HTML NOTES

By Sanjay Makwana

What are Semantic Elements?

A semantic element clearly describes its meaning to both the browser and the developer.

Examples of non-semantic elements: `<div>` and `<span>` - Tells nothing about its content.

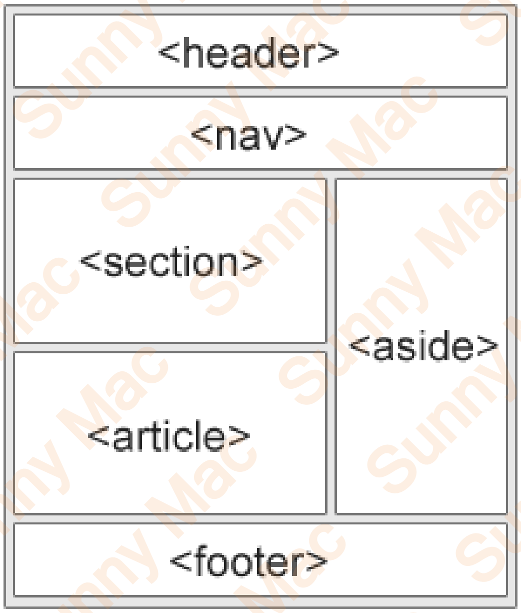
Examples of semantic elements: `<form>`, `<table>`, and `<article>` - Clearly defines its content.

Semantic Elements in HTML

Many web sites contain HTML code like: <div id="nav"> <div class="header"> <div id="footer"> to indicate navigation, header, and footer.

In HTML there are some semantic elements that can be used to define different parts of a web page:

- <article>
- <aside>
- <details>
- <figcaption>
- <figure>
- <footer>
- <header>
- <main>
- <mark>
- <nav>
- <section>
- <summary>
- <time>



HTML <section> Element

The `<section>` element defines a section in a document.

According to W3C's HTML documentation: "A section is a thematic grouping of content, typically with a heading."

Examples of where a `<section>` element can be used:

- Chapters
- Introduction
- News items
- Contact information

A web page could normally be split into sections for introduction, content, and contact information.

```html
<!DOCTYPE html>
<html>
<body>

<section>
  <h1>WWF</h1>
    <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>
  <h1>WWF's Panda symbol</h1>
   <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>

</body>
</html>
```

