jQuery Fundamentals

Eng. Niveen Nasr El-Den
SD & Gaming CoE
iTi

Day 1

jQuery

write less, do more.

jQuery

- jQuery is a fast and concise JavaScript Library created by John Resig in 2006 at Rochester Institute of Technology with a nice motto:

  "Write less, do more."

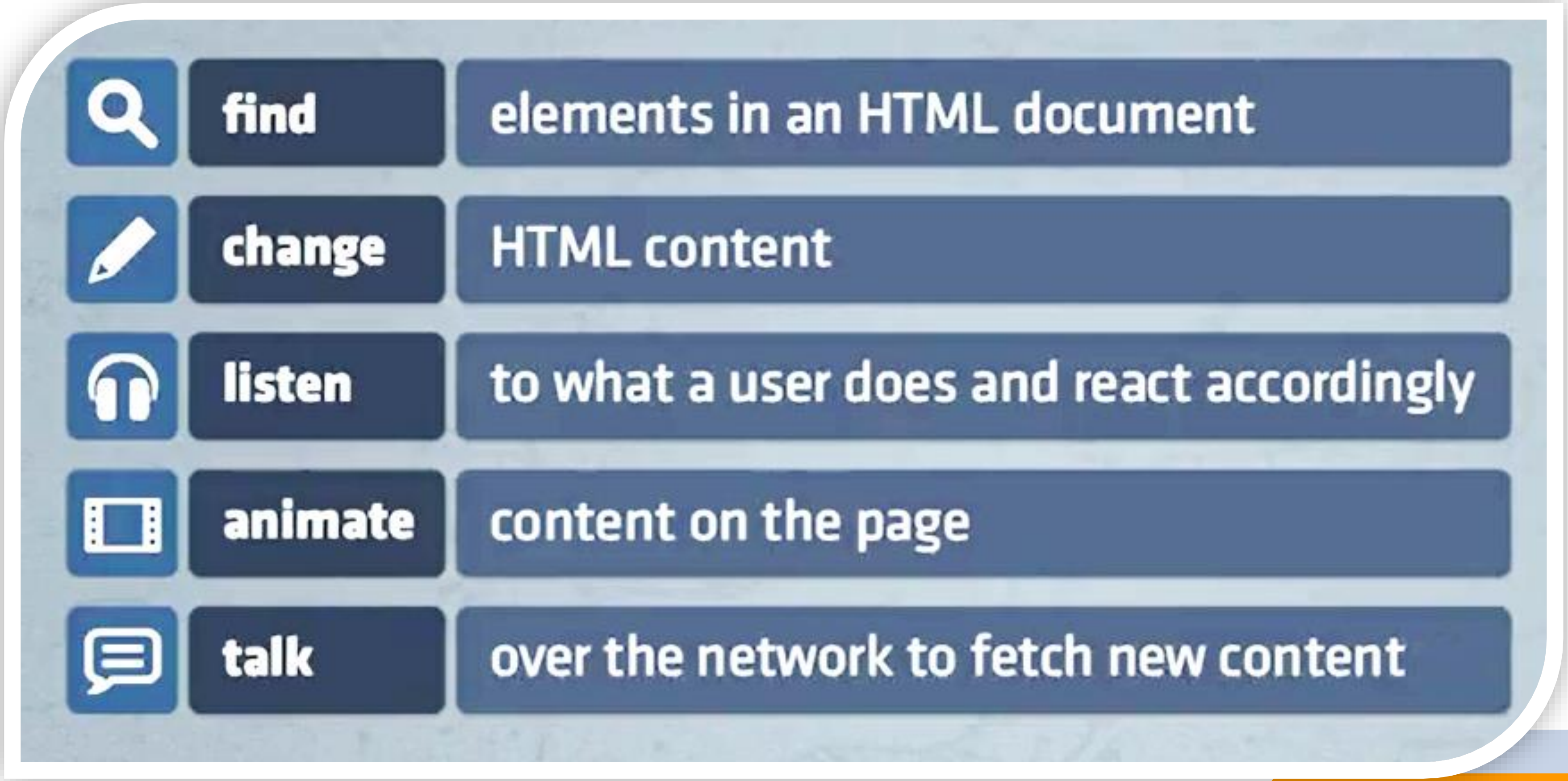
- It is one of many available libraries that
  - Provide functions for manipulating the web page with fairly good performance
  - Help to keep your JS code clean

