17. Working with interactive components

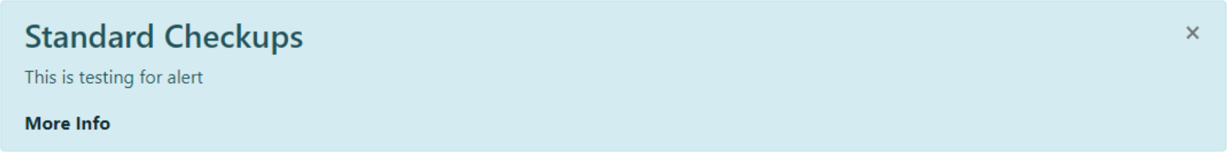
Setting alerts

Alerts are messages with special contextual styles that are designed to display in great looking boxes. They're optionally dismissed and can have any sort of markup so let's take a look at them. Now setting up alerts is pretty simply. You start out with the basic alert class on a container. There's a few contextual classes that you can use for color. So for example, success, info, warning, and danger. A few classes are available for internal content, like headings as well as links.

Although you can fit just about any HTML content inside an alert. Now just be aware that it is really meant for very simple content, so don't try too many crazy HTML tags in there. You can choose to dismiss the alerts so that the user can get rid of message. This is how you do that. You add a class of **`alert-dismissible`** and then if you want some animation you can also add the fade and show classes. So when you dismiss the alert, it'll have a little animation that fades out.

When you do this, you should also add a close button to the alert to close the alert when the click on the close button.

Example) Create an alert message as the following



Standard Checkups

This is testing for alert

More Info

HTML

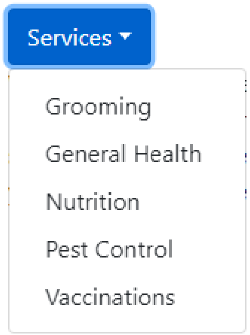
```
<div class='alert alert-info'>
  <button type='button' class='close' data-dismiss='alert'><span>&times;</span>
  </button>
  <h3 class='alert-heading'>Standard Checkups</h3>
  <p> This is testing for alert</p>
  <a class='alert-link' href="#">More Info</a>
</div>
```

Dropdown

Dropdowns are a common design patterns in Bootstrap. You can create them easily for different components, so let's take a look. You can use **`dropdown`** in a bunch of different components including `navs`, `tabs`, and `buttons`. Dropdowns can be created in two parts. First, you create a button to trigger the **`dropdown`** and then create the **`menu`** itself. They're tied under our dropdown container that toggles the dropdown menu.

For the menu, you can either use anchor tag links or buttons. The classes for a basic dropdown are the **`dropdown`** class on the container of both the trigger and the menu element, the **`dropdown-toggle`** class on the button that triggers the menu, and then the **`dropdown-menu`** class which is the container for the menu. You also add a **`dropdown-item`** on each individual menu item which is either a link or a button.

Example) Create a dropdown list for **Services**

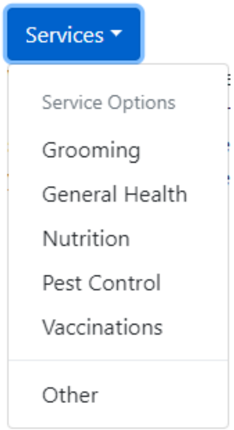


HTML

```
<div class='dropdown text-dark'>
<button type="button" class='btn btn-primary dropdown-toggle ' data-
toggle='dropdown'>Services</button>
    <div class='dropdown-menu'>
      <a class='dropdown-item' href="#">Grooming</a>
      <a class='dropdown-item' href="#">General Health</a>
      <a class='dropdown-item' href="#">Nutrition</a>
      <a class='dropdown-item' href="#">Pest Control</a>
      <a class='dropdown-item' href="#">Vaccinations</a>
    </div>
</div>
```

Inside the dropdown, you can add a few elements including **`dropdown-header`** that's a nice title to a group of elements in the dropdown, you can also you use **`dropdown-divider`** that lets you create a horizontal line to divide groups of elements, any menu item can also be tagged as disabled to prevent people from clicking on it.

Example) Create a dropdown list for **Services** with header and divider within the list



HTML

```
<div class='dropdown text-dark'>
<button type="button" class='btn btn-primary dropdown-toggle ' data-
toggle='dropdown'>Services</button>
    <div class='dropdown-menu' aria-labelledby='dropdownMenuButton'>
        <div class='dropdown-header'>Service Options</div>
            <a class='dropdown-item' href="#">Grooming</a>
            <a class='dropdown-item' href="#">General Health</a>
            <a class='dropdown-item' href="#">Nutrition</a>
            <a class='dropdown-item' href="#">Pest Control</a>
            <a class='dropdown-item' href="#">Vaccinations</a>
        <div class='dropdown-divider'></div>
            <a class='dropdown-item' href='#'>Other</a>
    </div>
</div>
```

There's a few options that control how things look and work. For example, you can control the size of the buttons using the **btn-sm** for a smaller button and the **btn-lg** for a bigger button that goes in the trigger of the dropdown menu. You can also cause the button to drop upwards instead of downwards with the **dropup** class. If you want to align the menu to the right instead of to the left which is a default, you can add a **dropdown-menu-right** class. Finally, you can create a menu elements where the dropdown and button are split, **btn-group dropdown-toggle-split**. That's a little bit more flexible than just having a dropdown since it allows you to choose an option before you submit the button.

Scrollspy

Scrollspy is a really cool component that allows you to keep track of the scroll of the page and modifies classes according to the position of your elements. So the way that you use it is by creating a data attribute called **`scroll`**, and you do that on the container of the element that you want to track the scrolling in. Now that element also needs to have a position of relative, and you also need to target that element with a data target attribute, and then an **`id`**. Data target is usually how we put together what we are doing with the object that we're targeting.

HTML

```
<body data-spy='scroll' data-target='#navbar-site' data-offset='80'>
<nav id="navbar-site" class="fixed-bottom navbar navbar-dark bg-dark navbar-expand-
sm">
  <div class="container">
    <ul class="navbar-nav">
      <li class="nav-item"><a class="nav-link" href="#cards">Cards</a></li>
      <li class="nav-item"><a class="nav-link" href="#drop">Dropdown</a></li>
      <li class="nav-item"><a class="nav-link"
href="#NYCAccordion">Accordion</a></li>
      <li class="nav-item"><a class="nav-link" href="#slideshow">Slideshow</a></li>
    </div><!-- navbar-nav -->
  </ul><!-- container -->
</nav>
```