Republic of the Philippines
Department of Education
REGION III
SCHOOLS DIVISION OFFICE OF NUEVA ECIJA

LEARNING ACTIVITY SHEET
SPECIAL PROGRAM IN ICT 9
WEB DESIGN 9
Third Quarter, Week 10

Name of Learner: _____

Grade Level /Section: _____ Date: _____

COMBINATION OF SEMANTIC AND NON-SEMANTIC HTML

BACKGROUND INFORMATION FOR LEARNERS

HTML was originally created as a mark-up language to describe documents on the early internet. As the internet grew and was adopted by more people, its needs also changed. The internet was originally intended for sharing scientific documents, now people wanted to share other things as well. Very quickly, people started wanting to make the web look nicer. Because the web was not initially built to be designed, programmers used different hacks to get things laid out in different ways. Rather than using the <table></table> to describe information using a table, programmers would use them to position other elements on a page. As the use of visually designed layouts progressed, programmers started to use a generic "non-semantic" tag like . They would often give these elements a class or id attribute to describe their purpose. For example, instead of <header> this was often written as . As HTML5 is still relatively new, this use of non-semantic elements is still very common on websites today.

Semantic HTML or semantic mark-up is HTML that introduces meaning to the web page rather than just presentation. These elements simply mean, elements with meaning. The reason being, there definition in the code tells the browser and the developer what they are supposed to do. Framing in simpler words, these elements describe the type of content they are supposed to contain. .For example, a <p> tag indicates that the enclosed text is a paragraph. This is both semantic and presentational because people know what paragraphs are, and browsers know how to display them. On the flip side of this equation, tags such as <b> and <i> are not semantic. They define only how the text should look (bold or italic), and don't provide any additional meaning to the markup.

On the other hand a Non-Semantic elements unlike, semantic elements they don't have any meaning. They don't tell anything about the content they contain. They can be used with different attributes to mark up semantics common to a group. Traditionally, developers have implemented non-semantic elements with a class attribute to define the structure and express the meaning of content. It tells nothing about its content. The element has no special meaning at all. It represents its children. It can be used with the class, lang, and title attributes to mark up semantics common to a group of consecutive elements.

The table below shows the difference between a semantic and a non-semantic element of HTML.

SEMANTIC ELEMENTS	NON-SEMANTIC ELEMENTS
they have meaning	they don't have meaning
they describe how the content within them is supposed to behave	they can contain anything
they have specific attributes for their structure	'class' attribute can be used to work with their structure

Table 1: Difference between semantic and non-semantic elements

Why You Should Care About Semantics?

The benefit of writing semantic HTML stems from what should be the driving goal of any web page: the desire to communicate. By adding semantic tags to your document, you provide additional information about that document, which aids in communication. Specifically, semantic tags make it clear to the browser what the meaning of a page and its content is. That clarity is also communicated with search engines, ensuring that the right pages are delivered for the right queries.

Semantic HTML tags provide information about the contents of those tags that goes beyond just how they look on a page. Text that is enclosed in the <code> tag is immediately recognized by the browser as some type of coding language. Instead of trying to render that code, the browser understands that you are using that text as an example of the code for the purposes of an article or online tutorial.

Using semantic tags gives you many more hooks for styling your content, too. Perhaps today you prefer to have your code samples display in the default browser style, but tomorrow, you might want to call them out with a gray background color; later still, you might want to define the precise mono-spaced font family or font stack to use for your samples. You can do all of these things easily by using semantic markup and smartly applied CSS.

HERE ARE SOME OF HTML SEMANTIC ELEMENT

1. HTML <header> Element

The **HTML <header> element** represents a container for introductory content or a set of navigational links. A <header> element typically contains; (a) one or more heading elements (<h1> - <h6>), (b) logo or icon and (c) authorship information. You can have several <header> elements in one HTML document. However, <header> cannot be placed within a <footer>, <address> or another <header> element.

