

Computer Science 571 2nd Exam
Prof. Horowitz
Tuesday, December 3, 2013, 12:30pm – 1:45pm

Name:

Student ID Number:

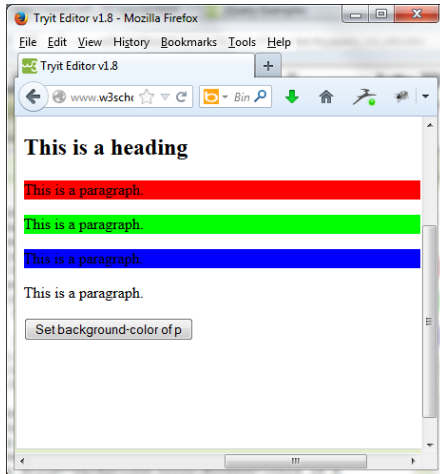
- 1. This is a closed book exam.**
- 2. Please answer all questions.**
- 3. Place all answers on the exam and return the entire exam**

Section	Student Grade	Max
jQuery		/20
Javascript + Ajax		/20
PHP + Regular Expressions		/15
Cookies and Privacy		/15
XML Schemas		/15
Web Security		/15
		/100

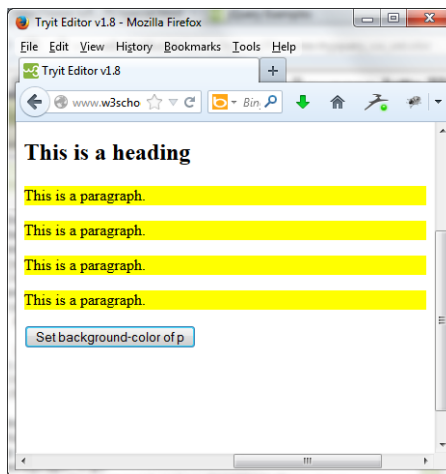
jQuery [20 pts]

1. [10 pts]

Below is a web page with three occurrences of the sentence “This is a paragraph”. The first sentence has a red background, the second has a green background and the third has a blue background. The button “Set background-color of p” resets the background color of all three sentences to yellow background. Below is the code that accomplishes this. Fill in the missing jQuery pieces.



Initial screen



all the same color

```
<!DOCTYPE html><html><head>
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">
</script>
<script>
.....
.....
.....
.....
.....
.....
.....
</script></head>
<body>
<h2>This is a heading</h2>
<p style="background-color:#ff0000">This is a paragraph.</p>
<p style="background-color:#00ff00">This is a paragraph.</p>
<p style="background-color:#0000ff">This is a paragraph.</p>
<p>This is a paragraph.</p>
<button>Set background-color of p</button></body></html>
```


JavaScript + Ajax [20 pts]

3. [10 pts]

