Q.1 <!DOCTYPE html> is not a tag in HTML; it is called a Document Type Declaration (DTD). It is used to specify the version of HTML being used in the document. In HTML5, <!DOCTYPE html> is used to indicate that the document is written in HTML5.

The purpose of the Document Type Declaration is to tell the web browser about the version of HTML being used in the document. It helps the browser to render the page correctly and handle any outdated or non-standard HTML features in a backward-compatible manner.

Q.2 Semantic tags in HTML are HTML elements that carry meaning about the content they enclose. Instead of using generic <div> or <span> tags, semantic tags provide descriptive and meaningful names for different sections of a web page.

Some examples of semantic tags in HTML5 include <header>, <nav>, <main>, <article>, <section>, <aside>, <footer>, <figure>, <figcaption>, etc.

The need for semantic tags arises from the following reasons:

SEO and Accessibility: Search engines and screen readers use semantic tags to understand the structure and content of a web page better. Properly using semantic tags can improve search engine rankings and make the website more accessible to users with disabilities.

Code Readability: Semantic tags make the HTML code more self-explanatory and easier to understand for developers and designers. It enhances code readability and maintainability.

Styling and Layout: Semantic tags often come with default styling and layout properties. Using semantic tags allows developers to apply consistent styling without relying heavily on CSS classes and IDs.

Q.3 The main difference between HTML tags and elements is as follows:

HTML Tags: HTML tags are used to define the structure of an HTML document. They are enclosed in angle brackets (< >) and are usually used in pairs, consisting of an opening tag and a closing tag. The opening tag starts with < followed by the tag name, and the closing tag starts with </ followed by the tag name. For example: <p> is an opening tag for a paragraph, and </p> is the closing tag for the same paragraph.

HTML Elements: HTML elements are made up of HTML tags along with the content that comes between the opening and closing tags. An element consists of a start tag, the content, and an end tag. For example, <p>Hello, World!</p> is an HTML element where <p> is the start tag, "Hello, World!" is the content, and </p> is the end tag.

In summary, HTML tags are used to define the structure of an HTML document, and HTML elements are composed of tags and the content enclosed within those tags.

Q.4 Building a complete resume using HTML alone is beyond the scope of a simple response. However, I can provide you with a basic template for a resume using HTML. You can further customize and expand it according to your requirements.

```
<!DOCTYPE html>
<html>

<head>
    <title>My Resume</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 20px;
        }

        h1 {
            text-align: center;
            margin-bottom: 20px;
        }

        .section {
            margin-bottom: 20px;
        }

        .section h2 {
            margin-bottom: 5px;
        }

        .section p {
            margin: 0;
        }
    </style>
</head>

<body>
    <h1>John Doe</h1>

    <div class="section">
        <h2>Contact Information</h2>
        <p>Email: john.doe@example.com</p>
        <p>Phone: (123) 456-7890</p>
    </div>

    <div class="section">
        <h2>Education</h2>
        <p>Bachelor of Science in Computer Science</p>
        <p>XYZ University, City, Country</p>
        <p>Graduated: May 20XX</p>
    </div>

    <div class="section">
        <h2>Work Experience</h2>
        <p>Software Engineer</p>
        <p>ABC Company, City, Country</p>
        <p>June 20XX - Present</p>
        <p>Responsibilities:</p>
        <ul>
            <li>Developing and maintaining web applications</li>
            <li>Collaborating with cross-functional teams</li>
            <li>Testing and debugging code</li>
        </ul>
    </div>

    <div class="section">
        <h2>Skills</h2>
        <ul>
            <li>HTML, CSS, JavaScript</li>
            <li>Java, Python, C++</li>
            <li>Version Control (Git)</li>
        </ul>
    </div>

    <div class="section">
        <h2>Projects</h2>
        <p>Project 1: Description of project 1</p>
        <p>Project 2: Description of project 2</p>
    </div>
</body>

</html>
```

Q.6 Some advantages of HTML5 over its previous versions are:

New Semantic Elements: HTML5 introduced several new semantic elements such as <header>, <nav>, <section>, <article>, <footer>, etc. These elements help in creating a more structured and meaningful HTML document, making it easier for search engines and assistive technologies to understand the content.

Audio and Video Support: HTML5 introduced native support for audio and video elements. This eliminates the need for third-party plugins like Flash to play multimedia content on web pages.

Canvas and SVG: HTML5 introduced the <canvas> element, which allows dynamic rendering of graphics and animations using JavaScript. Additionally, SVG (Scalable Vector Graphics) is fully supported in HTML 5, enabling the use of vector graphics in web pages.

Form Enhancements: HTML5 introduced new input types and attributes for forms, such as date picker, email, number, range, etc. It also introduced form validation and built-in error messages, making form handling easier and more user-friendly.

Offline Support: HTML5 introduced the Application Cache and Local Storage features, allowing web applications to work offline and store data locally on the user’s device.

Improved Accessibility: HTML5 includes several accessibility features like the aria-* attributes and landmark roles, making it easier to create accessible web pages for users with disabilities.

Mobile-Friendly: HTML5 is designed to be mobile-friendly and responsive, allowing web pages to adapt to different screen sizes and devices.

Q.7 Creating a complete music player using only HTML is not possible, as HTML alone does not have the functionality to play audio files. However, I can provide you with a basic HTML template for embedding an audio player on a web page

```
<!DOCTYPE html>
<html>

<head>
    <title>Music Player</title>
</head>

<body>
    <h1>My Music Player</h1>
    <audio controls>
        <source src="path/to/your/audio/file.mp3" type="audio/mpeg">
        Your browser does not support the audio element.
    </audio>
</body>

</html>
```

In this template, the <audio> element is used to embed an audio player on the web page. The controls attribute adds basic playback controls (play, pause, volume, etc.) to the player. The <source> element inside the <audio> element specifies the path to the audio file and its type.

Q.8 The main difference between the <figure> tag and the <img> tag is:

<figure> tag: The <figure> tag is a semantic element used to encapsulate media content, such as images, videos, audio, etc., along with their captions. It is often used with the <figcaption> tag to add a caption or description to the media content. It helps in associating the caption with the media and improves accessibility.

```
<figure>
    <img src="image.jpg" alt="Image description">
    <figcaption>This is an example image</figcaption>
</figure>
```

<img> tag: The <img> tag is a self-closing tag used to embed images in an HTML document. It does not require a closing tag and is used to display a single image on the web page.
Example:

```
<img src="image.jpg" alt="Image description">
```

Q.9 In HTML, a tag is used to define elements on a web page, while an attribute provides additional information about an element. Tags are enclosed in angle brackets (< >), and attributes are added to the opening tag of an element as name-value pairs.

```
<p>This is a paragraph.</p>
```

In this example, <p> is the tag used to define a paragraph element.

Example of an HTML tag with an attribute

```
<img src="image.jpg" alt="Image description">
```

In this example, <img> is the tag used to define an image element, and src and alt are attributes of the image element. The src attribute specifies the path to the image file, and the alt attribute provides an alternative text for the image.

Some global attributes that can be used with various HTML tags are:

class: Specifies one or more class names for an element (used for CSS styling and JavaScript).
id: Specifies a unique identifier for an element.
style: Specifies inline CSS styles for an element.

title: Specifies extra information about an element (displayed as a tooltip).

data-*: Custom data attributes that allow storing extra information within an HTML element for use in JavaScript or CSS. For example: data-attributeName="value".

aria-*: Accessibility attributes used to enhance the accessibility of web content for users with disabilities. For example: aria-label, aria-labelledby, etc.