

Computer Science 571 2nd Exam
Prof. Papa
Tuesday, December 2, 2014, 5:30pm – 6:50pm

Name:

Student ID Number:

1. This is a closed book exam.
2. Please answer all questions on the test

REST Questions [10 pts]

Each question is worth 2 points.

Q1: What are the 3 fundamental aspects of REST design patterns?

A1: client, server and resources

Q2: What are the 4 simple operations of REST?

A2: PUT, GET, POST, DELETE

Q3: What is the typical representation of resources in REST?

A3: documents

Q4: What is the major reason of the gain in popularity of REST versus other approaches?

A4: simplicity

Q5: REST:

- ☒ is Platform independent
- ☒ is Language Independent
- ☒ is based on the HTTP standard
- ☒ is able to run behind firewalls
- ☐ is a W3C recommendation
- ☐ is based on cookies
- ☒ does not offer built-in security
- ☐ ALL OF THE ABOVE

Web Performance Questions [10 pts]

Questions 1-5 are worth 1 point each.

Q1: Where should you put style sheets and why to optimize performance?

A1: At the top, because IE blocks rendering of the page until all style sheets have been examined.

Q2: Where should you put scripts and why to optimize performance?

A2: At the bottom, because scripts block everything from rendering below them in the page

Q3: Should you “inline” JavaScript code or use external files and why to optimize performance?

A3: Use external files because they are cached

Q4: Should you scale images or not, and why?

A3: No, because 1) rescaling in the browser is time consuming and 2) sending an image larger than needed transfers more bytes than necessary

Q5: What are the two rules that minimize the transfer of information between server and browser?

A5: compression and JavaScript minification

Q6. List any 5 recent rules (not the initial 14 rules) for faster Web pages from Yahoo’s YSlow that help speed up web performance. This question is worth 5 points.

A6:

Avoid empty src or href

Use get for AJAX requests

Reduce number of DOM elements

Avoid HTTP 404 (not found) Error

Reduce cookie size

Use cookie-free domain

Do not scale images in HTML

Make favicon small and cacheable

HTML5 Questions [10 pts]

Each question is worth 2 points.

Q1: In HTML5 is it possible to perform “document editing” just using HTML elements and attributes?

A1: No, JavaScript code using the HTML5 Document Editing API is required.

Q2: Which of the following capabilities have included in HTML5?

A2:

☒ canvas

☒ video and audio

☒ local SQL database

☒ geolocation

☒ CSS 2D/3D transformations

☐ flash plugin

☐ ALL OF THE ABOVE

Q3: Why have FRAMES been removed from HTML5?

A3: Frames have been removed from HTML5 because their usage affected usability and accessibility for the end user in a negative way in HTML4

Q4: What happened to the “align” attribute used on caption, iframe, img, input, and many other elements in HTML4?

A4: they have been removed from HTML5 and are handled by CSS exclusively

Q5: Consider the following HTML5:

```
<canvas id="myCanvas" width="200" height="100"
style="border:1px solid #000000;">
</canvas>
```

What is the purpose of the “id” attribute?

A5: It is used to obtain a “handle” to the canvas using getElementById so that a “drawing context” object can be create to draw on the canvas.

JSON /AJAX Questions [10 pts]

Each is worth 1 point.

Q1: Of the URLs below, which have the same origin?

- a. http://www.ajaxbook.com
- b. http://www.ajaxbook.com:8443
- c. https://www.ajaxbook.com
- d. http://ajaxbook.com
- e. http://carsearch.ajaxbook.com

A1: All of the above URLs have different origins

Q2: Creating an instance of the XMLHttpRequest object is the same in all browsers

- a. True
- b. False

A2: False

Q3: Which XMLHttpRequest method actually makes the request?

- a. getAllResponseHeaders()
- b. makeRequest()
- c. open()
- d. send()

A3: send()

Q4: What MIME type must be set when using XMLHttpRequest to send a POST request of a form?

- a. application/octet-stream
- b. Application/xhtml+xml
- c. application/x-www-form-urlencoded

A4: c

Q5: Which of the following MIME types will cause the web browser to automatically parse an XML response?

- a. text/xml
- b. text/javascript
- c. text/plain
- d. text/html

A5: a

Q6: Which JavaScript method is used primarily with JSON?

- a. `eval()`
- b. `test()`
- c. `run()`
- d. `exec()`

A6: a

Q7: Which field of XMLHttpRequest will be populated with a JSON response?

- a. `responseXML`
- b. `responseText`
- c. `response`
- d. `statusText`

A7: b

Q8: Which of the following is not valid JSON?

- a. `{ "name": "Bob" }`
- b. `{ "user": { " name": "Bob" } }`
- c. `{ "name": "Bob", "getName": function() { return this.name } }`

A8: c

Q9: The XMLHttpRequest object can be used to upload a file.

