



# MODULE 4

# QUIZZES & EXPERIMENTS

---

*Web Front End Certification Course - SkewCode*



# Quiz 1

## *call() and apply()*

# CALL( ) AND APPLY( )

---

- Math.min( ) is a method that returns the minimum of the arguments passed to it.

```
Math.min( 3, 8, 4, -2, 9, -10, 12 ); // return -10
```

How can you use Math.min to find the minimum among the items of an array?

- You would like to sort the arguments passed to a function you plan to write. How can you use Array's sort method to do this?

```
var nums = [1, 5, 3, 8, 2];
```

```
nums.sort( ); // nums is now [1, 2, 3, 5, 8]
```

Note: Array's methods are 'duck-typed'. If an object's properties are numbered 0, 1, ...etc. & they have a length property, the array methods can be 'called' in the context of the object.

# *Experiment - 1*

*call()* and  
*apply()*



EXPERIMENT

# CALL() AND APPLY()

---

```
► function Person( name, age ) {  
    this.name = name;  
    this.age = age;  
}  
Person.prototype.getAge = function() { return this.age; }  
Person.prototype.increaseAge = function() { this.age++; }
```

Define an Employee class that additionally has the property department and role. When creating Employee class, reuse the person class constructor to setup name and age, and set the Employee.prototype to inherit from Person.prototype. This way Employee objects can reuse Person object methods. Also add a method promote( newRole ) that sets a new role for the employee object.

► This is the way inheritance is established (usually) in JS



# Quiz 2

## *Array methods*

# ARRAY METHODS

---

- Explain how the initial value of accumulator is set in Array's reduce method. How does the value change in each iteration?
- How will you sort an array of objects (say persons), based on their email ids (say alphabetical).
- Which array iterator will you use to get a new array consisting of squares of items of a given array?

# *Experiment - 2*

## *Array Methods*



# ARRAY METHODS

---

- var shareHolders = [  
    { name: 'Ajay Singh', numStocks: 1000000 },  
    { name: 'Priya Singh', numStocks: 500000 },  
    { name: 'K. Morawn', numStocks: 2000000 }  
];
- Sort the array items in increasing order of numStocks
- Filter those that have numStocks  $\geq$  750000
- Find out if every share holder has numStocks  $\geq$  500000
- Find the total number of stocks
- Get an array of share holder names
- For all of the above (except sorting) use iterator methods



# *Quiz 3*

# *Date Methods*

# DATE METHODS

---

- What is Unix epoch time? What does `getTime()` return?
- Which of these will get the year 2017 when run now?
  - `getYear()`
  - `getFullYear()`
- A date includes which of these?
  - The date
  - The time
- How would the following date show up when the following are called?  
date is Sun Jul 09 2017 04:42:39 GMT+0530 (IST)
  - `date.toUTCString()`
  - `date.toLocaleString()`

# *Experiment - 3*

## *Date Methods*



DATE  
METHODS

# DATE METHODS

---

- Create the following date objects
  - Date representing today
  - Date representing tomorrow (With same time as now)
  - Date representing your birthday (midnight time)



# Quiz 4

## JSON

# JSON

---

- Can you convert arrays into a string using `JSON.stringify`?
- Explain the typical way JSON methods are used when working with an application's backend.
- How will you store and work with JSON when using local storage or session storage?

# *Experiment - 4*

*JSON*



# JSON

---

- Store the address object created in the previous module in local storage.
- Retrieve it and make changes to any contact info (email/ phone) and update local storage with that information.



# Quiz 5

## *Strict Mode Execution*

# STRICT MODE EXECUTION

---

- How can you enable strict mode execution of scripts?
- What happens when you enable strict mode execution at global and function level?
- What is the output of this?

```
'use strict';
function foo() {
    console.log( this );
}
foo();
```

What would your answer be if we removed the ‘use strict’ directive?

*Experiment - 5*

*Strict Mode*

*Execution*



# STRICT MODE EXECUTION

---

- Enable strict mode. Try assigning a value to an undeclared variable. What do you see? Remove the directive and run again. What do you see?
- Do the following sequentially.
  - Don't enable strict mode execution and create a MovieCharacter object from last module, without using the new operator. Where do the data members go? Is a new object really created?
  - Now enable strict mode execution within MovieCharacter constructor function and repeat. What happens?
  - Now, additionally, use the new keyword to create the object.