Example: A header for an <article>

```
<article>
  <header>
    <h1>What Does WWF Do?</h1>
    <p>WWF's mission:</p>
  </header>
  <p>WWF's mission is to stop the degradation of our planet's natural
environment,
  and build a future in which humans live in harmony with nature.</p>
</article>
```

2. HTML <nav> Element

The **HTML <nav> element** defines a set of navigation links. Notice that NOT all links of a document should be inside a <nav> element. The <nav> element is intended only for major block of navigation links. Browsers, such as screen readers for disabled users, can use this element to determine whether to omit the initial rendering of this content.

Example: A set of navigation links:

```
<nav>
  <a href="/html/">HTML</a> |
  <a href="/css/">CSS</a> |
  <a href="/js/">JavaScript</a> |
  <a href="/jquery/">jQuery</a>
```

3. HTML <main> Tag

The **HTML <main> element** represents the dominant content of the <body> of a document. The main content area consists of content that is directly related to or expands upon the central topic of a document, or the central functionality of an application.

Example: Specify the main content of the document

```
<main>
  <h1>Most Popular Browsers</h1>
  <p>Chrome, Firefox, and Edge are the most used browsers today.</p>

  <article>
    <h2>Google Chrome</h2>
    <p>Google Chrome is a web browser developed by Google, released in 2008.
Chrome is the world's most popular web browser today!</p>
  </article>

  <article>
    <h2>Mozilla Firefox</h2>
    <p>Mozilla Firefox is an open-source web browser developed by Mozilla.
Firefox has been the second most popular web browser since January, 2018.</p>
  </article>

  <article>
    <h2>Microsoft Edge</h2>
    <p>Microsoft Edge is a web browser developed by Microsoft, released in 2015.
Microsoft Edge replaced Internet Explorer.</p>
  </article>
</main>
```

4. HTML <footer> Element

The **HTML <footer> element** defines a footer for a document or section. A <footer> element typically contains: (a) authorship information, (b) copyright information, (c) contact information, (d) sitemap, (e) back to top links, and (f) related documents. You can have several <footer> elements in one document.

Example: A footer section in a document

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

5. HTML <article> Element

The **HTML <article> element** specifies independent, self-contained content. An article should make sense on its own, and it should be possible to distribute it independently from the rest of the web site. Examples of where an <article> element can be used are forum post, blog post, and newspaper article. The <main> tag specifies the main content of a document. The content inside the <main> element should be unique to the document. It should not contain any content that is repeated across documents such as sidebars, navigation links, copyright information, site logos, and search forms. There must not be more than one <main> element in a document. The <main> element must NOT be a descendant of an <article>, <aside>, <footer>, <header>, or <nav> element.

The content of a `<main>` element should be unique to the document. Content that is repeated across a set of documents or document sections such as sidebars, navigation links, copyright information, site logos, and search forms shouldn't be included unless the search form is the main function of the page. `<main>` doesn't contribute to the document's outline; that is, unlike elements such as `<body>`, headings such as `<h2>`, and such, `<main>` doesn't affect the DOM's concept of the structure of the page. It's strictly informative.

Example 1: Three articles with independent, self-contained content

```
<article>
<h2>Google Chrome</h2>
<p>Google Chrome is a web browser developed by Google, released in 2008. Chrome
is the world's most popular web browser today!</p>
</article>

<article>
<h2>Mozilla Firefox</h2>
<p>Mozilla Firefox is an open-source web browser developed by Mozilla. Firefox
has been the second most popular web browser since January, 2018.</p>
</article>

<article>
<h2>Microsoft Edge</h2>
<p>Microsoft Edge is a web browser developed by Microsoft, released in 2015.
Microsoft Edge replaced Internet Explorer.</p>
</article>
```

