Lab Exercise Solutions: HTML, JavaScript, D3.js, and PyQt5

Exercise 1: HTML Basics

Task

Create a simple HTML webpage that includes the following elements:

- A title in the browser tab that says **My First Web Page**.
- A main heading (`<h1>`) with the text **Welcome to HTML Basics**.
- A paragraph (`<p>`) explaining what HTML is.
- An unordered list (`<ul>`) of at least three topics you want to learn in web development.
- A hyperlink (`<a>`) to your favorite website.
- An image (`<img>`) with appropriate `src` and `alt` attributes.

Solution

Here is the HTML code for the webpage:

```
<!DOCTYPE html>
<html>
<head>
    <title>My First Web Page</title>
</head>
<body>
    <h1>Welcome to HTML Basics</h1>
    <p>HTML stands for HyperText Markup Language. It is the standard language for
       creating web pages.</p>
    <ul>
        <li>HTML</li>
        <li>CSS</li>
        <li>JavaScript</li>
    </ul>
    <p>Visit my favorite website: <a href="https://www.example.com">Example.com</a
       ></p>
    <img src="path_to_image.jpg" alt="Description of image" />
</body>
</html>
```

Exercise 2: Applying CSS Styles

Task

Enhance the HTML page from Exercise 1 by adding CSS styles:

- Change the background color of the page to a light shade.
- Set the font family of the entire page to Arial, sans-serif.
- Center the main heading.
- Change the color of the paragraph text to dark gray.
- Add a border around the image.

Solution

You can include CSS styles within a `<style>` tag in the `<head>` section:

```
<!DOCTYPE html>
<html>
<head>
    <title>My First Web Page</title>
    <style>
        body {
            background-color: #f0f0f0;
            font-family: Arial, sans-serif;
        }
        h1 {
            text-align: center;
        }
        p {
            color: #555555;
        }
        img {
            border: 2px solid #000000;
        }
    </style>
</head>
<body>
    <!-- Rest of the HTML content -->
</body>
</html>
```

Exercise 3: Adding Interactivity with JavaScript

Task

Enhance your HTML page by adding the following JavaScript functionality:

- Add a button that, when clicked, displays an alert message saying "Hello, World!".
- Change the text of the main heading to "You clicked the button!" when the button is clicked.

Solution

Add the following script to your HTML file:

```
<!DOCTYPE html>
<html>
<head>
    <title>My First Web Page</title>
    <style>
        /* Your CSS styles */
    </style>
    <script>
        function buttonClicked() {
            alert("Hello, World!");
            document.getElementById("main-heading").innerHTML = "You clicked the
                button!";
        }
    </script>
</head>
<body>
```

```
    <h1 id="main-heading">Welcome to HTML Basics</h1>
    <!-- Rest of the HTML content -->
    <button onclick="buttonClicked()">Click Me</button>
</body>
</html>
```

Exercise 4: Creating a Bar Chart with D3.js

Task

Using D3.js, create a simple bar chart that visualizes the following data: [10, 15, 20, 25, 30].

- Set up an SVG canvas with appropriate width and height.
- Bind the data to SVG rectangles to represent bars.
- Scale the bars appropriately within the SVG canvas.
- Add axes to the chart.

Solution

Include the D3.js library in your HTML file and add the following script:

```
<!DOCTYPE html>
<html>
<head>
    <title>D3.js Bar Chart</title>
    <script src="https://d3js.org/d3.v7.min.js"></script>
    <style>
        /* Optional CSS styles */
    </style>
</head>
<body>
    <script>
        var data = [10, 15, 20, 25, 30];

        var width = 500;
        var height = 300;
        var margin = { top: 20, right: 30, bottom: 30, left: 40 };

        var svg = d3.select("body")
            .append("svg")
            .attr("width", width)
            .attr("height", height);

        var xScale = d3.scaleBand()
            .domain(data.map(function(d, i) { return i; }))
            .range([margin.left, width - margin.right])
            .padding(0.1);

        var yScale = d3.scaleLinear()
            .domain([0, d3.max(data)])
            .nice()
            .range([height - margin.bottom, margin.top]);

        svg.selectAll("rect")
            .data(data)
            .enter()
            .append("rect")
            .attr("x", function(d, i) { return xScale(i); })
            .attr("y", function(d) { return yScale(d); })
            .attr("width", xScale.bandwidth())
            .attr("height", function(d) { return yScale(0) - yScale(d); })
            .attr("fill", "steelblue");

        var xAxis = d3.axisBottom(xScale).tickFormat(function(d, i) { return i +
            1; });
        var yAxis = d3.axisLeft(yScale);

        svg.append("g")
            .attr("transform", "translate(0," + (height - margin.bottom) + ")")
            .call(xAxis);

        svg.append("g")
            .attr("transform", "translate(" + margin.left + ",0)")
            .call(yAxis);
    </script>
</body>
</html>
```

Exercise 5: Building a Basic GUI with PyQt5

Task

Create a simple GUI application using PyQt5 that:

- Displays a window titled "Hello PyQt5".
- Contains a label that says "Welcome to PyQt5!".
- Has a button labeled "Click Me".
- When the button is clicked, the label text changes to "Button Clicked!".

Solution

Below is the Python code using PyQt5:

```
import sys
from PyQt5.QtWidgets import QApplication, QWidget, QLabel, QPushButton,
    QVBoxLayout

def on_button_clicked():
    label.setText('Button Clicked!')

app = QApplication(sys.argv)
window = QWidget()
window.setWindowTitle('Hello PyQt5')

layout = QVBoxLayout()

label = QLabel('Welcome to PyQt5!')
button = QPushButton('Click Me')
button.clicked.connect(on_button_clicked)
```

```
layout.addWidget(label)
layout.addWidget(button)

window.setLayout(layout)
window.show()
sys.exit(app.exec_())
```

Exercise 6: Integrating HTML, CSS, and JavaScript

Task

Create a webpage that:

- Has a styled button using CSS.
- When the button is hovered over, its color changes.
- When the button is clicked, a JavaScript function is called that changes the content of a paragraph to display the current date and time.

Solution

```
<!DOCTYPE html>
<html>
<head>
    <title>Interactive Webpage</title>
    <style>
        button {
            background-color: #008CBA;
            color: white;
            padding: 15px 32px;
            border: none;
            cursor: pointer;
        }
        button:hover {
            background-color: #006F8E;
        }
    </style>
    <script>
        function displayDateTime() {
            var now = new Date();
            document.getElementById("datetime").innerHTML = "Current date and time
                : " + now;
        }
    </script>
</head>
<body>
    <button onclick="displayDateTime()">Show Date and Time</button>
    <p id="datetime"></p>
</body>
</html>
```

Exercise 7: Manipulating the DOM with JavaScript

Task

Using JavaScript, create a webpage that:

- Contains a list of items (<ul>) with at least three list items.
- Has a button labeled “Add Item”.
- When the button is clicked, a new item is added to the list with the text “New Item”.

Solution

```
<!DOCTYPE html>
<html>
<head>
    <title>DOM Manipulation</title>
    <script>
        function addItem() {
            var ul = document.getElementById("item-list");
            var li = document.createElement("li");
            li.appendChild(document.createTextNode("New Item"));
            ul.appendChild(li);
        }
    </script>
</head>
<body>
    <ul id="item-list">
        <li>Item One</li>
        <li>Item Two</li>
        <li>Item Three</li>
    </ul>
    <button onclick="addItem()">Add Item</button>
</body>
</html>
```