# Quiz 6

## *Exception Handling*

# EXCEPTION HANDLING

---

- Explain how control flows when an exception is raised within `baz()`.

```
function baz() {  
    ...  
    // some exception occurs here (not within try..catch block)  
    ... // lines following  
}  
  
function bar() {  
    ...  
    baz();  
    ... // lines following call to baz()  
}  
  
function foo() {  
    try {  
        bar();  
    } catch( e ) { console.log( e ); }  
}
```

What would happen if the `try..catch` did not exist? What if the call to `baz()` were enclosed within a `try...catch`?

*Experiment - 6*

*Exception*

*Handling*



# EXCEPTION HANDLING

---

- Change MovieCharacter constructor so that you throw an error object with appropriate message when you get arguments that are of unexpected data type.
- Enclose the object creation within a try...catch block and log an appropriate message in console when error occurs.



# *Quiz 7*

## *Nodes & The DOM Tree*

# NODES & THE DOM TREE

---

- What is the type of each element in the HTML page? Do they share a particular base type? What is it?
- What does the following log/fill in the empty space.
  - `console.log( document );`
  - `console.log( document instance Document );`
  - `console.log( document instance Element );`
  - `console.log( document instance Node );`
  - `console.log( document.body instance Element );`
  - `console.log( _____ . _____ ); // html element`
  - `console.log( _____ . _____ ); // head element`
  - `console.log( _____ . _____ ); // body element`

# *Experiment - 7*

## *Nodes & The*

### *DOM Tree*



# NODES & THE DOM TREE

---

- Open any web page. Check out the Elements tab in the dev tools and select an element. Check the properties tab on the right. What do you see?
- Select a comment. Is it of type Element? Is it of type Node?
- Select a text (plain text node and not an HTML tag with text). Is it of type Element? Is it of type Node?
- Comments and text nodes have a `nodeValue` property. Check it out.
- Only text and HTML tags are of type Node.
- Find what the following evaluates to
  - `document.nodeType`, `document.nodeName`
  - `document.body.nodeType`, `document.body.nodeName`
  - Select some element within the page using `document.getElementById()`. Find its `nodeType` and `nodeName`



Quiz 8

*Node Relationships and DOM  
Tree Traversal*

# NODE RELATIONSHIPS AND DOM TREE TRAVERSAL

---

- What does the following evaluate to?
  - `document.firstChild`
  - `document.firstElementChild`
  - `document.body.children`, `document.body.childNodes`
  - `document.body.children[n].previousSibling`,  
`document.body.previousElementSibling` (n is some number)
  - `document.body.childNodes` instanceof `NodeList`
  - `document.body.childNodes` instanceof `Array`
  - `document.body.parentNode`
  - `document.body.children[10].childNodes[2]`

