[This is heading 1 2](#_Toc478306012)

[This is heading 2 2](#_Toc478306013)

[This is heading 3 2](#_Toc478306014)

[This is heading 4 2](#_Toc478306015)

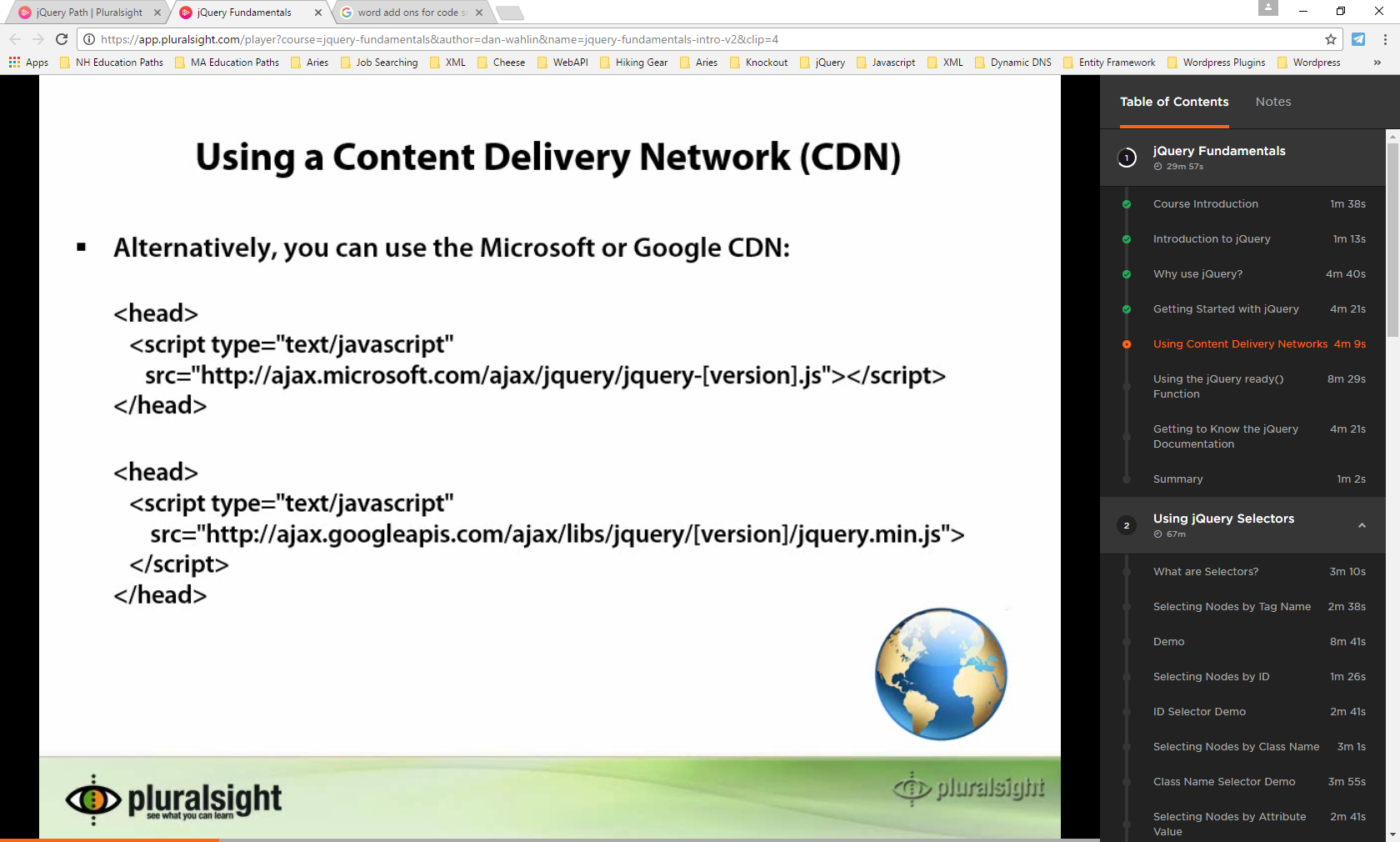
[This is heading 5 2](#_Toc478306016)

[This is heading 6 2](#_Toc478306017)