- Current release 3.4.1

jQuery

- Powerful JavaScript library
  - Simplify common JavaScript tasks
  - Access parts of a page
  - Modify the appearance of a page
  - Alter the content of a page
  - Change the user’s interaction with a page
  - Add animation to a page
  - Provide AJAX support
  - Abstract away browser quirks

Simply jQuery Makes it easy to:



find	elements in an HTML document
change	HTML content
listen	to what a user does and react accordingly
animate	content on the page
talk	over the network to fetch new content

Why jQuery

- Rich Internet Applications (RIA)
- Dynamic HTML (DHTML)
- Friendly and Elegant API's
- DOM traversing is very easy especially for complex search criteria
- The core library size is very small
- jQuery UI Extension provides very rich controls
- Cross Browser Support

```
$ ('#something').hide().css('background', 'red').fadeIn();
```

Things jQuery Provides

- Select DOM elements on a page
- Set properties of DOM elements, in groups ("Find something, do something with it")
- Creates, deletes, shows, hides DOM elements
- Defines event behavior on a page (click, mouse movement, dynamic styles, animations, dynamic content)
- Reduces browser inconsistencies
- Plugins are available to cover all needs

Using jQuery

- Installation
  - just download the jquery-3.x.x.js file and put it in your website folder get it from “www.jquery.com”
  - Use development version in developing phase
    - Useful for debugging
  - Use production (minified) version for deploying
- Reference the JS file in your HTML
  - preferably insert it within the <head> tag.

```
<script type="text/javascript“ src="jquery-3.x.x.min.js">

</script>
```

Using jQuery via CDN

- Google

```
<script type="text/javascript“
src="http://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js">
</script>
```

- Microsoft

```
<script type="text/javascript“
src="http://ajax.microsoft.com/ajax/jquery/jquery-3.2.1.min.js">
</script>
```

Using jQuery via CDN

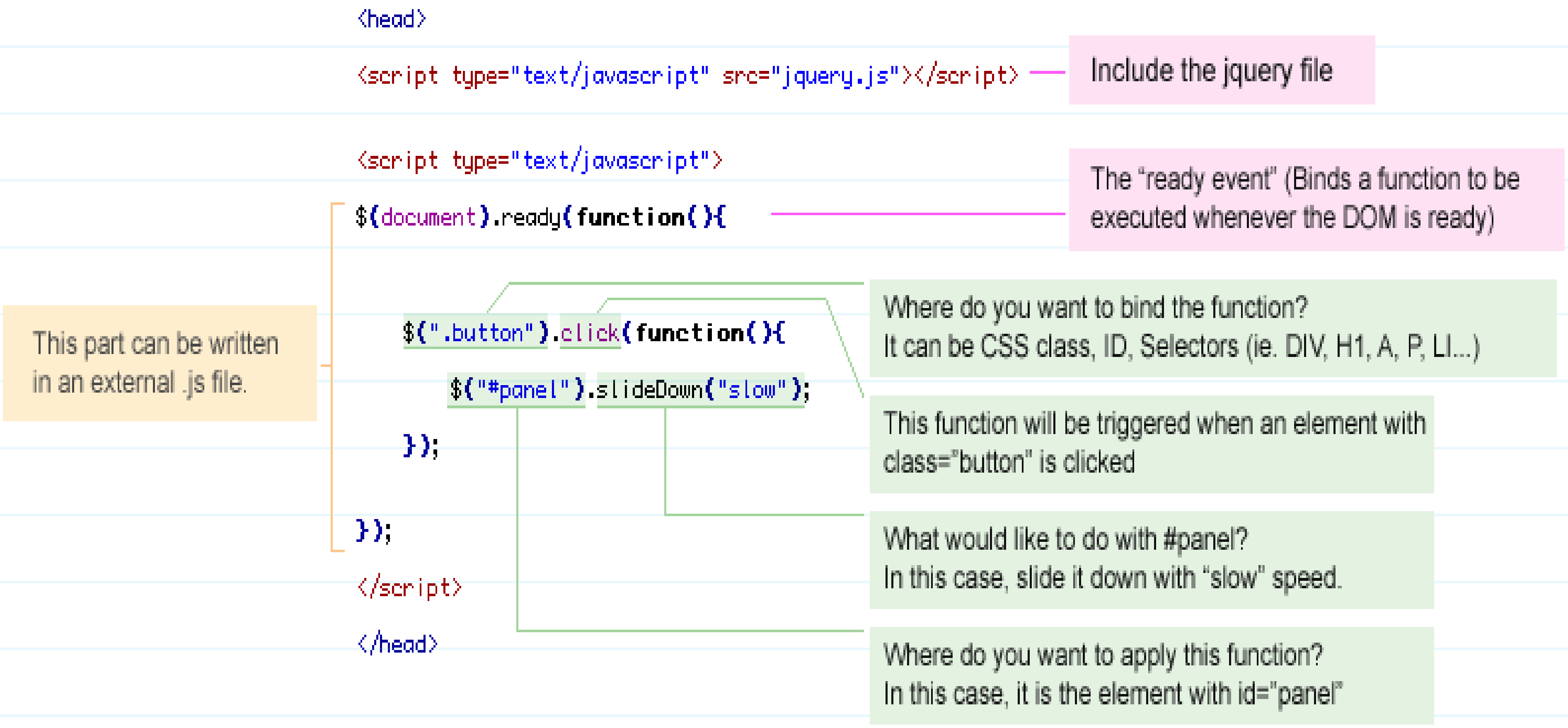
- jQuery

```
<script type="text/javascript“
      src=“http://code.jquery.com/jquery-3.2.1.min.js">
</script>
```

