# Question 4a

Should we place JavaScript in the HTML head or body? Please specify the necessary arguments for your choice.

## Answer:

Javascript can be placed anywhere in code, either in head or body.

**Argument to place in head section**:

Since Javascript code should not be displayed on webpage, it should be placed in head section. If we place script in head section, the Javascript code is executed before the webpage is actually displayed which is not recommended behavior of a web page.

**Argument to place in body section**:

In general we should place <script> tag in the location on the page (somewhere inside the body) where we want the script to write its message. In fact, it’s common to put <script> tags just below the closing </body> tag. This approach makes sure the page is loaded and the visitor sees it before running any JavaScript.

**My choice**: It is better to write Javascript separately in a .js file rather than writing in HTML page.

# Question4b

Which JQuery event allows for proper handling of JavaScript code present in the HTML document head section? Why is this event handling required?

## Answer

As mentioned in above answer, if JavaScript code is placed in HTML head section, JS code is executed before webpage is displayed. And moreover if JavaScript is going to access any DOM element in body section, it is not accessible for JavaScript code (which is in head section) because HTML code is executed from top to bottom.

To overcome this problem, we should use $(document).ready() function and write Javascript code inside this function. The $(document).ready() function is a built-in jQuery function that waits until the HTML for a page loads before it runs the script.

If JavaScript code is placed just before the end of body section (</body>), there is no need for $(document).ready() function. This event handling is required only when JavaScript code is to be executed at beginning or somewhere in the middle of body section.

# Question4c

Provide example web application use cases that require:

· Asynchronous communication

· Synchronous communication

## Answer

Synchronous communication: Request and Response is made on same port. Hence when user requests something from a server, he has to wait till the response is made available.

For example: If you want to display some information from a database, you usually need to load a new web page i.e. when you search a database for information, you leave the search page and go to new page of results.

Use case: Amazon is typical example of online database system. When you search Amazon for books, you get a list of books that can fit into a single webpage usually 10 items. If you wish to see next 10 items, you need to jump from page to page. The user cannot perform any action until the page is completely loaded.

Asynchronous communication: Request and Response are made on different ports. Hence when user requests something from server, he need not wait for response. He can continue with other events on same page, since here response is made independent of request. The server provides information to user whenever it is available.

Example: Display new HTML content without reloading the page.

Use-case: Suppose a user is reading news, headlines on e-paper. If he wants to read some article, he can just click a headline and the article appears on same page without loading header, footer, banner, sidebar etc. The news story appears on same webpage without needing to reload.

Use case 2: Other example is Twitter or Facebook. When you open the application you can see bunch of posts fitting on the webpage. When you scroll down the webpage, more posts appear on the same page without reloading the web page or without loading a new page. Scroll again and new posts appear. It’s like never ending webpage.