

ANSWER KEY

1. CSS Coding

```
#menu p {
    color: white;
    background-color: black;
    float: left;
    padding-left: 20px;
    padding-right: 20px;
    margin-right: 3px;
}

#pic {
    clear: left;
}

#pic, #score, #rating {
    float: left;
}

#score {
    width: 50%;
}

#rating {
    background-color: green;
    color: white;
    font-size: 30pt;
    padding: 20px;
    margin: 5px;
}

body {
    background-color: grey;
}

h1 {
    text-align: center;
}

#all {
    background-color: white;
    margin-left: auto;
    margin-right: auto;
    overflow: hidden;
    padding: 20px;
    width: 80%;
}
```

2. PHP

```
<?php
if (!isset($_GET["letter"]) || !isset($_GET["times"]) ||
    strlen($_GET["letter"]) != 1 || $_GET["times"] <= 0) {
    header("HTTP/1.1 400 Invalid Request");
    die("invalid request");
}

$letter = strtolower($_GET["letter"]);
$times = (int) $_GET["times"];

$matches = 0;
foreach (file("peeps.txt", FILE_IGNORE_NEW_LINES) as $line) {
    $lowerline = strtolower($line);
    $count = 0;
    for ($i = 0; $i < strlen($line); $i++) {
        $ch = $lowerline[$i];
        if ($ch == $letter) {
            $count++;
        }
    }
    if ($count >= $times) {
        $matches++;
        ?>
        <p><strong><?= $line ?></strong>
            contains '<?= $letter ?>' exactly <?= $times ?> times.</p>
        <?php
    }
}

if ($matches == 0) {
    ?>
    <p>No names contained '<?= $letter ?>' enough times.</p>
    <?php
}
?>
```

```

3.
<?php
$category = "";
if (isset($_GET["category"])) {
    $category = $_GET["category"];
} else {
    header("HTTP/1.1 400 missing parameter");
    die("ERROR 400: missing mandatory parameter category");
}

$file = file("books.txt");

$json_data = array(
    "count" => count($file),
    "books" => array()
);

foreach ($file as $line) {
    # Bachelor Chow in 20 Minutes|Gloria Demartelaere|cooking|2005|30.00
    list($title, $author, $book_category, $year, $price) = explode("|",
trim($line));
    if ($book_category == $category) {
        $json_data["books"][] = array(
            "category" => $category,
            "year" => $year,
            "price" => $price,
            "title" => $title,
            "author" => $author
        );
    }
}

header("Content-type: application/json");
print json_encode($json_data);
?>

```

3.

```
window.onload = function() {
    makeRequest("begin");
};

function makeRequest(nodeid) {
    var request = new XMLHttpRequest();
    request.onload = processData;
    request.open("GET", "adventure.php?nodeid=" + nodeid, true);
    request.send();
}

function processData() {
    var question = this.responseXML.querySelector("question");
    if(question) {
        document.getElementById("questionparagraph").innerHTML =
            question.textContent;
        var answers = this.responseXML.querySelectorAll("answer");
        document.getElementById("answer").innerHTML = "";
        for(var i = 0; i < answers.length; i++) {
            var button = document.createElement("button");
            button.innerHTML = answers[i].textContent;
            button.onclick = go;
            console.log(button);
            document.getElementById("answer").appendChild(button);
        }
    } else {
        document.getElementById("questionparagraph").innerHTML =
            this.responseXML.querySelector("died").textContent;
        document.getElementById("answer").innerHTML = "";
    }
}

function go() {
    makeRequest(this.innerHTML);
}
```

4. Regular Expressions

- a) `/^[0-9]{5}(-[0-9]{4})?$/`
- b) `^\$[0-9]*\.[0-9][0-9]$\`
- c) `<[a-zA-Z]+([a-zA-Z]+="[a-zA-Z]")*\>`

5. SQL

```
-- all the times a director has appeared in his own movie, twice

SELECT DISTINCT d.first_name, d.last_name
FROM directors d
JOIN movies_directors md ON md.director_id = d.id
JOIN roles r1 ON r1.movie_id = md.movie_id
JOIN roles r2 ON r2.movie_id = md.movie_id
JOIN actors a ON a.id = r1.actor_id AND a.id = r2.actor_id
WHERE a.first_name = d.first_name AND a.last_name = d.last_name
      AND r1.role < r2.role
ORDER BY d.last_name, d.first_name;
```