- a. `True`
- b. `False`

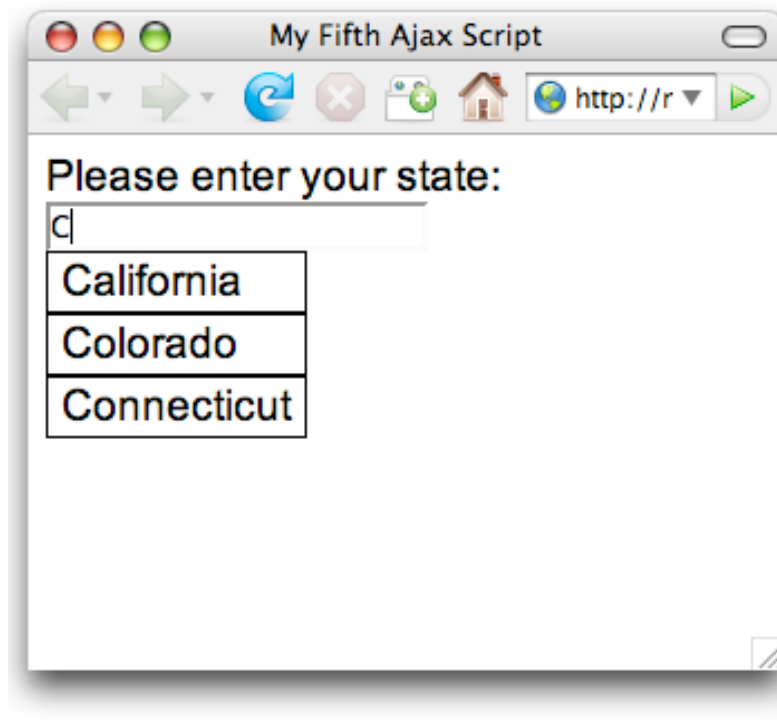
A9: b

Q10: The web browser displays a visual progress indicator while an XMLHttpRequest is being processed.

- a. `True`
- b. `False`

A10: b

JavaScript and Ajax Questions [10 pts]



Below is the HTML source code that produces the web page above. There is one edit box, and when each character is typed, a list appears that shows the “suggested” valid answers, obtained from the XML file us-states.xml.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN">
<html>
<head>
  <title>My Fifth Ajax Script</title>
  <link rel="stylesheet" rev="stylesheet" href="script05.css" />
  <script src="script05.js" type="text/javascript"
language="Javascript">
  </script>
</head>
<body>
  <form action="#">
    Please enter your state:<br />

    <input type="text" id="searchField" autocomplete="off" /><br />
    <div id="popups"> </div>
  </form>
</body>
</html>
```

Below is the JavaScript source code, script05.js, that was imported into the HTML above, but some of the lines are missing, replaced by XXXXXXs. Fill in the missing.

```
window.onload = initAll;
var xhr = false;
var statesArray = new Array();
```

```

function initAll() {
    document.getElementById("searchField").onkeyup = searchSuggest;

    if (window.XMLHttpRequest) {
        xhr = new XMLHttpRequest();
    }
    else {
        if (window.ActiveXObject) {
            try {
                xhr = new ActiveXObject("Microsoft.XMLHTTP");
            }
            catch (e) { }
        }
    }

    if (xhr) {
        xhr.onreadystatechange = setStatesArray;
        xhr.open("GET", "us-states.xml", true);
        xhr.send(null);
    }
    else {
        alert("Sorry, but I couldn't create an XMLHttpRequest");
    }
}

function setStatesArray() {
    if (xhr.readyState == 4) {
        if (xhr.status == 200) {
            if (xhr.responseXML) {
                var allStates =
xhr.responseXML.getElementsByTagName("item");
                for (var i=0; i<allStates.length; i++) {
                    statesArray[i] =
allStates[i].getElementsByTagName("label")[0].firstChild;
                }
            }
            else {
                alert("There was a problem with the request " +
xhr.status);
            }
        }
    }
}

function searchSuggest() {
    var str = document.getElementById("searchField").value;
    document.getElementById("searchField").className = "";
    if (str != "") {
        document.getElementById("popups").innerHTML = "";

        for (var i=0; i<statesArray.length; i++) {
            var thisState = statesArray[i].nodeValue;

            if (thisState.toLowerCase().indexOf(str.toLowerCase()) ==
0) {
                var tempDiv = document.createElement("div");

```

```

        tempDiv.innerHTML = thisState;
        tempDiv.onclick = makeChoice;
        tempDiv.className = "suggestions";

        document.getElementById("popups").appendChild(tempDiv);
    }
}
var foundCt =
document.getElementById("popups").childNodes.length;
    if (foundCt == 0) {
        document.getElementById("searchField").className =
"error";
    }
    if (foundCt == 1) {
        document.getElementById("searchField").value =
document.getElementById("popups").firstChild.innerHTML;
        document.getElementById("popups").innerHTML = "";
    }
}

function makeChoice(evt) {
    var thisDiv = (evt) ? evt.target : window.event.srcElement;
    document.getElementById("searchField").value = thisDiv.innerHTML;
    document.getElementById("popups").innerHTML = "";
}

```

[for graders: each line is worth 2 points]

JSON Question [20 pts]

The REST Yahoo LocalSearch Service allows you to search the Internet for businesses near a specified location, and returns both the latitude and longitude and Yahoo! user ratings of the establishment, as well as search by business categories.

A sample REST call is shown below, followed by an edited version of the XML returned:

<http://local.yahooapis.com/LocalSearchService/V3/localSearch?appid=YahooDemo&query=pizza&zip=94306&results=1&output=json>

