EXAMINATION INSTRUCTIONS

This examination has 17 pages. Check that you have a complete paper.
Answer all questions.
You have 3 hours to complete this examination. Use your time accordingly.

READ AND OBSERVE THE FOLLOWING RULES:

► Names are pre-printed on the exam booklets. Ensure that you have YOUR exam.
► Sign, using your usual signature, in the space provided on the back cover.
► All of your writing must be handed in. This booklet must not be torn or mutilated in any way, and must not be taken from the examination room.
► Show all of your work in arriving at an answer, unless instructed otherwise. Partial credit will be awarded as appropriate.
► Candidates are not permitted to ask questions of the invigilators, except in cases of supposed errors or ambiguities in examination questions.
► CAUTION – Candidates guilty of any of the following, or similar, dishonest practices shall be immediately dismissed from the examination and shall be liable to disciplinary action.
  ♦ Having books, papers, calculators, computers, cellphones, or other memory aid or electronic device.
  ♦ Speaking or communicating with other candidates.
  ♦ Purposely exposing written papers to the view of other candidates. The plea of accident or forgetfulness shall not be received.

DO NOT WRITE BELOW THIS LINE!
(see midterm exam for sample questions for modules 1 and 2)
Module 3 - Question 1 [20 points, 4 points each]

For each item labelled (a) though (e), match it with its *best* description from items (1) through (9). Note that four of the items from the numbered list will not be used.

(a) AJAX GET request  
(b) AJAX POST request  
(c) callback function  
(d) `<script>` tag  
(e) route

(1) a way to uniquely identify an HTML element  
(2) a way for a client to obtain code from a server  
(3) a function to process a response from an AJAX request  
(4) a way for a client to send data to a server  
(5) a way to determine, using a URL, what page a server should respond with  
(6) a way to start up a web server  
(7) a way for a client to request data from a server  
(8) a function to translate to and from JSON  
(9) a way to encrypt data transmissions

Give you matches below:

(a) best matches with _____

(b) best matches with _____

(c) best matches with _____

(d) best matches with _____

(e) best matches with _____
Module 3 - Question 2 [20 points total]

Define a JavaScript function named `activityScore` with one parameter. Assume the function will be called with an array of `Numbers`. Define the function so that it creates an Object. This object should have a key equal to "score", and its paired value should be set to the sum of the array’s entries. Return the JSON blob encoding the Object.

Part 1 [1 point]
What does `activityScore( [ 17, 0, 2 ] )` return?

Part 2 [1 point]
What does `activityScore( [ 0 ] )` return?

Part 3 [1 point]
What does `activityScore( [ 2, 3, 4, 2, 3, 4 ] )` return?

Part 4 [17 points]
[ ] 17 points: perfect
[ ] 6 points: clearly wrong, some correct elements
[ ] 12 points: essentially correct but with small mistakes
[ ] 0 points: for anything else

Define, in JavaScript, the function `activityScore`: 

Module 3 - Question 3 [20 points total, 5 points]

Part 1 [10 points]
[ ] 10 points: perfect
[ ] 3 points: clearly wrong, some correct elements
[ ] 7 points: essentially correct but with small mistakes
[ ] 0 points: for anything else

Define a Python function named `serveHTML`. This function should include any annotations needed so that it responds to requests for `/m3q31`. To handle these requests, `serveHTML` should return the contents of the file named `answer.html`. You can assume the file exists and is in the same directory as the file containing this code. Write your answer below:

```python
import bottle
```

Part 2 [10 points]
[ ] 10 points: perfect
[ ] 3 points: clearly wrong, some correct elements
[ ] 7 points: essentially correct but with small mistakes
[ ] 0 points: for anything else

Define a Python function named `loginData`. `loginData` will have a single parameter. This parameter will be a JSON blob. Your function should convert the JSON blob into usable Python data. If both "username" and "password" are keys in that data, `loginData` should return `True`. If the data is missing at least one of these keys, `loginData` should return `False`. Write your answer below:

```python
import json
```
Module 3 - Question 4 [20 points total]

