Here are the responses to the questions in the image:

---

a) Difference between block-level and inline-level elements in HTML:

- **Block-level elements:**

  - Always start on a new line and take up the full width available.
  - Examples: `<div>`, `<p>`, `<h1>`, `<section>`, etc.
  - Used for structuring content (e.g., paragraphs, sections).
- **Inline-level elements:**

  - Do not start on a new line and only take up as much width as necessary.
  - Examples: `<span>`, `<a>`, `<strong>`, `<em>`, etc.
  - Used for formatting text or content within a block.

---

b) HTML code snippet for a form with a mandatory text input field and a submit button:

```
<form action="/submit" method="post">
 <label for="name">Name:</label>
 <input type="text" id="name" name="name" required>
 <button type="submit">Submit</button>
</form>
```

---

c) How `<canvas>` works and example:

- The `<canvas>` element is used to draw graphics (like charts, shapes, and animations) using JavaScript.

- It provides a 2D drawing context where developers can render visual content dynamically.

Example:

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid black;"></canvas>
<script>
  const canvas = document.getElementById('myCanvas');
  const ctx = canvas.getContext('2d');
  ctx.fillStyle = 'blue';
  ctx.fillRect(50, 20, 100, 50); // Draws a blue rectangle
</script>
```

---

d) What is the `<iframe>` element? Differences from `<embed>` and `<object>`:

- **`<iframe>`:**

  - Embeds another HTML document within the current page.
  - Commonly used to embed external web pages, videos (e.g., YouTube), or ads.
- **Differences:**

  - `<iframe>`: Specifically for embedding HTML documents.
  - `<embed>`: Used for embedding external content like audio, video, or plugins (e.g., Flash).
  - `<object>`: A general-purpose container for embedding external resources like PDFs, Flash, or images.

Example of `<iframe>`:

```
<iframe src="https://www.example.com" width="600" height="400"
style="border:none;"></iframe>
```

Here are the answers to the questions in the image:

---

a) Distinction between "padding" and "margin" in the CSS box model:

- **Padding:**

  - Space between the content of an element and its border.
  - Affects the inner spacing of the element.
- **Margin:**

  - Space outside the border, separating the element from other elements.
  - Affects the outer spacing of the element.

---

b) Differences between GET and POST methods in form submission:

- **GET:**

  - Sends data as URL parameters (query string).
  - Data is visible in the URL and has a size limit.
  - Suitable for retrieving data (e.g., search queries).

Example:
```
<form action="/search" method="get">
 <input type="text" name="query">
 <button type="submit">Search</button>
</form>
```

- **POST:**

  - Sends data in the request body, not visible in the URL.
  - Suitable for submitting sensitive or large data (e.g., login forms).

Example:
```
<form action="/submit" method="post">
 <input type="text" name="username">
 <button type="submit">Submit</button>
</form>
```

---

c) What HTML stands for and image linking:

- **HTML**: HyperText Markup Language.

Image tags for linking `contact.jpg` from `links.html`:

i) **Both in the same folder:**
```
<img src="contact.jpg" alt="Contact">
```

ii) **`contact.jpg` in a subfolder named "images":**
```
<img src="images/contact.jpg" alt="Contact">
```

iii) **`links.html` in "otherfiles":**
```
<img src="../contact.jpg" alt="Contact">
```

---

d) Differences between `id` and `class` attributes:

- **`id`:**

  - Uniquely identifies a single element on the page.
  - Used for JavaScript and CSS targeting specific elements.

Example:
```
<div id="header">Header</div>
```

- **`class`:**

  - Can be assigned to multiple elements.
  - Used to group elements for styling or behavior.

Example:

```
 <div class="content">Content 1</div>
<div class="content">Content 2</div>
```