Unit 3: HTML 5 and Features:

1. **Introduction:**
   HTML5 is the newest version (fifth) of HTML. It is the updated version of HTML which has new elements and attributes. It supports new set of technologies to build more powerful and complex websites, and applications. HTML5 has introduce Application Programming Interface (API) and Document Object Model (DOM).

2. **Difference between HTML and HTML5**
   HTML was originally designed to describe the scientific documents, now it has evolved to describe much more. HTML first version was developed in 1993, now most of the web today uses HTML 4. However, it still has limitations.
   Biggest was if the developers or designers wanted to add content or feature to their site that were not supported by HTML, they have to use non-standard proprietary technologies like Adobe Flash. These technologies required the users to install the browser plugins so to access the content or the feature. Even-though some users could not access the content or features.
   HTML5 was designed to cut the need of these technologies. Not only this, it supports offline browsing, supports high definition video and animations, etc.

3. **Advantages of HTML5.Why use HTML5?**
   - It has several advanced features that make it simple and engaging for both users and designers/developers.
   - You may play both audio and video files with it.
   - You can sketch on a canvas using it.
   - It makes it easier for you to create offline web applications and better form designs.
   - It gives you advanced features for which you would typically need to write JavaScript.

4. **Disadvantages of HTML5**
   - It is only supported by modern browsers.
   - Code may be long which is time consuming.

5. **HTML5 New Semantics Elements (Header, Footer, Section)**

HTML5 declaration: <!doctype html>
It is used to inform the browser about the fifth version of HTML used in the document

HTML5 tags can be classified into two types:

a. Semantic Elements:
   These elements have meaningful names that describes the type of content.
   Semantic Elements have a simple and clear meaning for both, the browser and developer.
   Examples: <form>, <table>, <header>, <footer>, etc.

b. Non-Semantic Elements:
   These elements don't have meaningful names to describe the type of content.

Examples: , , etc.

Why use semantic elements??

First block of codes using semantic elements:

```
<header></header>
<section>
    <article>
        <figure>
            <img>
            <figcaption></figcaption>
        </figure>
    </article>
</section>
<footer></footer>
```

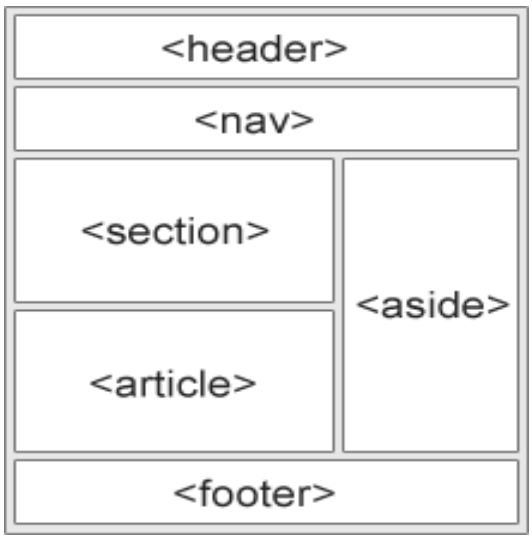
Second block of codes using non-semantic elements:

```
<div id="header"></div>
<div class="section">
    <div class="article">
        <div class="figure">
            <img>
            <div class="figcaption"></div>
        </div>
    </div>
</div>
<div id="footer"></div>
```

Reasons:
1.Easier to read using semantic elements. For very long chunk(thousands) of line codes, it is easier to read and understand the codes.
2. search engines and assistive technologies (screen reader) are also able to understand the context and content of the website.

Some of the semantic elements:

1. Header (syllabus)
2. Section (syllabus)
3. Footer (syllabus)
4. Aside
5. Nav
6. Article



Header:

- Header tag (<header>) is used to define the header of the document.

