

## Solutions

### 1. CSS

```
#sidebar {
    width: 25%;
    padding: 15px;
    float: left;
    background-color: #EEE;
}
body {
    font-family: "Century Gothic", "Helvetica", sans-serif;
    background-image: url("background.jpg");
}
.caption {
    margin-top: 0;
    text-align: right;
}
h1 {
    text-align: center;
}
#main {
    background-color: #222;
    width: 70%;
    float: right;
    padding-bottom: 15px;
}
#main p, #main h1, #gallery {
    background-color: #EEE;
    padding: 5px;
    width: 90%;
    margin-left: auto;
    margin-right: auto;
}
#gallery {
    overflow: hidden;
}
#gallery img {
    height: 150px;
    float: right;
    margin: 5px;
}
```

## 2. PHP

```
<?php
    $allCats = glob("cats/*");
    $newCat = $allCats[array_rand($allCats)];
    if(isset($_COOKIE["cats"])) {
        setcookie("cats", $_COOKIE["cats"] . " " . $newCat, time() + 60 * 60);
    } else {
        setcookie("cats", $newCat, time() + 60 * 60);
    }
?>

<!DOCTYPE html>
<html>
    <head><title>cats</title></head>
    <body>
        <div>
            <?php
                $files = explode(" ", $_COOKIE["cats"]);
                foreach($files as $file) {
                    ?>
                        
                    <?php
                }
            ?>
        </div>
    </body>
</html>
```

### 3. PHP/XML

```
<?php
    if(!isset($_GET["shelter"])) {
        header("HTTP/1.0 400 Doesn't exist");
        die();
    }
    $param = $_GET["shelter"];
    $lines = file("pets.txt");
    $dom = new DOMDocument();
    $shelter = $dom->createElement($param);
    $dom->appendChild($shelter);
    $cats = $dom->createElement("cats");
    $shelter->appendChild($cats);
    $dogs = $dom->createElement("dogs");
    $shelter->appendChild($dogs);

    $catCount = 0;
    $dogCount = 0;
    foreach($lines as $line) {
        list($type, $name, $age, $currentShelter, $desc) = explode("|", $line);
        if($param == $currentShelter) {
            if($type == "cat") {
                $cat = $dom->createElement("cat");
                $cat->setAttribute("name", $name);
                $cat->setAttribute("age", $age);
                $cats->appendChild($cat);
                $catCount++;
            } else if($type == "dog") {
                $dog = $dom->createElement("dog");
                $dog->setAttribute("name", $name);
                $dog->setAttribute("age", $age);
                $dogs->appendChild($dog);
                $dogCount++;
            }
        }
    }
    $cats->setAttribute("count", $catCount);
    $dogs->setAttribute("count", $dogCount);
    header("Content-type: text/xml");
    print($dom->saveXML());
?>
```

#### 4. AJAX/JSON

```
window.onload = function() {
    document.getElementById("go").onclick = searchForPokemon;
    document.getElementById("randomize").onclick = getRandomPokemon;
};

function searchForPokemon() {
    var name = document.getElementById("pokemon").value;

    var ajax = new XMLHttpRequest();
    ajax.onload = display;

    ajax.open("GET", "pokedex.php?pokemon=" + name, true);
    ajax.send();
}

function getRandomPokemon() {
    var ajax = new XMLHttpRequest();
    ajax.onload = display;

    ajax.open("GET", "pokedex.php?random=true", true);
    ajax.send();
}

function display() {
    if(this.status == 200) {
        var pokeData = JSON.parse(this.responseText);

        document.getElementById("name").innerHTML = pokeData["name"];
        document.getElementById("pokepic").src = pokeData["images"]["photo"];
        document.getElementById("weakness").src =
pokeData["images"]["weaknessIcon"];
        document.getElementById("type").src = pokeData["images"]["typeIcon"];
        document.getElementById("info").innerHTML =
pokeData["info"]["description"];
    }else{
        document.getElementById("json").innerHTML = "error message goes here";
    }
}
```

## 5. Regex

- a) `<div( id="[a-zA-Z]+"?)( class="[a-zA-Z]+"?)*</div>/`
- b) `/foreach(\([a-zA-Z_]+ as \([a-zA-Z_]+\)) {/`
- c) `/^ATG([ATCG]{3})*(TAA|TAG|TGA)$/`

## 6. SQL

```
SELECT DISTINCT m.name
FROM actors a1, actors a2, roles r1, roles r2, movies m
WHERE a1.id = r1.actor_id AND a2.id = r2.actor_id AND
      r1.movie_id = m.id AND r2.movie_id = m.id AND
      a1.film_count > 1 and a2.film_count > 1 AND
      (a1.last_name <> a2.last_name OR a1.first_name <> a2.first_name)
ORDER BY m.name
```

```
SELECT DISTINCT m.name
FROM movies m
JOIN roles r1 ON r1.movie_id = m.id
JOIN roles r2 ON r2.movie_id = m.id
JOIN actors a1 ON a1.id = r1.actor_id
JOIN actors a2 ON a2.id = r2.actor_id
WHERE a1.film_count > 1 AND a2.film_count > 1 AND
      (a1.first_name <> a2.first_name OR a1.last_name <> a2.last_name)
ORDER BY m.name
```