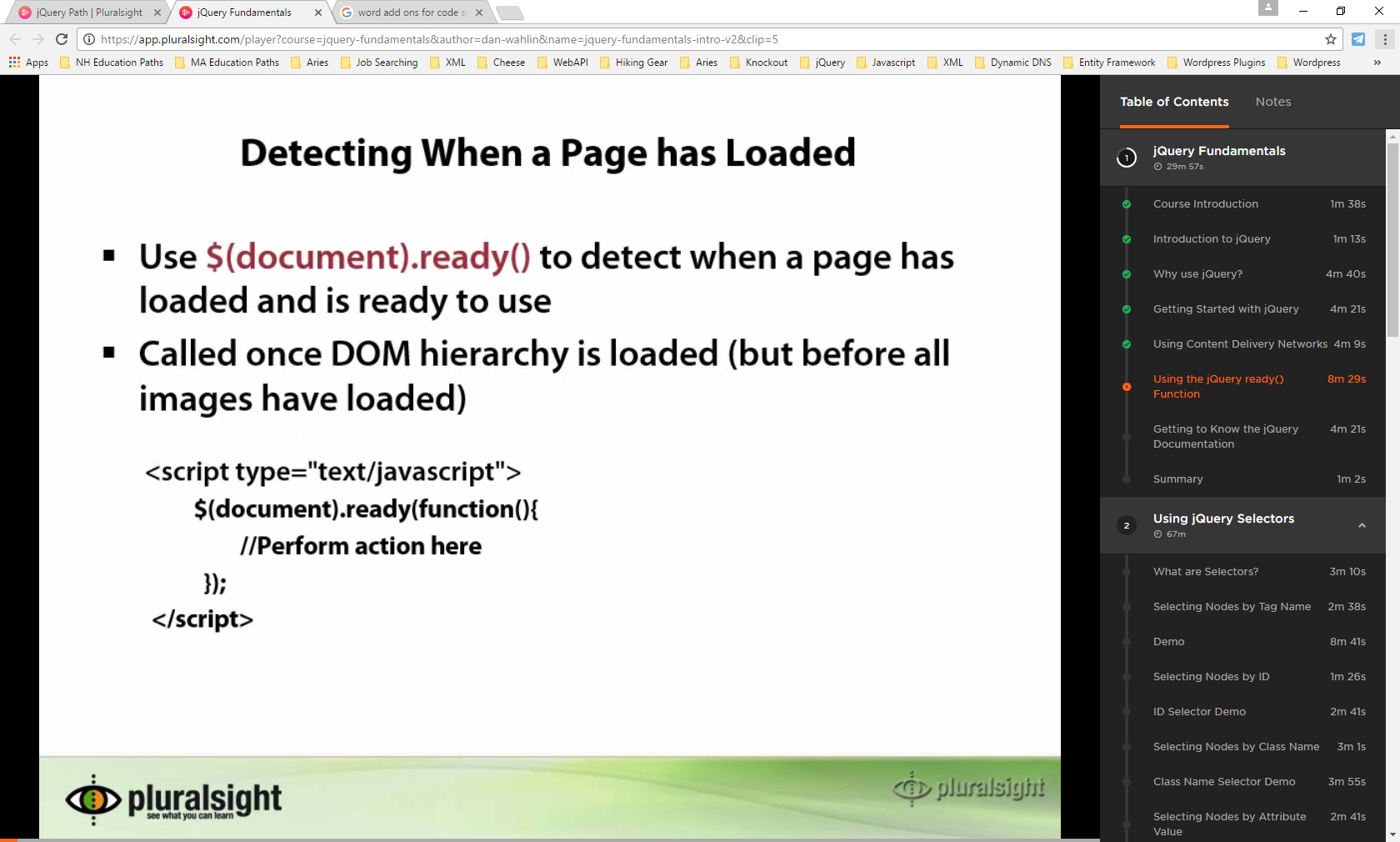
# Basics

## Referencing in web page

<script type = **"text/javascript"** src=**"jquery.js"**></script>



## Document Ready



<body>

<script type = **"text/javascript"**

$(document).ready(Startup)