Example 2: Use CSS to style the <article> element

```
<html>
<head>
<style>
main {
  margin: 0;
  padding: 5px;
  background-color: lightgray;
}

main > h1, p, .browser {
  margin: 10px;
  padding: 5px;
}

.browser {
  background: white;
}

.browser > h2, p {
  margin: 4px;
  font-size: 90%;
}
</style>
</head>
<body>

<main>
  <h1>Most Popular Browsers</h1>
  <p>Chrome, Firefox, and Edge are the most used browsers today.</p>
  <article class="browser">
    <h2>Google Chrome</h2>
    <p>Google Chrome is a web browser developed by Google, released in 2008.
Chrome is the world's most popular web browser today!</p>
  </article>
  <article class="browser">
    <h2>Mozilla Firefox</h2>
    <p>Mozilla Firefox is an open-source web browser developed by Mozilla.
Firefox has been the second most popular web browser since January, 2018.</p>
  </article>
  <article class="browser">
    <h2>Microsoft Edge</h2>
    <p>Microsoft Edge is a web browser developed by Microsoft, released in 2015.
Microsoft Edge replaced Internet Explorer.</p>
  </article>
</main>

</body>
</html>
```

6. HTML <section> Tag

The **HTML <section> tag** defines a section in a document. It represents a generic standalone section of a document, which doesn't have a more specific semantic element to represent it. Sections should always have a heading, with very few exceptions.

Example: Two sections in a document

```
<section>
<h2>WWF History</h2>
<p>The World Wide Fund for Nature (WWF) is an international organization working on issues regarding the conservation, research and restoration of the environment, formerly named the World Wildlife Fund. WWF was founded in 1961.</p>
</section>

<section>
<h2>WWF's Symbol</h2>
<p>The Panda has become the symbol of WWF. The well-known panda logo of WWF originated from a panda named Chi Chi that was transferred from the Beijing Zoo to the London Zoo in the same year of the establishment of WWF.</p>
</section>

<section>
<h2>WWF History</h2>
<p>The World Wide Fund for Nature (WWF) is an international organization working on issues regarding the conservation, research and restoration of the environment, formerly named the World Wildlife Fund. WWF was founded in 1961.</p>
</section>
```

Nesting <article> in <section> or Vice Versa

The <article> element specifies independent, self-contained content. The <section> element defines section in a document. Can we use the definitions to decide how to nest those elements? No, we cannot! So, you will find HTML pages with <section> elements containing <article> elements, and <article> elements containing <section> elements.

7. HTML <aside> Element

The **HTML <aside> element** defines some content aside from the content it is placed in (like a sidebar). The <aside> content should be indirectly related to the surrounding content. It represents a portion of a document whose content is only indirectly related to the document's main content. Asides are frequently presented as sidebars or call-out boxes.

Example 1: Display some content aside from the content it is placed in

```
<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>
```

Example 2: Use CSS to style the <aside> element

```
<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>
```

8. HTML <figure> and <figcaption> Elements

The **HTML <figure> tag** specifies self-contained content, like illustrations, diagrams, photos, code listings, etc. The <figcaption> tag defines a caption for a <figure> element. The **HTML <figcaption> element** can be placed as the first or as the last child of a <figure> element. The <img> element defines the actual image/illustration.

The <figure> element is intended to be used in conjunction with the <figcaption> element to mark up diagrams, illustrations, photos, and code examples (among other things). The spec says this about <figure>. It represents a unit of content, optionally with a caption, that is self-contained, that is typically referenced as a single unit from the main flow of the document, and that can be moved away from the main flow of the document without affecting the document's meaning.

The <figcaption> element has been the subject of much debate. The spec initially wanted to repurpose <legend> rather than introduce a new element. Other suggestions included <label>, <caption>, <p> or the <h1>–<h6> elements. <legend> was changed, so we then used a combination of <dt> and <dd> inside <figure> at Jeremy's suggestion. Most of these suggestions failed since there was no backwards compatibility for styling with CSS. The <figcaption> element is optional and can appear before *or* after the content within the <figure>. Only one <figcaption> element may be nested within a <figure>, although the <figure> element itself may contain multiple other child elements (e.g., <img> or <code>).

