(ctl) {  
  // Get the total amount  
  var total = $(totalOptions.id).text();  
  // Strip currency symbols and thousands separator  
  total = stripCurrency(total);  
  
  // Get the price from within this panel  
  var price = $(ctl).closest(totalOptions.priceContainerClass)  
    .find(totalOptions.priceClass).text();
```

# Final Sample

- Put all CSS and jQuery into .css and .js files
- Genericize total calculation function
  - Add JavaScript object
  - Set defaults

```
var totalOptions = {  
  id: "#total",  
  priceClass: ".price",  
  priceContainerClass: ".panel"  
}
```

```
function calculateTotal(ctl) {  
  // Get the total amount  
  var total = $(totalOptions.id).text();  
  // Strip currency symbols and thousands separator  
  total = stripCurrency(total);  
  
  // Get the price from within this panel  
  var price = $(ctl).closest(totalOptions.priceContainerClass)  
    .find(totalOptions.priceClass).text();
```

# Final Sample

- Put all CSS and jQuery into .css and .js files
- Genericize total calculation function
  - Add JavaScript object
  - Set defaults

```
var totalOptions = {  
  id: "#total",  
  priceClass: ".price",  
  priceContainerClass: ".panel"  
}
```

```
function calculateTotal(ctl) {  
  // Get the total amount  
  var total = $(totalOptions.id).text();  
  // Strip currency symbols and thousands separator  
  total = stripCurrency(total);  
  
  // Get the price from within this panel  
  var price = $(ctl).closest(totalOptions.priceContainerClass)  
    .find(totalOptions.priceClass).text();
```

Final sample

Everything moved into .css and .js files



# Summary



# Summary

- Combine Bootstrap elements together to create new widgets





# Summary

- Combine Bootstrap elements together to create new widgets
- Add your own styling within new widgets



# Summary

- Combine Bootstrap elements together to create new widgets
- Add your own styling within new widgets
- Put your widgets into Bootstrap columns



# Summary

- Combine Bootstrap elements together to create new widgets
- Add your own styling within new widgets
- Put your widgets into Bootstrap columns
  - Don't have to do all the responsive coding



# Summary

- Combine Bootstrap elements together to create new widgets
- Add your own styling within new widgets
- Put your widgets into Bootstrap columns
  - Don't have to do all the responsive coding
- Create JavaScript objects to avoid hard-coding



# Summary

- Combine Bootstrap elements together to create new widgets
- Add your own styling within new widgets
- Put your widgets into Bootstrap columns
  - Don't have to do all the responsive coding
- Create JavaScript objects to avoid hard-coding
- Put jQuery into .js files for reusability



# Summary

- Combine Bootstrap elements together to create new widgets
- Add your own styling within new widgets
- Put your widgets into Bootstrap columns
  - Don't have to do all the responsive coding
- Create JavaScript objects to avoid hard-coding
- Put jQuery into .js files for reusability
- Put CSS into .css files for reusability