JSONscriptRequest is a simple class for making HTTP requests using dynamically generated script tags and JSON. Below is a sample that uses the Yahoo service for returning Latitude and Longitude given a zip code. Fill in the missing code.

```
<script type="text/javascript" src="jsr_class.js"></script>

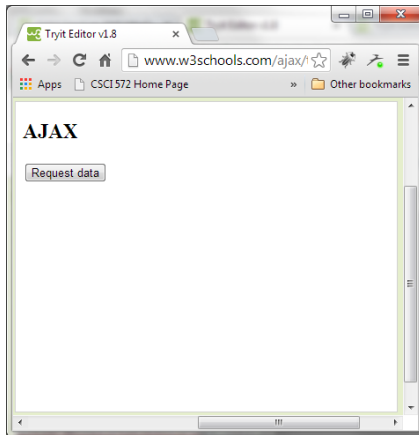
function callbackfunc(jsonData) {
    alert('Latitude = ' + jsonData.ResultSet.Result[0].Latitude +
        ' Longitude = ' + jsonData.ResultSet.Result[0].Longitude);
    aObj.removeScriptTag();
}

request =
'http://api.local.yahoo.com/MapsService/V1/geocode?appid=YahooDemo&
    output=json&callback=callbackfunc&location=78704';
aObj = new JSONscriptRequest(request);
aObj.buildScriptTag();
aObj.addScriptTag();

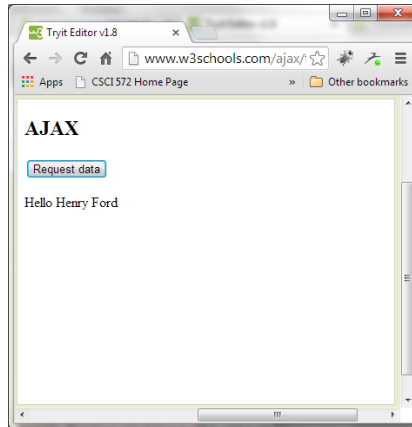
function JSONscriptRequest(fullUrl) {
    this.fullUrl = fullUrl;
    this.noCacheIE = '&noCacheIE=' + (new Date()).getTime();
    this.headLoc = document.getElementsByTagName("head").item(0);
    this.scriptId = 'JscriptId' + JSONscriptRequest.scriptCounter++;
}
JSONscriptRequest.prototype.buildScriptTag = function () {
    this.scriptObj = .....;
    this.scriptObj.setAttribute("type", .....);
    this.scriptObj.setAttribute("charset", .....);
    this.scriptObj.setAttribute(....., ..... + this.noCacheIE);
    this.scriptObj.setAttribute("id", this.scriptId);
}
```

4. [10 pts]

Below are two screenshots, one representing an initial page and a second showing what occurs after the “Request data” button is clicked. Below that is the code that produces the effect. Supply the missing code.



Initial Screen



After the button is clicked

```
<!DOCTYPE html><html><head><script>
function loadXMLDoc()
{
var xmlhttp;
if (window.XMLHttpRequest)
{
xmlhttp=new ..... ();
}
else
{
xmlhttp=new ActiveXObject(".....");
}

xmlhttp.onreadystatechange=function()
{
if (xmlhttp.readyState==4 && xmlhttp.status==200)
{
document.getElementById("myDiv").innerHTML=.....;
}
}

xmlhttp.open("POST","demo_post2.asp",true);
xmlhttp.setRequestHeader("Content-type",
.....);
xmlhttp.send("fname=Henry&lname=Ford");
}
</script></head><body>
<h2>AJAX</h2>
<button type="button" onclick="loadXMLDoc()">Request data</button>
<div id="myDiv"></div>
</body></html>
```

PHP and Regular Expressions [15 pts]

5. [5 pts]

Determine what the following PHP code outputs:

```
<?php
$str = 'foobar: 2008';
preg_match('/(?P<name>\w+): (?P<digit>\d+)/', $str, $matches);
print_r($matches);
?>
```

Answer:

.....

.....

.....

.....

.....

6. [5 pts]

Below is a PHP program; once you understand what the program does supply appropriate output messages

```
<?php
$string = firstname.lastname@aaa.bbb.com;
$regexp = "/^[^0-9][A-z0-9_]+([.][A-z0-9_]+)*[@][A-z0-9_]+([.][A-z0-9_]+)*[.][A-z]{2,4}$/" ;

if (preg_match($regexp, $string)) {
    echo .....;
} else {
    echo .....;
}
?>
```

7. [5 pts]

Determine what the following PHP function returns:

```
function doit($x)
{return preg_match('|^http(s)?://[a-z0-9-]+(\.[a-z0-9-]+)*(:[0-9]+)?|/.*)?$', $x);
}
```

a. What is the result when the input \$x is:

<http://www.usc.edu:a1234/index.html>

b. What is the result when the input \$x is:

<https://www.google.com>

c. In one sentence explain what the function does and what you would use it for?