[ ] 20 points: perfect
[ ] 7 points: clearly wrong, some correct elements
[ ] 14 points: essentially correct but with small mistakes
[ ] 0 points: for anything else

Study this abbreviated HTML and JavaScript code:

```html
<html><head>...contents unimportant to question and so skipped...</head>
<body onload="getMessage();">
Today’s Important Message is: <div id="disp"></div><br/><br/>
How do you respond: <input type="text" id="secretResponse"/>
</body>

function getMessage() {
  ajaxPostRequest("/message", "notVerySecret", displayMessage);
}

function ajaxPostRequest(path, data, callback) {
  var req = new XMLHttpRequest();
  req.onreadystatechange = function() {
    if ((this.readyState === 4) && (this.status === 200)) {
      callback(this.response);
    }
  };
  req.open("POST", path);
  req.send(data);
}
// YOUR CODE WILL GO HERE

Complete the `displayMessage` function. The input to this function will be a JSON blob. Your function should first convert the parameter to usable JavaScript data. You can assume that JavaScript data will be an Object. Get the value associated with the "ans" key from that Object. Then, get the div element with an id of "disp". Update the element, so that it displays the value you found in the Object.

function displayMessage(response) {

Module 4 - Question 1 [20 points total]

[ ] 20 points: perfect
[ ] 14 points: essentially correct but with small mistakes
[ ] 7 points: clearly wrong, some correct elements
[ ] 0 points: for anything else

You are working for a restaurant that has a customer list, which includes dietary restrictions, such as vegetarian, vegan, shellfish allergy, nut allergy, lactose intolerance, etc. Assume, for the sake of this question, that each customer has at most one dietary restriction and that those customers with no restrictions have "none" entered in that field. An example of a customer record is ["Jane", "Goodall", "vegetarian"]

In JavaScript, define a comparator which orders customer records first by their dietary restriction, then by last name, and finally by first name.
Module 4 - Question 2  [20 points total]

Part A [10 points]
[ ] 10 points: perfect
[ ] 3 points: clearly wrong, some correct elements
[ ] 7 points: essentially correct but with small mistakes
[ ] 0 points: for anything else

What is an HTML injection attack and how can you guard against it?

Part B [10 points]
[ ] 10 points: perfect
[ ] 3 points: clearly wrong, some correct elements
[ ] 7 points: essentially correct but with small mistakes
[ ] 0 points: for anything else

What is a SQL injection attack and how can you guard against it?
Module 4 - Question 3 [20 points total]

Part 1 [10 points]

[ ] 10 points: perfect
[ ] 3 points: clearly wrong, some correct elements
[ ] 7 points: essentially correct but with small mistakes
[ ] 0 points: for anything else

Explain when and why it is better to use:

cursor.execute("SELECT * FROM tableName WHERE field = ?", (variable,))
rather than the simpler:

cursor.execute("SELECT * FROM tableName WHERE field = " + variable)

Part 2 [10 points]

[ ] 10 points: perfect
[ ] 3 points: clearly wrong, some correct elements
[ ] 7 points: essentially correct but with small mistakes
[ ] 0 points: for anything else

Briefly explain what the commit() function does and why it is important.
Module 4 - Question 4 [20 points total]

Here is the code for binary search:

```python
def binarySearch(X, Z):
    left = 0
    right = len(X)
    while (right - left) > 0:
        mid = (left + right) // 2
        print(mid)
        if Z < X[mid]:
            right = mid
        elif Z > X[mid]:
            left = mid + 1
        else:
            return True
    return False
```  

What is printed when the following code executes?

```python
data = [10, 13, 17, 21, 24, 28, 42, 45, 53, 59]
target = 51

print( binarySearch(data, target) )
```