- It is a section of a HTML page which contains information related to the title and heading of the document
- <header> elements usually contain:
    - One or more heading elements (<h1> - <h6>).
    - Logo or icon.
    - Authorship information.
- It is a container tag.
- Syntax: <header> header contents </header>
- Note: this tag cannot be placed within a <footer>, <address> or other <header>

Code:

```
<header>
 <h1>Company A</h1>
 <ul>
  <li><a href="/home">Home</a></li>
  <li><a href="/about">About</a></li>
  <li><a href="/contact">Contact us</a></li>
 </ul>
 <form target="/search">
  <input name="q" type="search" />
  <input type="submit" />
 </form>
</header>
```

Section:

- Section tag <section> is used to define the section of the web page documents.
- It divides the content into sections and subsections.
- Its advantage is when there is requirement of two headers or footer or any other section in a document, section tag is used.
- It is a container tag.
- Syntax: <section> section contents </section>

Article:

- It is used to define the article content within a web page document.
- Article may be any forum post, magazine, newspaper article or, a story.
- It is a container tag.
- Syntax: <article> article content </article>

Code for section and article:

```
<section>
 <p>Top Stories</p>
 <section>
  <p>News</p>
  <article>Story 1</article>
  <article>Story 2</article>
  <article>Story 3</article>
```

```
  </section>
  <section>
    <p>Sport</p>
    <article>Story 1</article>
    <article>Story 2</article>
    <article>Story 3</article>
  </section>
</section>
```

Nav:

- Nav tag <nav> is used to define navigational sections in the documents.
- It organizes and indicates the links of the webpage to facilitate the user in navigation.
- It is used for primarily in navigation areas such as menu, table of contents, or indexes.
- It is a container tag.
- Syntax: <nav>links</nav>

Code for NAV:

```
<nav>
  <ul>
    <li><a href="/home">Home</a></li>
    <li><a href="/about">About</a></li>
    <li><a href="/contact">Contact us</a></li>
  </ul>
</nav>
```

Footer:

- It is used to define the footer of a document.
- It contains the footer information
    - author information
    - copyright information
    - contact information
    - back to top links, etc.
- It is a container tag.
- Syntax: <footer> footer contents </footer>

Note: It is used inside the body tag.

Code:

```
<!DOCTYPE html>
<html>
<body>
<footer>
  <p>Author: Hege Refsnes</p>
  <p><a href="mailto:hege@example.com">hege@example.com</a></p>
</footer>
</body>
</html>
```

Aside:

- It is used to describe the main object of the web page (like a highlighter).
- Aside contents represent primary content but not the main content of the webpage.
- It may contain information like author information, links, etc.
- It is a container tag.
- Syntax: <aside> content </aside>

Code:

```
<p>My family and I visited The Epcot center this summer. The weather was nice, and Epcot was amazing! I had a great summer together with my family!</p>
<aside>
<h4>Epcot Center</h4>
<p>Epcot is a theme park at Walt Disney World Resort featuring exciting attractions, international pavilions, award-winning fireworks and seasonal special events.</p>
</aside>
```

Here: it does not set aside contents to the side of the page; however, it tells the developer that the content is primary, a highlight.

Code using CSS;

```
<html>
<head>
<style>
aside {
  width: 30%;
  padding-left: 15px;
  margin-left: 15px;
  float: right;
  font-style: italic;
  background-color: lightgray;
}
</style>
</head>
<body>
<p>My family and I visited The Epcot center this summer. The weather was nice, and Epcot was amazing! I had a great summer together with my family!</p>
<aside>
<p>The Epcot center is a theme park at Walt Disney World Resort featuring exciting attractions, international pavilions, award-winning fireworks and seasonal special events.</p>
</aside>
<p>My family and I visited The Epcot center this summer. The weather was nice, and Epcot was amazing! I had a great summer together with my family!</p>
<p>My family and I visited The Epcot center this summer. The weather was nice, and Epcot was amazing! I had a great summer together with my family!</p>
</body>
</html>
```

6. HTML5 New Features:

1. Semantic Elements: Header, section and footer tags, Figure and figcaption, (already explained above)
2. Multimedia tags: audio and video tags.
3. Input types: email, url, required, color, date, tel, time, datetime, month, range, number, etc.
4. Form tags:
5. Graphic tags:

Multimedia tags:

Audio and Video:

One of the HTML5 feature is the support for audio and video. It allows the developers to embed video and audio on the webpage. Before, there was a hassle of required a third-party application/ service (Adobe Flash Player) to support the audio and video in a webpage. Using audio and video tag, the problem has been solved.

Both the tags are container tag.

