Unit 6.2 Highlights

(Source: Wikipedia)
API Key - Code passed in by computer programs calling an API to identify the calling program, its developer, or its user. API keys are used to track and control how the API is being used to prevent malicious use, abuse of the API, etc.

HTTP Methods (Source: W3Schools)

GET - used to request data from a specified resource
POST - used to send data to a server to create/update a resource
PUT - used to send data to a server to create/update a resource
HEAD - almost identical to GET, but without the response body
DELETE - deletes the specified resource
PATCH - used to make partial changes to an existing resource
OPTIONS - describes the communication options for the target resource

---

(Source: W3Schools)
HTML data-* Attributes - used to store custom data private to the page or application and gives the ability to embed custom data attributes on any HTML element Two rules:

- The attribute name should not contain any uppercase letters, and must be at least one character long after the prefix "data-"
- The attribute value can be any string

<u>Ex. 1 (Pattern)</u>

```
<element data-*="somevalue">
```

<u>Ex. 2</u>

```
<h1 data-service-value="power systems">Power Systems</h1>

console.log($("h1").attr("data-service-value")); //power systems
```

---

event.preventDefault() - stops the default action of an element from happening (i.e. Prevent a submit button from submitting a form or a link from following the URL)

<u>Ex. 3</u>

```
$("a").on("click", function(event) {
        event.preventDefault(); //the hyperlink is not followed
});
```

---

jQuery Get/Set Input Field Values (Source: W3Schools)

<u>Ex. 1</u>

```
// Get the value from a dropdown select directly
$( "select" ).val();

// Set the value from an input field
$( "input" ).val("setting a default value for input field”);
```

JavaScript Text Whitespace (Source: W3Schools)

.trim() - Remove whitespace from both sides of a string:

<u>Ex. 2</u>

```
var str = "       Some kind of text string       ";
console.log(str.trim()); // "Some kind of text string”
```

(Source: jQuery API Docs)

$("someselector").attr() - Get the value of an attribute for the first element in the set of matched elements or set one or more attributes for every matched element.

Ex. 4 (Pattern)

```
$(selector).attr(attribute)
$(selector).attr(attribute,value) //set single attribute values
$(selector).attr({attribute:value, attribute:value,…})  //set multiple attribute values
```

Ex. 5

```
$("body").attr("width");
$("#result").attr("id","newID") //set single attribute values
$("#main").attr({name: "cleverDiv", alt:"alternative text"}) //set multiple attribute values
```

$("someselector").css() - Get the value of a computed style property for the first element in the set of matched elements or set one or more CSS properties for every matched element.

Ex. 6 (Pattern)

```
$(selector).css(property); //get single property value
$(selector).css(property,value) //set single property values
$(selector).css({property:value, property:value,…})  //set multiple property values
```

Ex. 7

```
$("body").css("width"); //get single property value
$("#result").css("backgroundColor","PALEGOLDENROD") //set single property values
$(“div").css({backgroundColor: "SIENNA", width:"500px", height:"200px"}) //set property
attribute values
```

$("someselector").text() - Get the combined text contents of each element in the set of matched elements, including their descendants, or set the text contents of the matched elements

Ex. 8

```
$("#content-div").text("This is some sample text");
```

$("someselector").html() - Get the HTML contents of the first element in the set of matched elements or set the HTML contents of every matched element

Ex. 9

```
$("#content-div").html("<p>This is some sample text</p> and an image <img src='https://placehold.it/
100x100/'>");
```

———————————————————————

someArrayVariable.push() - Append one or more elements to the end of an array

Ex. 10

```
var villains = var villains = ["Galactus","Juggernaut","Loki","Thanos","Dr. Doom"];
villains.push("Omega Red");
console.log(villains); //[ 'Galactus','Juggernaut','Loki','Thanos','Dr. Doom','Omega Red' ]
```

———————————————————————

someArrayVariable.unshift() - Prepends one or more elements to the beginning of an array

Ex. 11

```
var creatures = ["Gators", "Jackets", "Cyclones", "Panthers"];
creatures.unshift("Huskies");
console.log(creatures); //[ 'Huskies', 'Gators', 'Jackets', 'Cyclones', 'Panthers' ]
```

---

someArrayVariable.shift() - Pulls the first element off of the given array and returns it

Ex. 12

```
var colors = ["Yellow", "Orange", "Green", "Red"];
var result = colors.shift();
console.log(result); //Yellow
console.log(colors); //[ 'Orange', 'Green', 'Red' ]
```

---

someArrayVariable.pop() - Pulls the last element off of the given array and returns it

Ex. 13

```
var annoyingThings =["mosquitos","cold soup","uncapped toothpaste","bullet ants"];
var result = annoyingThings.pop();
console.log(result); //bullet ants
console.log(annoyingThings); //[ 'mosquitos', 'cold soup', 'uncapped toothpaste' ]
```

(Source: jQuery API Docs)

Event delegation - Allows the attachment of a single event listener, to a parent element, that will fire for all descendants matching a selector, whether those descendants exist now or are added in the future.

Events propagation occurs any time one of our selected tags is clicked. A clickevent is fired for that tag, and then bubbles up the DOM tree, triggering each of its parent click event handlers.

Ex. 14

```
// Add a new element on to our existing list
$( "#list" ).append( "<li><a href='http://newdomain.com'>Item #5</a></li>" );

// Attach a delegated event handler
$( "#list" ).on( "click", "a", function( event ) {
    event.preventDefault();
    console.log( $( this ).text() );
});
```