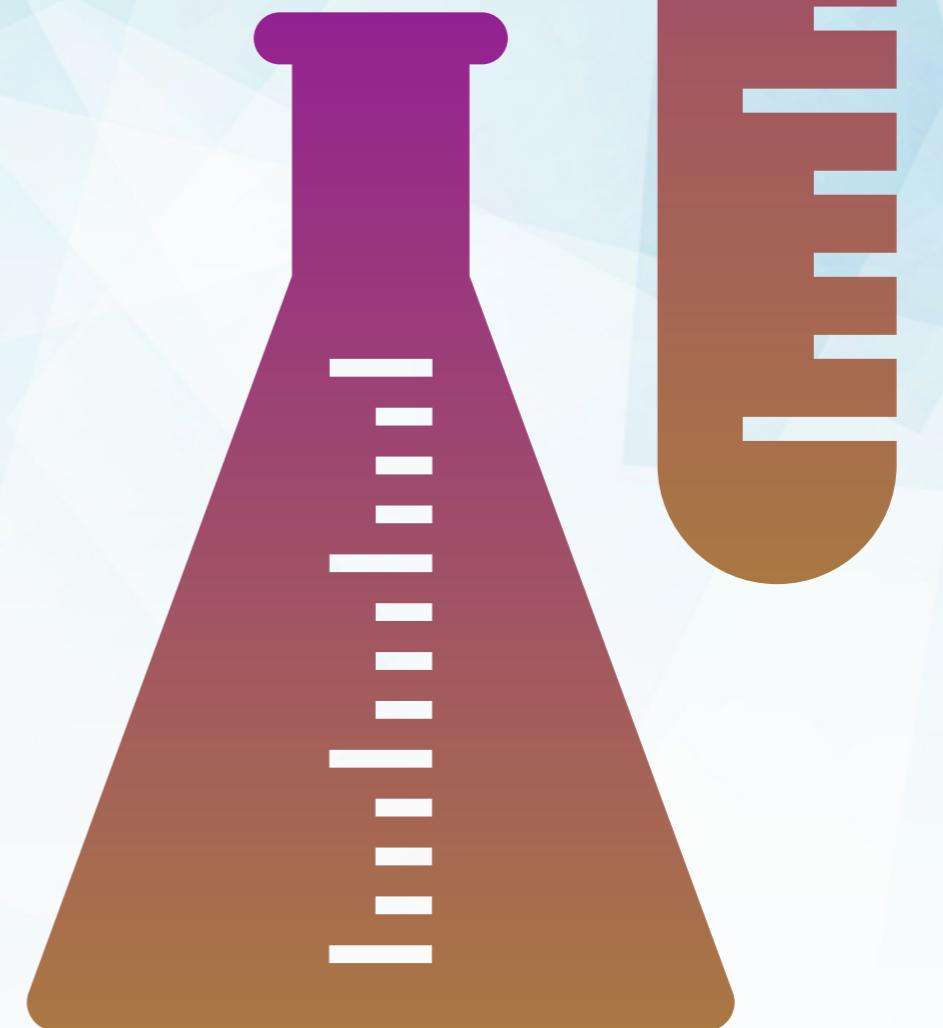
# *Experiment - 8*

## *Node*

## *Relationships and*

## *DOM Tree*

## *Traversal*



# NODE RELATIONSHIPS AND DOM TREE TRAVERSAL

---

- Open a page and do the following. For each step, log the selected node's nodeType, nodeName and nodeValue.
- Select the 2nd element child of body
- Now select the 3rd childNode of the selected element
- Move to its next sibling
- Move to the sibling's parent node



## Quiz 9

*Methods for DOM Manipulation*

# METHODS FOR DOM MANIPULATION

---

- How will you get an element by its id? Does this return a Node or a NodeList?
- How will you get a set of elements using a CSS selector? Does this return a Node or a NodeList?
- How will you create a paragraph node? How will you add text to it?
- How will you add this paragraph to the end of the body?
- How will you add it as the first child of the body?
- How are standard HTML element attributes accessed using their corresponding element nodes?
- How are inline styles of a node accessed and modified?

*Experiment - 9*

*Methods for DOM*

*Manipulation*



# METHODS FOR DOM MANIPULATION

---

- Create an empty HTML page (empty body).
- Using JS, create and add a paragraph to it. Set the id of the paragraph.
- Create and add a link to it. Set the href to <https://www.google.com>.
- Create a text node and set the text ‘Google Search’ and add it as a child of the link.
- Now modify the link styles to give a custom foreground and background color.
- Use setAttribute to change the href to <https://mail.google.com>. Change the text within to ‘Gmail’ using innerHTML property of the link.



# Quiz 10

## *Various Browser Events*

# VARIOUS BROWSER EVENTS

---

- Name some browser events.
- What event is fired and on which elements?
  - A button is clicked
  - The page loads
  - The page unloads
  - An image loads
  - User types within a text box
  - User clicks on a text box and the cursor shows up within it
  - User moves away from the text box and the cursor no more blinks inside it.
  - The HTML page is fetched and DOM nodes are created for all elements on the page.
  - User moves the mouse pointer inside an element.



# Quiz 11

## *Event Handling*

# EVENT HANDLING

---

- What are the phases of event propagation? Which order are the event handlers executed in?
- What are the the 3 ways of adding event handlers? What are the pros and cons? Which is the best way to handle events?

*Experiment - 11,*

*12*

*Various Browser*

*Events, Event*

*Handling*



# EVENT HANDLING

---

- Declare an input text box. Add a span below it. When the user types anything within the text box, the value should also get reflected in the text box below it.
- When the user tries to close the browser tab, display a confirmation dialog asking if the user really wants to leave the web site.



# Quiz 12

## *Event Object*

# EVENT OBJECT

---

- What is the event object? What does it contain. How is it obtained inside the event handler?
- List some special information available in the click event handler?
- You want to display a confirm dialog when user clicks on a link and navigate only if OK button is pressed. How will you do it?
- You want to cancel all event handlers lined up to run after the current one. How can you do it?
- You want to cancel all event handlers lined up to run after the current one. However any other event handlers setup on the current event target need to run. How can you do it?