```

<?xml version="1.0"?>
<ResultSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
    xmlns="urn:yahoo:lcl"
xsi:schemaLocation="urn:yahoo:lcl
    http://local.yahooapis.com/LocalSearchService/V3/Loca
lSearchResponse.xsd">
    <Result id="21300761">
        <Title>Oregano's Wood-Fired Pizza</Title>

```



```

        <Address>4546 El Camino Real, #A6</Address>
        <City>Los Altos</City>
        <State>CA</State>
        <Phone>(650) 941-3600</Phone>
        <Latitude>37.401434</Latitude>
        <Longitude>-122.114407</Longitude>
        <Rating>
            <AverageRating>4.5</AverageRating>
            <TotalRatings>11</TotalRatings>
            <TotalReviews>11</TotalReviews>

        <LastReviewDate>1326963606</LastReviewDate>
        <LastReviewIntro>This place is great.
    </LastReviewIntro>
    </Rating>
    <Distance>1.19</Distance>
    <Url>http://local.yahoo.com/info-21300761-oregano-s-woodfired-pizza-los-altos</Url>
</Result>
</ResultSet>

```

When the “output=json”, a JSON response is returned. Please fill in the missing code:

```

{"ResultSet":
  {"Result":
    {"id":"21300761",
     "Title":"Oregano's Wood-Fired Pizza",
     "Address":"4546 El Camino Real, #A6",
     "City":"Los Altos",
     "State":"CA",
     "Phone":"(650) 941-3600",
     "Latitude":"37.401434",
     "Longitude":"-122.114407",
     "Rating":
       {"AverageRating":"4.5",
        "TotalRatings":"11",
        "TotalReviews":"11",
        "LastReviewDate":"1326963606",
        "LastReviewIntro":"This place is great."},
        "Distance":"1.19",
        "Url":"http://local.yahoo.com/info-21300761-oregano-s-woodfired-pizza-los-altos",
      }
    }
  }
}

```

Grading Guidelines:

There are 15 properties so give 1 point for each property. Writing the opening and the closing parenthesis of the Rating object weighs 5 points.

JQuery Questions [10 pts]

Q1 (2 pts): What is the JavaScript object added by jQuery?

A1: jQuery or \$

Q2 (2 pts): What commonly used JavaScript objects are abstracted by jQuery?

A2:

☐ DOM

☐ XMLHttpRequest

☐ JSON

☒ ALL OF THE ABOVE

Q3 (6 pts): [This question is worth 6 points] Consider the following example without JQuery:

```
<FORM>
<INPUT ID="counter1" STYLE="position:relative;
    left:0px" TYPE="button" VALUE="Move Button right once"
    onclick="document.getElementById('counter1').style.left = '500px';">
</FORM>
```

```
<FORM> <INPUT ID="counter2" STYLE="position:relative;
    top:0px" TYPE="button" VALUE="Move Button down Once"
    onclick="document.getElementById('counter2').style.top =
'15px';">
</FORM>
```

A3: Rewrite it using JQuery.

```
<FORM>
<INPUT ID="counter1" STYLE="position:relative;
    left:0px" TYPE="button" VALUE="Move Button right once"
    onclick= "$('#counter1').css('left', '500px');" >
</FORM>

<FORM> <INPUT ID="counter2" STYLE="position:relative;
    top:0px" TYPE="button" VALUE="Move Button down Once"
    onclick= "$('#counter2').css('top', '15px');" >
</FORM>
```

Grading Guidelines:

- Writing \$('#counter1') twice → 2 points
 - If a student forget writing \$ or # deduct 1 point
- Writing .css twice → 2 points
- The CSS properties and its values → 2 points

If any student came with different solution might be correct please spread the solution to the other graders to take a decision

Responsive Website Design Questions [20 pts]

Each question is worth 4 points.

Q1: A site designed with RWD adapts the layout to the viewing environment by using 3 things. What are they?

A1:

- (1) fluid, proportion based GRIDS**
- (2) flexible images**
- (3) CSS3 media queries**

Q2: List 3 “usability” guidelines for Websites on mobile devices?

A2:

- (1)**
- (2)**
- (3)**

Any 3 of: (1) reduce amount of content, (2) use single columns, (3) change navigation, (4) minimize text entry, (5) design for touchscreens, (6) take advantage of built-in functionality (e.g. maps Apps)

Q3: What is the main difference between “adaptive” and “fluid” grids?

A3: In fluid grids we define relative-based dimensions; in adaptive grids we define pixel-based dimensions.

Q4: In RWD, how do you avoid having an image deformed due to screen size?

A4: Set CSS max-width property to 100%, as in

img { max-width: 100%; }

Q5: Bootstrap comes with 3 components, each store in its own folder. What are these components?

A5: JavaScript, CSS and fonts