Example

```
<figure>
  <img src="pic_trulli.jpg" alt="Trulli">
  <figcaption>Fig1. - Trulli, Puglia, Italy.</figcaption>
</figure>
```

9. HTML <audio> Tag

The **HTML <audio> tag** is used to embed sound content in a document, such as music or other audio streams. The <audio> tag contains one or more <source> tags with different audio sources. The browser will choose the first source it supports. The text between the <audio> and </audio> tags will only be displayed in browsers that do not support the <audio> element. There are three supported audio formats in HTML: MP3, WAV, and OGG.

Audio Format and Browser Support

Browser	MP3	WAV	OGG
Edge / IE	YES	NO	NO
Chrome	YES	YES	YES
Firefox	YES	YES	YES
Safari	YES	YES	NO
Opera	YES	YES	YES

Example: Play a sound file

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  Your browser does not support the audio tag.
</audio>
```

10. HTML Attributes

HTML attributes provide additional information about HTML elements. Attributes define additional characteristics or properties of the element such as width and height of an image. Attributes are always specified in the start tag (or opening tag) and usually consists of name/value pairs like name="value". Attribute values should always be enclosed in quotation marks. Also, some attributes are required for certain elements. For instance, an <img> tag must contain a src and alt attributes. Let's take a look at some examples of the attributes usages

HTML Attributes

- All HTML elements can have **attributes**
- Attributes provide **additional information** about elements
- Attributes are always specified in **the start tag**
- Attributes usually come in name/value pairs like: **name="value"**

a. The href Attribute

The <a> tag defines a hyperlink. The href attribute specifies the URL of the page the link goes to.

Example

```
<a href="https://www.w3schools.com">Visit W3Schools</a>
```

b. The src Attribute

The <img> tag is used to embed an image in an HTML page. The src attribute specifies the path to the image to be displayed

Example

```
<img src="img_girl.jpg">
```

There are two ways to specify the URL in the src attribute:

1. Absolute URL - Links to an external image that is hosted on another website. Example: src="https://www.w3schools.com/images/img_girl.jpg".External images might be under copyright. If you do not get permission to use it, you may be in violation of copyright laws. In addition, you cannot control external images; it can suddenly be removed or changed.

2. Relative URL - Links to an image that is hosted within the website. Here, the URL does not include the domain name. If the URL begins without a slash, it will be relative to the current page. Example: src="img_girl.jpg". If the URL begins with a slash, it will be relative to the domain. Example: src="/images/img_girl.jpg". It is almost always best to use relative URLs. They will not break if you change domain.

c. The width and height Attributes

The <img> tag should also contain the width and height attributes, which specifies the width and height of the image (in pixels)

Example

```
<img src="img_girl.jpg" width="500" height="600">
```

d. The alt Attribute

The required alt attribute for the <img> tag specifies an alternate text for an image, if the image for some reason cannot be displayed. This can be due to slow connection, or an error in the src attribute, or if the user uses a screen reader.

Example

```
<img src="img_girl.jpg" alt="Girl with a jacket">
```

e. The style Attribute

The style attribute is used to add styles to an element, such as color, font, size, and more.

Example

```
<p style="color:red;">This is a red paragraph.</p>
```

f. The lang Attribute

You should always include the lang attribute inside the <html> tag, to declare the language of the Web page. This is meant to assist search engines and browsers.

Example 1: This example specifies English as the language:

```
<!DOCTYPE html>
<html lang="en">
<body>
...
</body>
</html>
```

Country codes can also be added to the language code in the lang attribute. So, the first two characters define the language of the HTML page, and the last two characters define the country.

Example 2: This example specifies English as the language and United States as the country

```
<!DOCTYPE html>
<html lang="en-US">
<body>
...
</body>
</html>
```

g. The title Attribute

The title attribute defines some extra information about an element. The value of the title attribute will be displayed as a tooltip when you mouse over the element.

Example