- Then, access the jQuery functions via the $ object

```
$(document).ready(
      function() {$("div.someClass").show();}
);
```

How jQuery works



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

<script type="text/javascript">
$(document).ready(function(){

    $(".button").click(function(){
        $("#panel").slideDown("slow");
    });

});
</script>
</head>
```

Include the jquery file

The "ready event" (Binds a function to be executed whenever the DOM is ready)

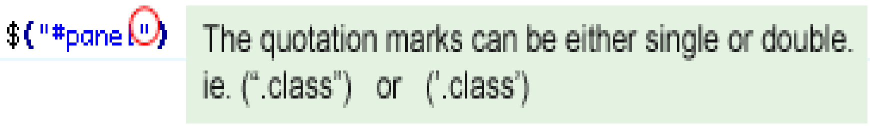
This part can be written in an external .js file.

Where do you want to bind the function?
It can be CSS class, ID, Selectors (ie. DIV, H1, A, P, LI...)

This function will be triggered when an element with class="button" is clicked

What would like to do with #panel?
In this case, slide it down with "slow" speed.

Where do you want to apply this function?
In this case, it is the element with id="panel"

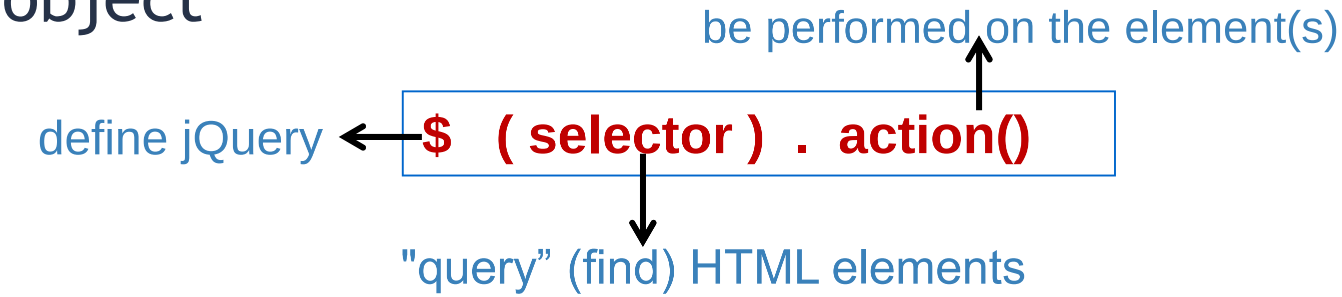