</script>

<script>

***function*** Startup**(){**

**}**

</script>

</body>

# No Conflict Mode

<body>

<script>

// assign jQuery to another variable

// to avoid other libraries that use the $ sign as a

// namespace referece

***var*** $jQuery **=** jQuery.noConflict**();**

//or

***var*** $j **=** jQuery.noConflict**();**

</script>

</body>

# Selectors

## By tag name

<script>

$j**(**'tagmoniker'**)**

$j**(**'a'**)** // selects all hyperlinks

</script>

<script>

$j**(**'a, span, div'**)** // selects all hyperlinks, span elements and div

// elements

</script>

## Selecting Descendants

<script>

$j**(**'table tr'**)** // selects all row elements for all tables

</script>

## Selecting By Element ID (#)

<script>

$j**(**'#IDNAME'**)** // selects element with this id

</script>

## Select By Class Name (.)

<script>

$j**(**'.CLASSNAME'**)** // selects elements with this class name

</script>

## Selecting By Attribute ([x])

<script>

$j**(**'[ATTRIBUTENAME'**)** // selects elements with this attribute name

</script>

## Selecting By Attribute Value([x = “”])

<script>

$j**(**'[ATTRIBUTENAME ="Value of Attribute"'**)** // selects elements with this

// attribute name that have the given value of the attribute

</script>

## Simple Multiple Selection (,)

<script>

$j**(**'.class1, .class2'**)** // selects elements with these class names

</script>

## Mixing Selectors

<script>

$j**(**'a.CLASSNAME'**)** // selects all hyperlinks wih class name

</script>

## Selecting Input Elements (:input[type=””])

<script>

$j**(**':input[type="radio"]'**)** // selects radio button input elements

</script>

## Selecting Via Element Attribute

<script>

// will find all the input elements that 'begin' with xxx

$jVARIABLE **=** $**(**'[attributename]'**);**

</script>

<input type=**"button"** value=**"xxx - "**/>

<script>

// will find all the input elements that 'begin' with xxx

$jVARIABLE **=** $**(**'input[value^="xxx"]'**);**

</script>

<script>

// will find all the input elements that ‘end’ with xxx

$jVARIABLE **=** $**(**'input[value$="xxx"]'**);**

</script>

<script>

// will find all the input elements that ‘contain’ xxx

$jVARIABLE **=** $**(**'input[value\*="xxx"]'**);**

</script>

## Getting to value of input control (.val)

$j**(**':input[type="text"]'**)[0].val()**

// get the value of a text box

## Chaining Selectors

# Capturing elements into variables

<script>

$jVARIABLENAME**;**

$jVARIABLENAME **=** $j**(**':input[type="text"]'**)**

// VARIABLE now holds a collection of input elements of type text

</script>

<script>

$jVARIABLENAME**;**

$jVARIABLENAME **=** $j**(**':input[type="text"]'**)**

// can now loop through the variable which may contain several elements

foreach**(**element ***in*** $jVARIABLENAME**)**

**{**

element.val**();**

**}**

</script>

# Looping through arrays of jquery variables (.each)

<script>

$jVARIABLENAME**;**

$jVARIABLENAME **=** $j**(**':input[type="text"]'**)**

// can now loop through the variable which may contain several elements

$jVARIABLENAME.each**(*function*(**index**,** element**){**

// do something here with the element

alert**(**element.val**());**

**});**

// each presents an element by default because you are iterating a jquery

// variable which is by default an array of elements

// to access the element that is currently in the loop

// can use $(this)

$jVARIABLENAME.each**(*function*(){**

// do something here with the element

alert**(**$j**(*this*).**val**());**

**});**

</script>

# This vs $This

You can always get to the root dom object element by using *this* instead of using $this.