Syntax:

<audio> </audio>

<video> </video>

Code for video:

<video width = "300" height = "200" controls autoplay>
<source src = "./dog.mp4" type ="video/mp4" />
</video>

Code for audio:

<audio controls>
<source src="dog.mp3" type="audio/mp3">
</audio>

Form Input type features:

These are the new features added to forms. These input tags introduced in HTML5 are:

color	Used to define an input field to indicate a color selector.
date	Used to define an input field to indicate a date selector.
datetime	Used to display date and time along with the time zone information.
datetime-local	Used to display date and time without the time zone information.
email	Used to specify an input field with an email pattern validation property.
month	Used to specify an input field to enter month for the particular year.

number	Used to specify a field that accepts a numeric value only.
range	Used to create a numeric value selector for a range of 1 to 100.
search	Used to create a search field.
tel	Used to define a control to enter a telephone number.
time	Used to define a control to enter time value with no time zone.
url	Used to define an input field to enter a URL.
week	Used to create a week value selector for a particular year.

Form tags and attributes: output and datalist tag, and placeholder, autofocus, required attribute:

Output tag: It is used to specify the output of a calculation or an outcome of the user.
Container tag
Syntax: <output></output>
Code for mathematical calculation:

```
<p>Multiplication:</p>
  <form oninput="a3.value = a1.valueAsNumber * a2.valueAsNumber">
    a1:<input type="number" id="a1"> *
    a2:<input type="number" id="a2"> =
    <output name="a3" for="a1 a2"> </output>
  </form>

Range slider value output:
<form oninput="a1.value = a2.value">
    <input type="range" id="a2" value="10">
    <br /><br />
    The value is <output name="a1" for="a2">10</output>
  </form>

Adding output of slider and input from user:
<form oninput = "result.value = parseInt(a1.value)+parseInt(a2.value)">
    <input type = "range" name = "a1" value = "0" /> +
    <input type = "number" name = "a2" value = "5" /> <br />
    The output is: <output name = "result"></output>
  </form>
```

Datalist tag:
It provides a list of predefined options to the user to select the data.
It is container tag, syntax: <datalist> elements </datalist>

Code for datalist:
Enter your favorite cricket player:

```
<input type="text" id="favCktPlayer" list="CktPlayers">
<datalist id="CktPlayers">
<option value="Sachin Tendulkar">
<option value="Brian Lara">
<option value="Jacques Kallis">
<option value="Ricky Ponting">
<option value="Rahul Dravid">
<option value="Shane Warne">
<option value="Rohit Sharma">
<option value="Donald Bradman">
<option value="Saurav Ganguly ">
<option value="AB diVilliers">
<option value="Mahendra Singh Dhoni">
<option value="Adam Gilchrist">
</datalist>
```

Placeholder attribute:
Placeholder is a hint provided to the user for the input to be entered in the input text.
Syntax:
<input type=”text” name=”fname” placeholder=”Ganesh”>
<input type="email" name="email" placeholder="email@example.com">

Autofocus attribute:
It automatically focusses on a particular form field.
Syntax: <input type=”text” name=”lastname” autofocus>

Required attribute:
It is used for the validation of user to provide a value for particular form field. User cannot submit the form unless providing the value for certain form field.
Syntax: <input type=”text” name=”lastname” required>

Graphics tag: canvas and svg tag:

Canvas tag:
It is used to draw graphics on webpage; however, JavaScript must be used in addition.
It is a container tag.
Syntax: <canvas> canvas elements </canvas>
It contains two attributes: width and height.

Code for canvas:

```
<canvas id = "mycanvas" width = "100" height = "100"></canvas>
```

Using inline style to draw a rectangle;

```
<canvas id="geeks"
        height="200"
        width="200"
        style="border:1px solid black">
  </canvas>
```

Drawing circle using Javascript;

```
<canvas id="geeks"
        height="200"
        width="200"
        style="border:1px solid black">
  </canvas>
<script>
      var c = document.getElementById("geeks");
      var cx = c.getContext("2d");
      cx.beginPath();
      cx.arc(100, 100, 90, 0, 2 * Math.PI);
      cx.stroke();
  </script>
```

Svg tag:

Svg stands for scalable vector graphics. It is used to draw vector type diagrams like pie-charts, two dimensional graphs, shapes, etc.

It is a container tag.

Syntax: <svg> svg elements </svg>