*Experiment - 12*

*Event Object*



# EVENT OBJECT

---

- Write a script that intercepts all links with an href value set to an external url (internal links need to be ignored) and sets up code so that a confirm dialog is displayed when user clicks on the link. If OK is pressed proceed with navigation. Else cancel navigation to the linked page.
- Write a script that display screen scroll coordinates as user scrolls. When the end of the document is reached, an alert is displayed.
- Note that the scroll event is handled to implement a feature like infinite scroll (e.g. new products/posts load when you reach the bottom of the page).



# Quiz 13

## *jQuery Basics*

# JQUERY BASICS

---

- Which of the following are advantages of using jQuery
  - Cross-browser compatibility of code
  - Easier to code and quickens development
  - Large number of useful methods
  - Extensible via large set of plugins
- Why is jQuery code encapsulated within a \$(document).ready(function() {})? Is this strictly required?
- What is returned by a \$() call? What does it contain?
- Name some jQuery collection methods. Can these methods be applied to DOM nodes?

*Experiment - 13*

*jQuery Basics*



# JQUERY BASICS

---

- Create an HTML document with 4 divs. Add class “message” to the 2nd and 4th divs. Also add unique ids.
- Add a span child for the 2nd and 4th divs. Clicking the span should alert the id of the corresponding div containing them.
- Once page loads, select the divs with class message using jQuery code and add the text “Test message” within them.
- Add a text box and a button. When the user types a named CSS color (red, blue etc.) within the text box and clicks the buttons, the message divs should turn that color.
- Add a data-index property to each div and set it to the serial number of the div (1, 2, etc.). Update code so that the text within has the number of the div (use jQuery each() to do it).



# Quiz 14

## *DOM Manipulation in jQuery*

# DOM MANIPULATION IN JQUERY

---

- What is the difference between append( ), appendTo( ), after( ) and insertAfter( )?
- What do the following return for the input element when the page loads? What will they return after the checkbox has been unchecked?

```
<input type="checkbox" checked="checked" id="choice" />
```

- \$('#choice').prop('checked');
- \$('#choice').attr('checked');

# DOM MANIPULATION IN JQUERY

---

- How will you do the following on elements with class message?
  - Remove class message-default (if exists) and add another class called message-success
  - Insert a horizontal line after each message
  - Create a copy of all messages on the page and append the copies to the body

*Experiment - 14*

*DOM*

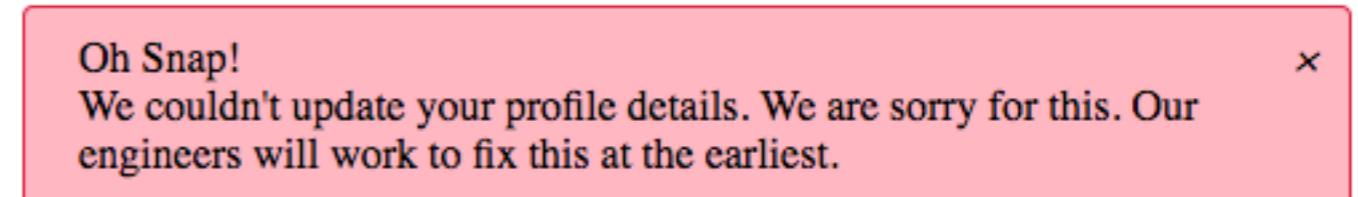
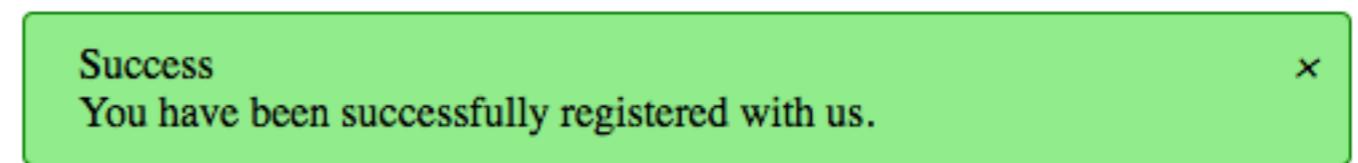
*Manipulation in  
jQuery*



# DOM MANIPULATION IN JQUERY

---

- Create a CSS class that defines a message that can be displayed on the page (e.g. a message alerting user of a successful action like form submit).
- A message element is to have a close button on the top-right corner.
- Create additional modifier classes for success (green color message background) and error (red color)
- Create a function that when passed a container element, message text and type (success/error), adds a new message element within the passed container (message is of appropriate type). It returns the new message element (jQuery collection object).
- Add a button to the page, clicking which adds a message to the page.
- Clicking on the close button of a message should remove it from the page.
- Make sure that the close button is still functional if the removed message element was added back to the page.





