<!doctype html>

<html>

<head>

<link href="style.css" rel="stylesheet">

</head>

<body>

...

<script src="jquery.min.js"></script>

<script src="app.js"></script>

</body>

</html>

* To include CSS files in HTML we used the **link**element.
* Similarly, to include JavaScript files in HTML, we use the **script** element.

var main = function() {

$('.dropdown-toggle').click(function() {

$ ('.dropdown-menu').toggle();

});

};

$(document).ready(main);

////////////////////////////////////////////////////////////////////////////////////////////

We can write a function that specifies what to do when an event occurs. This function is called an **event handler**.

The **.click()** method attaches an event handler to an HTML element so that it can respond to a click event.

Another common user event is the **keypress event**. A keypress event occurs when a user types a key on the keyboard.

The **.keypress()** method attaches an event handler to an HTML element so that it can respond to a keypress event.

/////////////////////////////////////////////////////////////////////////////////////////////////////////

Triggering a keypress event handler when any key is typed isn't particularly useful. For more control, we can use the **event** object.

$(document).keypress(function(event) {

if(event.which === 109) {

$('.dropdown-menu').toggle();

}

});

The **event** object contains more information about an event, such as which mouse button was clicked or which key was pressed.

**event.which** contains which key was pressed. Keyboard keys are identified by [key codes](http://help.adobe.com/en_US/AS2LCR/Flash_10.0/help.html?content=00000520.html). The **m**key is identified by the key code 109.

///////////////////////////////////////////////////////////////////////////////////////////////////

var main = function() {

$('.article').click(function() {

$(this).children('.description').show();

});

};

$(document).ready(main);

/////////////////////////////////////////////////////////////////////////////////////////

The description of the current article shows and all other descriptions hide.

To make more clear which article is currently open, let's add the class 'current' to it. This is a CSS class in **style.css** that will shade it gray to indicate it is open.

var main = function() {

$('.article').click(function() {

$('.article').removeClass('current');

$('.description').hide();

$(this).addClass('current');

$(this).children('.description').show();

});

};

$(document).ready(main);

////////////////////////////////////////////////////////////////////////////

In jQuery, **$( )** has two main uses:

* To select existing HTML elements on the page.

**$('p')** selects all **p** elements on the page.

* To create new HTML elements to add to the page.

**$('<h1>')** creates a new **h1** element. The **< >**indicates that we want to create a new HTML element.

The **.text()** method adds text to an HTML element.

The **.appendTo()** method adds HTML elements to the end of the selected element, the **.prependTo()** method adds HTML elements to the beginning of the selected element.

* **.hide()** hides the selected HTML element
* **.show()** displays an element
* **.toggle()** alternates hiding and showing an element
* **.addClass()** adds a CSS class to an element
* **.removeClass()** removes a class from an element
* **.toggleClass()** alternates adding and removing a class from an element

The tree-like (forming a hierarchy) representation of HTML is called the **Document Object Model (DOM)**.

The **.next()** method gets the next sibling of the selected element.

The **.prev()** method gets the previous sibling of the selected element.

The **.children()** method gets the children of the selected element.

If provided a selector, the **.children()** method can get a specific child.

The **.slideDown()** method shows the selected HTML element by sliding it down; The **.slideUp()** method hides the selected HTML element by sliding it up.

The **.fadeIn()** method shows the selected HTML element by fading it in.

The **.fadeOut()** method hides the selected HTML element by fading it out.

The **.animate()** method lets you create your own custom animations. HTML elements can be animated by specifying the CSS properties to change to over a time duration:

**$('.menu')** selects the navigation menu element on the page.

**.animate()** takes two parameters:

1. A set of CSS properties,
2. A time duration over which to change them.

Here, we want to change the **width** of the **.menu**element from its current width to 193px.