DOM QUESTIONS [25 Points]

In JavaScript, `arr.map(callback[, thisArg])` calls a provided callback function once for each element in an array, in order, and constructs a new array from the results. For example:

```javascript
var numbers = [1, 4, 9];
var roots = numbers.map(Math.sqrt);
/* roots is now [1, 2, 3], numbers is still [1, 4, 9] */
```

1. [5 pts] Here is a question about the use of map; fill in the missing code.

```javascript
var obj = {
    divisor: 10,
    remainder: function (value) {
        return value % this.divisor;
    }
}
var numbers = [6, 12, 25, 30];
var result = numbers.map(obj.remainder, obj);
document.write("output: <br />" + result);
// Output:
// 6,2,5,0
```
Above is a web page that was produced by the program below. But some of the code is missing. Fill in the missing code.

```html
<!DOCTYPE html><html lang="en"><head><title>width/height example</title>
<script>
function init() {
    var arrImages = new Array(3);
    arrImages[0] = document.getElementById("image1");
    arrImages[1] = document.getElementById("image2");
    arrImages[2] = document.getElementById("image3");
    var objOutput = document.getElementById("output");
    var strHtml = "<ul>"
        for (var i = 0; i < arrImages.length; i++) {
            strHtml += "<li>image" + (i+1) + 
                "; height=" + arrImages[i].height + 
                "; width=" + arrImages[i].width + 
                "; style.height=" + arrImages[i].style.height + 
                "; style.width=" + arrImages[i].style.width + 
                "</li>";
        }
    strHtml += "</ul>";
    objOutput.innerHTML = strHtml;
}
</script></head>
<body onload="init();">
<p>Image 1: no height, width, or style
    <img id="image1" src="http://www.mozilla.org/images/mozilla-banner.gif">
</p>
<p>Image 2: height="50", width="500", but no style
    <img id="image2" src="http://www.mozilla.org/images/mozilla-banner.gif" height="50" width="500"></p>
<p>Image 3: no height, width, but style="height: 50px; width: 500px;"
    <img id="image3" src="http://www.mozilla.org/images/mozilla-banner.gif" height="50" width="500"></p>
</body></html>
```
3. [10 pts] Below is the HTML for a web page and above is the output when the web page is displayed. The effect is produced by the JavaScript program below. Fill in the missing code.

```html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<head>
<title>Example</title>
</head>
<body>
<div>x</div>
<div>y</div>
<div>a</div>
<div>b</div>
<div>c</div>
<div>d</div>
<script type="text/javascript" src="example.js"></script>
</body>
</html>

Below is the JavaScript file example.js that produces the output above, but some of the code is missing. Fill in the missing code.
bodywalk = function(body) {
    var str = new Array();
    for (var i = 0; i < body.childNodes.length; i++) {
        if (body.childNodes[i].nodeName == 'DIV') {
            str.push(body.childNodes[i].innerHTML);
        }
    }
    return str;
}
var txtnodes = bodywalk(document.getElementsByTagName('body')[0]);
document.write("<br /><b>Content of DIV Tags: </b>" + txtnodes);

JavaScript Questions [50 Points]

[10 pts] Above to the left is a screenshot of the web page produced by the source code below and to the right is a new page containing the web page that results when the first selection is chosen. Fill in the missing code below

<script language="javascript" type="text/javascript">
function jumpto(x)
    if (document.form1.jumpmenu.value != "null") {
        window.open(x, '_blank');
    }
</script>
<p>Jump Menu:</p>
<form name="form1">
<select name="jumpmenu" onChange="jumpto(document.form1.jumpmenu.options[document.form1.jumpmenu.options.selectedIndex].value)"
    onChange="">
    <option>Jump to...</option>
</select>
</form>
[5 pts] What are the values of x(6) and y(7)?

```javascript
function makeAdder(a) {
    return function(b) {
        return a + b;
    };
}
x = makeAdder(5);
y = makeAdder(20);
x(6)
?
y(7)
?

Answer: x(6) returns 11 and y(7) returns 27; makeAdder(5) returns an anonymous function which adds 5 to the function argument; makeAdder(20) returns an anonymous function which adds 20 to the function argument;

The regular expression below includes three sets of parentheses (...). Explain in one sentence the pattern contained in each of the parentheses.

```
var pattern = "^([a-zA-Z_]+).([a-zA-Z]{2,3})$";
```

8. [5 pts] What is the purpose of the pattern within the first set of parentheses

Answer: One or more alphanumeric characters

9. [5 pts] What is the purpose of the pattern within the second set of parentheses

Answer: one or more alphabetic upper/lower case or underscore

10. [5 pts] What is the purpose of the pattern within the third set of parentheses

Answer: two or three alphabetic characters

11. [5 pts] What is the purpose of the ^ and the $
**Answer:** the `^` anchors the pattern at the start of the string,
The `$` anchors the pattern at the end of the string;

**[15 pts]** Below is a JavaScript function for evaluating a password. Fill in the missing code:

```javascript
function validatePassword() {
  var p = document.getElementById('newPassword').value,
      errors = [];

  if (p.length < 8) { // 12. [5 pts]
    errors.push("Your password must be at least 8 characters");
  }

  if (p.search(/[a-z]/i) < 0) { // 13. [5 pts]
    errors.push("Your password must contain at least one letter");
  }

  if (p.search(/[0-9]/) < 0) { // 14. [5 pts]
    errors.push("Your password must contain at least one digit");
  }

  if (errors.length > 0) {
    alert(errors.join("\n"));
    return false;
  }

  return true;
}
```