*Quiz 15*

*Event Handling in jQuery*

# EVENT HANDLING IN JQUERY

---

- What are the arguments to the on( ) method?
- How will you remove an event handler set on an element? What if you want to remove all event handlers set on an element?
- What is the difference between trigger( ) and triggerHandler( )
- You would like to event handlers on elements with the .dialog class - even those that may be added in future. How can you achieve this?

```
➤ $(‘body’).on(‘click’, ‘.message-close’, function( $event ) {  
    console.log( this );  
    console.log( $event.target );  
    console.log( $event.currentTarget );  
    console.log( $event.delegateTarget );  
});
```

When is/are the event handler(s) executed? How many event handlers are set and on what elements (assume the page has 10 .message elements, each with 2 .message-close elements). What is logged in each of the above statements?

- Answer the above questions if the handlers were set as below.

```
$(‘.message-close’).on(‘click’, function( $event ) {  
    console.log( this );  
    console.log( $event.target );  
    console.log( $event.currentTarget );  
    console.log( $event.delegateTarget );  
});
```

*Experiment - 15*

*Event Handling in*

*jQuery*



# EVENT HANDLING IN JQUERY

---

- Define an Employee class. Create an array of 3 - 4 Employee objects.
- Create a table of employees by iterating through the array. Store the employee details as data in the table `$(()).data`). Each row has a ‘log details’ button.
- Set up a delegated event handler on the table. When the user clicks on the log details button, the data for the employee should be read from the data set on the table and log it to console. You can set a `data-*` attribute on the row with a way to identify the clicked row (say, `data-empid=“1001”` etc.)
- Make sure to namespace the click handling for the row (say using “`click.employee`”)



# Quiz 16

## *Ajax Requests in jQuery*

# AJAX REQUESTS IN JQUERY

---

- Fetch all messages by hitting the /messages endpoint. Handle success and failure.

```
$._____({  
    _____: _____,  
    _____: _____,  
    _____: _____  
});
```

- Send details of a message (which has to, subject, body). Handle success and failure. Add a header to indicate the message body contains JSON data, and expects JSON data as response.

```
$._____({  
    _____: _____,  
    _____: _____,  
    _____: _____,  
    _____: _____,  
    _____: _____  
});
```

# AJAX REQUESTS IN JQUERY

---

- Which `*.ajax*()` method is appropriate for each of the below situations? Which element is the event handler set up on?
- Log the error details to console for every Ajax requests being made by the app
- Add a fixed set of headers for each Ajax request made by the app

*Experiment - 16*

*Ajax Requests in  
jQuery*



# AJAX REQUESTS IN JQUERY

---



*Quiz 17*

*Important Utility Methods*

# IMPORTANT UTILITY METHODS

---

- Fill in the arguments as per requirements.
- Check if .message contains .message-body and vice versa

\$.\_\_\_\_\_\_ ( \_\_\_\_\_, \_\_\_\_\_ )

- We want an object that has properties from both obj1 and obj2. We want the extended object to refer to obj1, obj2. How does you answer differ if we want a new copy of these in the new object. What is the value of obj.c.c1?

```
var obj1 = {  
    a: { a1: 100, a2: 200 }, c: { c1: 1, c2: 2 },
```

```
};
```

```
var obj2 = {  
    b: { b1: 100, b2: 200 }, c: { c1: 3, c2: 4 },  
};
```

```
var obj = $.______ ( _____, _____, _____ );
```

# IMPORTANT UTILITY METHODS

---

- Fill in the arguments as per requirements.

- Check if [1,2,3] is an array or not

\$.\_\_\_\_\_\_ ( \_\_\_\_\_ )

- Parses a string str in JSON format

\$.\_\_\_\_\_\_ ( \_\_\_\_\_ )

- Remove leading and trailing spaces from the value of an input with id “first-name”

\$.\_\_\_\_\_\_ ( \$( \_\_\_\_\_ ).\_\_\_\_\_ () )

# ATTRIBUTIONS

---

- Abstract blue background from Freepik  
[http://www.freepik.com/free-vector/abstract-blue-background\\_859016.htm](http://www.freepik.com/free-vector/abstract-blue-background_859016.htm)