```
$("#panel")
```

The quotation marks can be either single or double.
ie. (".class") or ('.class')

How jQuery works

- Set up a basic HTML page and add jQuery
- Create a “ready” function
- Write functions to tell jQuery what to do via the $ object



be performed on the element(s)

define jQuery

$ (selector) . action()

"query” (find) HTML elements

- The jQuery syntax is tailor made for finding HTML elements and do some action on the element(s)

The Ready Function

- It checks the document and waits until document is ready to be manipulated
- Many ways to specify the ready function
  - $(document).ready(function(){…});
  - $(‘document’).ready(function(){…});
  - $().ready(function(){…}) → not recommended
  - $(function(){…}) → recommended starting from 3.x version
  - etc.
- Generally

  selector.*ready* (func)

  func → function to be executed whenever the DOM is ready to be traversed and manipulated.

jQuery

- jQuery Library supports us with a variety of utilities and features

1. Selectors
2. Attributes
3. DOM Manipulation
4. Traversing
5. CSS
6. Events
7. UI
8. Ajax
9. Animation
10. Plug-ins

jQuery Selectors & Filters

jQuery Selector

- jQuery can help you find elements based on their ID, classes, types, attributes, values of attributes and much, much more

- *A jQuery Selector* is a function which makes use of expressions to find out matching elements from a DOM based on the given criteria.

- The jQuery library harnesses the power of *Cascading Style Sheets (CSS) selectors* to let us quickly and easily access elements or groups of elements in the Document Object Model (DOM).

- All type of selectors available in jQuery, always start with the dollar sign and parentheses: *$().*

jQuery Selector

- The factory function **$(,)** makes use of following building blocks while selecting elements in a given document
  - The #id selector
  - The .class selector
  - The element selector
  - etc..

- The jQuery selector takes two parameters
  - selector,
  - context

jQuery Selector

jQuery	Description
Name	Selects all elements which match with the given element Name.
#ID	Selects a single element which matches with the given ID
.Class	Selects all elements which match with the given Class.
Universal (*)	Selects all elements available in a DOM.
Multiple Elements E, F, G	Selects the combined results of all the specified selectors E, F or G.

jQuery Selector

Selector	Result
$('*')	all elements in the document.
$("p > *")	all elements that are children of a paragraph element.
$("#specialID")	the element with id="specialID".
$(".specialClass")	all the elements that have the class of *specialClass*
$("p:empty")	all elements matched by <p> that have no children.
$("p a.specialClass")	matches links with a class of *specialClass* declared within <p> elements.
$("input[name=myname]")	all elements matched by <input> that have a name value exactly equal to *myname*.
$(":radio")	all radio buttons in the form.
$("li:even“)	all elements matched by <li> that have an even index value.

Context parameter

- The context is where the jQuery selector runs

```
$ (selector , context)  =  $  (context) . find (selector)

Similar to ➔ $ (context   selector)
```

- Example:
  $(“a”,”div”);   Similarly as $(“div a”)
  It means select all anchors (a) but only those in div

- Context parameter is confusing and not worth using

Selecting Elements by Index Order

- Depending on what you want to do, you have the following filters.
- These may look like CSS pseudo-classes, but in jQuery they're called filters

filter	Result
:first	Matches the first selected element
:last	Matches the last selected element
:even	Matches even elements (zero-indexed)
:odd	Matches odd elements (zero-indexed)
:eq(n)	Matches a single element by its index (n)
:lt(n)	all elements matched with an index below n
:gt(n)	all elements matched with an index above n

Attribute Selectors

- Results of a selector statement can be filtered based on attribute contents

$('[attr]')	element with attr
$('[attr="val"]')	attr equals val
$('[attr!="val"]')	attr does not equal val
$('[attr~="val"]')	attr has val with space-sep.
$('[attr^="val"]')	attr begins with val
$('[attr$="val"]')	attr ends with val
$('[attr*="val"]')	attr contains val
$('[attr1][attr2]')	element with attr1 and attr2

Form Selectors

Selector	Description
:input	Finds all form elements (input, select, textarea & buttons)
:text	Returns all text elements
:password	Returns all password elements
:radio	Returns all radio button elements
:checkbox	Returns all checkbox elements
:submit	Returns all submit elements
:reset	Returns all reset elements
:image	Returns all image elements
:button	Returns all button elements
:enabled	Returns all enabled form elements
:disabled	Returns all disabled form elements
:checked	Returns all checked form elements
:selected	Returns all selected form elements

CSS Form Selectors

$(':text')	<input type="text">
$(':password')	<input type=" password ">
$(':radio')	<input type="radio">
$(':button')	<input type="button">
$(':selected')	<option selected="selected">
etc.	

:contains(txt) Selector

- Used to select an element based on string it contains
- it’s case sensitive.

- Example:
  `$('span:contains("A")');`

:not() Selectors

- jQuery gives us the :not filter, which you can use in the following way:
  $('div:not(#content)');
  → Select all DIV elements except #content

- The selector can be as complex as you like:
  $('a:not(div.important a, a.nav)');
  →Selects anchors that do not reside within 'div.important' or have the class 'nav'

Custom Misc. Selectors

Selector	Description
:contains(str)	Used to select an element based on what it contains. Case sensitive.
:animated	Selects currently animated elements
:parent	Matches all elements that are parents, they should contain at least one other element including text Finds all elements with children, including text.
:has(sel)	Selects elements that contain at least one matching the passing sel parameter
:visible	Filters the returning selection results with only visible elements
:hidden	Filters the returning selection results with only hidden elements
:header	Selects all header elements (h1→h6)

Other Selectors

- :nth-child(index)
- :first-child
- :last-child
- :only-child
- etc…

filter() Method

- The benefit of using this function is that you can make a combination of more than one filter
- It is used Instead of writing the filter expression within selector methods $()
- Example:
  - $(‘ul li:eq(1)’)
    →is same as ('ul li').filter(':eq(1)');
  - $('ul li').filter(':eq(0),:eq(2)');
    → selecting 1st and 3rd items in an unordered list

filter() Method

- Used to select an element varying characteristics such as programmatic states not expressible as selector expressions.
  - Pass function to filter() method as a parameter instead of a text.

- Example:

```
$('div').filter(function(){
    var width = jQuery(this).width();
    return width > 100 && width < 200;
});
```

→Select all DIV elements with a width between 100px and 200px.

jQuery Attributes Methods

Attributes & Properties

- The basic components we can manipulate when it comes to DOM elements are the *properties* and *attributes* assigned to those elements.

- Most of these attributes are available through JavaScript as DOM node properties.

- Some of the more common properties are:
  - className
  - tagName
  - id
  - href
  - title
  - src

Attributes & Properties

- Getting/setting & removing attributes
  - .attr(attr_nm [,val])
  - .removeattr(attr_nm)

- Getting/setting elements content & values
  - .text([“str”])
  - .val([“val”])
  - .html([“str”])

jQuery attr() Method

- Used to either *get* or *set* the value of an attribute from the first element in the matched set or from all matched elements.

```
<img id="myImage" src="image.gif“  title="This is an image"/>
```

- Get an Attribute value → *attr*(“attName”)

  var imgTitle = $(“img").*attr*(“title");

- Set an Attribute value: → *attr*(name, value)

  $("#myimg").*attr*("src", "/images/jquery.jpg");

jQuery attr() Method

Attribute Methods that handles *attribute values*

Method	Description
attr(props)	Set a key/value object as properties to a matched elements. e.g. $(‘img’).attr({src:””, title:””…});
attr(key,fn)	Set a single property to a computed value, on all matched elements. **key:** The name of the property to set. **func:** A function returning the value to set
removeAttr(nm)	Remove an attribute from each of the matched elements e.g. $(‘img’).removeAttr(title);

jQuery element content Methods

jQuery Methods that handles *html and text content*

Method	Description
html()	Get the html contents (innerHTML) of the first matched element.
html(val)	Set the html contents of every matched element.
text()	Get the combined contents of all matched elements.
text(val)	set the text contents of all matched elements

jQuery element content Methods

jQuery Methods that handles *form elements values*

Method	Description
val()	Get the input value of the first matched element
val(val)	Set the value attribute of every matched element if it is called on <input> but if it is called on <select> with the passed <option> value then; passed option would be selected, if it is called on check box or radio box then all the matching check box and adiobox would be checked.

css() Method

Method	Description
css(name)	Get the value of the css named property for the 1st matched element
css(prop, val)	Set the css prop with the given val to all matched elements. e.g. $("#cssSpan").css(“background-color”, “Yellow”);
css(props)	Set the css properties for all matched elements. e.g. $("#cssSpan").css({'background-color':'Yellow', 'color':'black’});

jQuery Animations & Effects

Show/Hide Animation

[selector].*show* (speed , [callback]);

❑***speed*** → "slow“ , "normal“, or "fast“
200ms 400ms 600ms

❑***callback*** → a function to be executed whenever the animation completed

[selector].*hide* (speed, [callback]);

Fading Animation

fadeIn(speed, [callback])	Fade in all matched elements by adjusting their opacity and firing an optional callback after completion.
fadeOut(speed, [callback])	Fade out all matched elements by adjusting their opacity to 0, then setting display to "none" and firing an optional callback after completion.
fadeTo(speed, opacity, callback)	Fade the opacity of all matched elements to a specified opacity and firing an optional callback after completion.
fadeToggle(speed, [callback])	Toggle Fading in and out

Sliding Animation

slideDown(speed, [callback])	Reveal all matched elements by adjusting their height and firing an optional callback after completion.
slideUp(speed, [callback])	Hide all matched elements by adjusting their height and firing an optional callback after completion.
slideToggle(speed, [callback])	Toggle the visibility of all matched elements by adjusting their height and firing an optional callback after completion.

Assignment