**HTTP Questions [20 Points]**

Below are the entire HTTP headers exchanged between the browser and www.google.com.

```json
https://www.google.com/
GET / HTTP/1.1
Host: www.google.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:26.0) Gecko/20100101 Firefox/26.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: PREF=ID=6071bff31af7b094:U=5b697a0afdbaf161:LD=en:TM=136436987:LM=1391892814:GM=1:SG=1:S=12CgY0xgmlOBnDi_; NID=67=PRYGZxwK39Rf_o6W8n61BnspMsOu9VAVKQcLziks5X5VtP4Uc2I9PJ-QtEnjxADGTYWHpivyvKLViU1uICU12SLJ-31-NJc5qGFrvc-A173da6137tUs3-p4RTG3Yht4kzxkpxVKFSWBDC0997593gM3DLMUUS5Sy2D5zVvS5vzB5kbrvJU1bn9W-B7nPrKrToc2_C7S0K08N3PjStFQFBY-FleJ6SFUKuge4;
SID=DQAAMUAAAB1XWv1F7NF08orDc=0Q0bu14Lp8OhoxFl6jzaBvUlgYHgt1ihMzrSPaDzADJ8ecaQMD0_YWSh eGmVdKzxBAY1RdCvzHBvQ3DthrMioiSktkZ_FL60a47C_Jk59YoF6KBbzoSKXO11bEdEdPjicBzQ3s7Lv6djar s9F6d679AB7OAGWwKtAMWCyf9DCyYwNc4nC1GFfrBvUXgUu_h00xDrJeMEux1nL51fK90jeQ3gQ8sn9Mm4Tby qxsGgJdwr4DPHrarkS8sonf16DRnrcV-; HSID=AGA2I2P09edxpHm2N; HSID=AdpAV2WHo091IXwB; APISID=D-Yuaw6HF3kXx9eCAlotmp7Xh0t8t8yq9D; SAFAPISID=LKdAgPoxwUrndV1/AIj79hu7WxrSir7Ve
Connection: keep-alive
```
15. [5 pts] What browser, name and version, participated in the exchange of HTTP data?

Answer: Firefox, version 26

16. [5 pts] List the names of all web servers involved in these transactions?

Answer: gws, GSE
17. [5 pts] What are the names of the cookies sent by the web servers to the client?

Answer: PREF

18. [5 pts] How many transactions between client and server occur?

Answer: 5

CGI Question [5 Points]

19. [5 points] In class we discussed a set of CGI environment variables. List any 5 and briefly describe their content.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOCUMENT_ROOT</td>
<td>The root directory of your server</td>
</tr>
<tr>
<td>HTTP_COOKIE</td>
<td>The visitor's cookie, if one is set</td>
</tr>
<tr>
<td>HTTP_HOST</td>
<td>The hostname of the page being attempted</td>
</tr>
<tr>
<td>HTTP_REFERER</td>
<td>The URL of the page that called your program</td>
</tr>
<tr>
<td>HTTP_USER_AGENT</td>
<td>The browser type of the visitor</td>
</tr>
<tr>
<td>HTTPS</td>
<td>&quot;on&quot; if the program is being called through a secure server</td>
</tr>
<tr>
<td>PATH</td>
<td>The system path your server is running under</td>
</tr>
<tr>
<td>QUERY_STRING</td>
<td>The query string (see GET, below)</td>
</tr>
<tr>
<td>REMOTE_ADDR</td>
<td>The IP address of the visitor</td>
</tr>
<tr>
<td>REMOTE_HOST</td>
<td>The hostname of the visitor (if your server has reverse-name-lookups on; otherwise this is the IP address again)</td>
</tr>
<tr>
<td>REMOTE_PORT</td>
<td>The port the visitor is connected to on the web server</td>
</tr>
<tr>
<td>REMOTE_USER</td>
<td>The visitor's username (for .htaccess-protected pages)</td>
</tr>
<tr>
<td>REQUEST_METHOD</td>
<td>GET or POST</td>
</tr>
<tr>
<td>REQUEST_URI</td>
<td>The interpreted pathname of the requested document or CGI (relative to the document root)</td>
</tr>
<tr>
<td>SCRIPT_FILENAME</td>
<td>The full pathname of the current CGI</td>
</tr>
<tr>
<td>SCRIPT_NAME</td>
<td>The interpreted pathname of the current CGI (relative to the document root)</td>
</tr>
<tr>
<td>SERVER_ADMIN</td>
<td>The email address for your server's webmaster</td>
</tr>
<tr>
<td>SERVER_NAME</td>
<td>Your server's fully qualified domain name (e.g. <a href="http://www.cgi101.com">www.cgi101.com</a>)</td>
</tr>
<tr>
<td>SERVER_PORT</td>
<td>The port number your server is listening on</td>
</tr>
<tr>
<td>SERVER_SOFTWARE</td>
<td>The server software you're using (e.g. Apache 1.3)</td>
</tr>
</tbody>
</table>