```
<p title="I'm a tooltip">This is a paragraph.</p>
```

11. HTML <video> Tag

The **HTML <video> tag** is used to embed video content in a document, such as a movie clip or other video streams. The <video> tag contains one or more <source> tags with different video sources. The browser will choose the first source it supports. The text between the <video> and </video> tags will only be displayed in browsers that do not support the <video> element.

There are three supported video formats in HTML: MP4, WebM, and OGG

Browser	MP4	WebM	Ogg
Edge	YES	YES	YES
Chrome	YES	YES	YES
Firefox	YES	YES	YES
Safari	YES	YES	NO
Opera	YES	YES	YES

Example: Play a video

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>
```

12. HTML <embed> Tag

The **HTML <embed>** tag defines a container for an external resource, such as a web page, a picture, a media player, or a plug-in application. It embeds external content at the specified point in the document. This content is provided by an external application or other source of interactive content such as a browser plug-in.

Example: An embedded image

```
<embed type="image/jpg" src="pic_trulli.jpg" width="300" height="200">
```

Example: An embedded HTML page

```
<embed type="text/html" src="snippet.html" width="500" height="200">
```

Example: An embedded video

```
<embed type="video/webm" src="video.mp4" width="400" height="300">
```

LEARNING COMPETENCY

Use a combination of semantic and non-semantic HTML.

ACTIVITY NO. 1:

Identification: Identify the word/s describe in each of the following sentences. Write your answers in a one whole sheet of paper. (10 points).

1. It is used to embed video content in a document, such as a movie clip or other video streams.
2. It embeds external content at the specified point in the document.
3. It define additional characteristics or properties of the element such as width and height of an image.
4. It represents a portion of a document whose content is only indirectly related to the document's main content.
5. It introduces meaning to the web page rather than just presentation.
6. The _____ is used to embed sound content in a document, such as music or other audio streams.
7. It represents the dominant content of the <body> of a document.
8. The _____ defines a set of navigation links.
9. It represents a container for introductory content or a set of navigational links.
10. The _____ specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.

ACTIVITY NO. 2

Using a notepad or sublime text application, do the examples above about the following semantic and non-sematic HTML:

1. HTML <header> element
2. HTML <nav> element
3. HTML <main> element
4. HTML <footer> element
5. HTML <article> element
6. HTML <section> element
7. HTML <aside> element
8. HTML <figure> and <figcaption> element
9. HTML <audio> element
10. HTML Attributes
11. HTML <video> element
12. HTML <embed> element

Each example that you will do, will be graded based on the rubrics below. Attach the screenshot of your output in our google classroom.

Rubric:

	Excellent (20 points)	**Good (15 points)**	**Fair (10 points)**	**Poor (5 points)**
Coding Validation	There are no errors in the HTML, on the site as found by me or an online validator.	There are 1-3 coding errors on the site as found by me or an online validator.	There are 4-5 coding errors on the site as found by me or an online validator.	There are more than 6 coding errors on the site as found by me or an online validator.

REFERENCES FOR LEARNERS:

- https://www.w3schools.com/html/html5_semantic_elements.asp
- https://www.lifewire.com/why-use-semantic-html-346827
- https://www.freecodecamp.org/news/semantic-html5-elements/
- https://www.geeksforgeeks.org/difference-between-semantic-and-non-semantic-elements
- http://findnerd.com/list/view/Non-Semantic-Elements-and-Semantic-Elements-in-HTML5/10153/
- https://developer.mozilla.org/en-US/docs/Web/HTML/Element/main
- http://html5doctor.com/the-figure-figcaption-elements/
- https://www.tutorialrepublic.com/html-tutorial/html-attributes.php

Prepared by: **ANGELIQUE T. DEL ROSARIO**
Name of Writer

Noted by: **LABERNE A. LADIGNON, JR**
Division ICT Coordinator/ OIC EPS