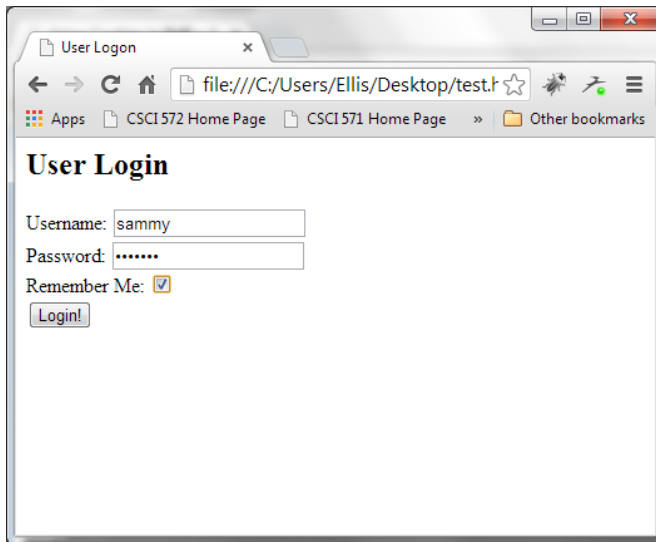
Answer:

a.

b.

c.

Cookies and Privacy [15 pts]



```
<html><head><title>User Logon</title></head>
<body>
  <h2>User Login </h2>
  <form name="login" method="post" action="login.php">
    Username: <input type="text" name="username"><br>
    Password: <input type="password" name="password"><br>
    Remember Me: <input type="checkbox" name="rememberme" value="1"><br>
    <input type="submit" name="submit" value="Login!">
  </form>
</body>
</html>
```

Below is the source code for login.php. Fill in the missing code and answer the questions below.

```
<?php
/* These are our valid username and passwords */
$user = 'jonny4';
$pass = 'delafoo';
```

```

if (isset($_POST['username']) && isset($_POST['password'])) {
    if (($_POST['username'] == $user) && ($_POST['password'] == $pass))
    {
        if (isset($_POST['.....'])) {
            setcookie('username', $_POST['username'],
                time()+60*60*24*365,
                '/account', 'www.example.com');
            setcookie('password', md5($_POST['password']),
                time()+60*60*24*365,
                '/account', 'www.example.com');
        }
        else
        {
            setcookie('username', $_POST['username'],
                false, '/account', 'www.example.com');
            setcookie('password', md5($_POST['password']),
                false, '/account', 'www.example.com');
        }
        header('Location: index.php');
    }
    else {
        echo '.....';
    }
}
else {
    echo 'You must supply a username and password.';
}
?>

```

8. [5 pts]

If *rememberme* is chosen, how long will the cookie last?

9. [5 pts]

If *rememberme* is NOT chosen, how long will the cookie last?

10. [5 pts]

What is md5?

XML Schemas [15 pts]

Below is an XML schema that describes a Bookstore. Answer the questions below.

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             targetNamespace="http://www.books.org"
             xmlns="http://www.books.org"
             elementFormDefault="qualified">
  <xsd:element name="BookStore">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="Book" minOccurs="1"
maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="Book">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="Title" minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="Author" minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="Date" minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="ISBN" minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="Publisher" minOccurs="1"
maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="Title" type="xsd:string"/>
  <xsd:element name="Author" type="xsd:string"/>
  <xsd:element name="Date" type="xsd:string"/>
  <xsd:element name="ISBN" type="xsd:string"/>
  <xsd:element name="Publisher" type="xsd:string"/>
</xsd:schema>
```

11. [5 pts]

a. what is the file suffix of this file?

Answer:

12. [5 pts]

b. what is the name of the default namespace?

Answer:

13. [5 pts]

c. What is the name of the data type defined above that can be exported to other Schema files?

Answer:

Web Security [15 pts]

Define the following terms:

14. [5 pts]

Brute force attack:

15. [5 pts]

Cross-site scripting (XSS)

16. [5 pts]

Injection Attack

The End
Good Luck