$jVARIABLENAME.each**(*function*(){**

// do something here with the element

alert**(**$j**(*this*).**val**());**

**});**

$jVARIABLENAME.each**(*function*(){**

// do something here with the element

alert**(*this*.**title**);**

**});**

# Adding/Removing Elements

.appendTo

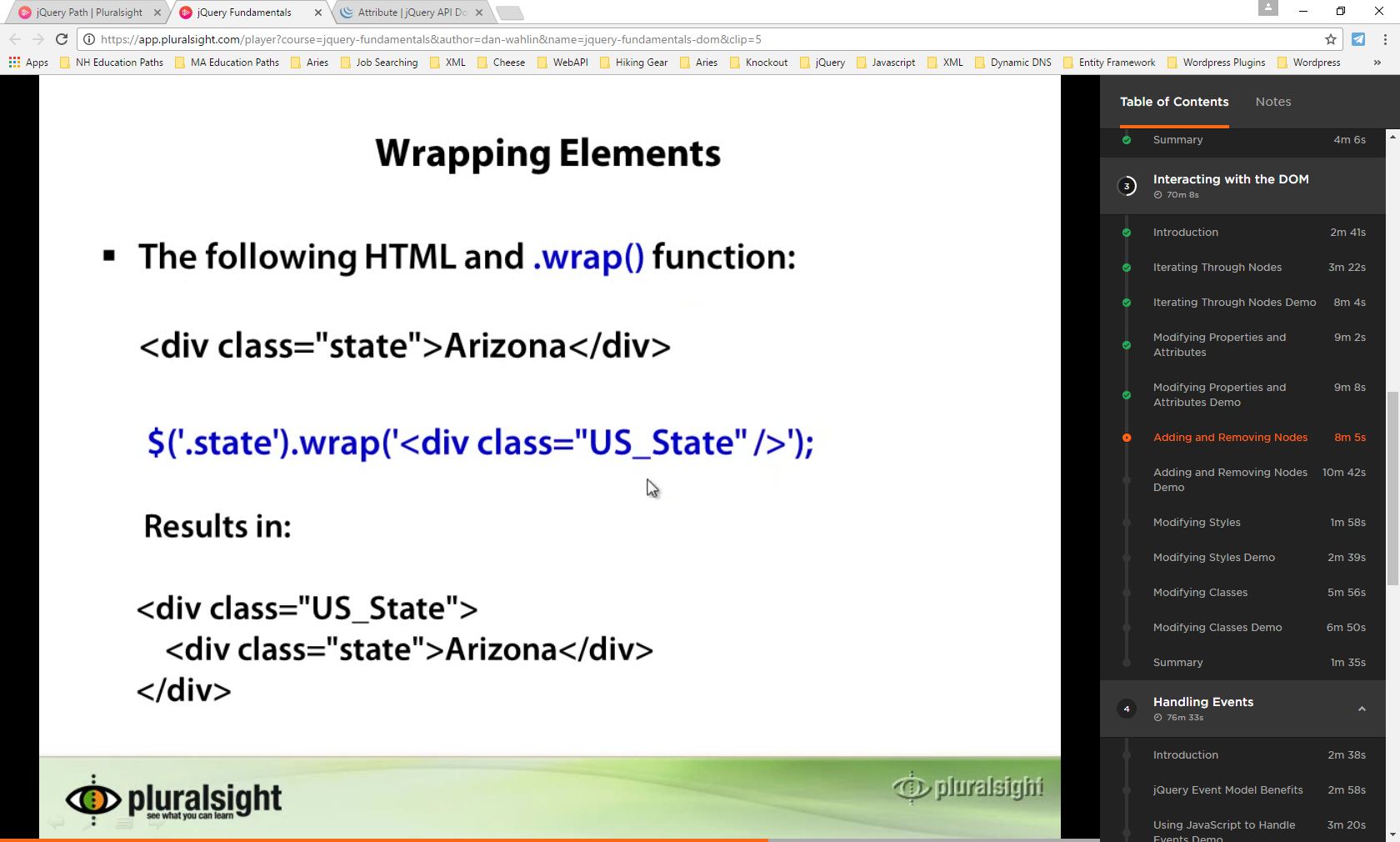
.append

.prependTo

.prepend

.wrap

.remove



# Working With Classes

.addClass

.hasClass

.removeClass

.toggleClass

# Functions don’t always need to be anonymous

Whenever an anonymous function is allowed in jQuery, you can actually point to the name of an actual javascript function instead.

<script>

$jVARIABLENAME**;**

$jVARIABLENAME **=** $j**(**':input[type="text"]'**)**

$jVARIABLENAME.each**(**DoTheAlert**);**

// now DoTheAlert will be called for each element

// in $jVARIABLENAME

//

***function*** DoTheAlert**(**element**){**

alert element.val**();**

**}**

</script>