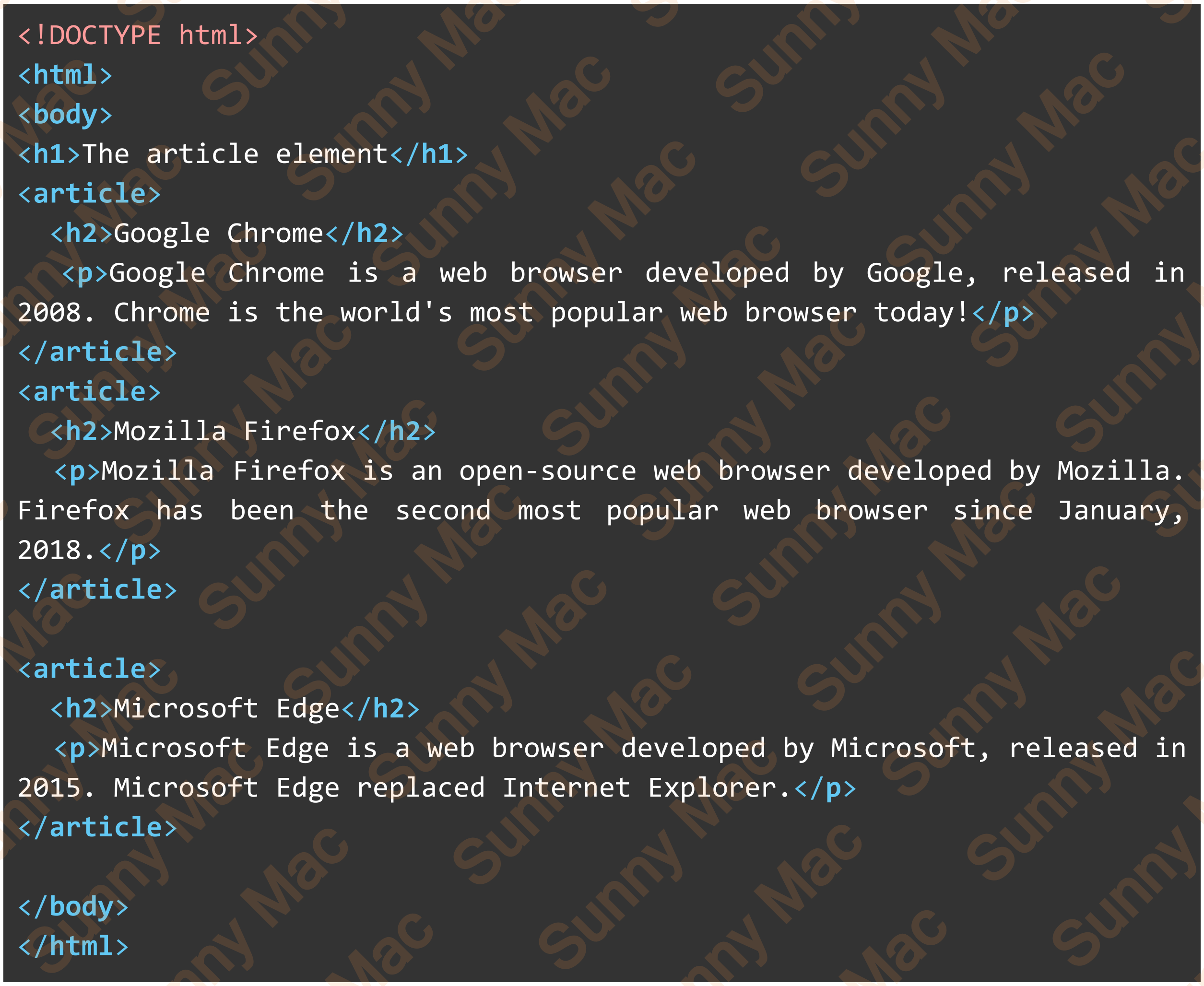
HTML <article> Element

The `<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 the `<article>` element can be used:

- Forum posts
- Blog posts
- User comments
- Product cards
- Newspaper articles



```
<!DOCTYPE html>
<html>
<body>
<h1>The article element</h1>
<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>

</body>
</html>
```

HTML <header> Element

The <header> element represents a container for introductory content or a set of navigational links.

A <header> element typically contains:

- one or more heading elements (<h1> - <h6>)
- logo or icon
- authorship information

Note: You can have several <header> elements in one HTML document. However, <header> cannot be placed within a <footer>, <address> or another <header> element.

```html
<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>
```

HTML <footer> Element

The <footer> element defines a footer for a document or section.

A <footer> element typically contains:

- authorship information
- copyright information
- contact information
- sitemap
- back to top links
- related documents

You can have several <footer> elements in one document.

```
<footer>
  <p>Author: Sanjay Makwana</p>
  <p><a href="mailto:sanjay@example.com">sanjay@example.com</a></p>
</footer>
```

HTML <nav> Element

The <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 blocks of navigation links.

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

HTML <aside> Element

The `<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.

```
<!DOCTYPE html>
<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>
```

HTML <figure> and <figcaption> Elements

The <figure> tag specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.

The <figcaption> tag defines a caption for a <figure> element. The <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.

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

Why Semantic Elements?

According to the W3C: "A semantic Web allows data to be shared and reused across applications, enterprises, and communities."

Summary

Tag	Description
<article>	Defines independent, self-contained content
<aside>	Defines content aside from the page content
<details>	Defines additional details that the user can view or hide
<figcaption>	Defines a caption for a <figure> element
<figure>	Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
<footer>	Defines a footer for a document or section
<header>	Specifies a header for a document or section
<main>	Specifies the main content of a document
<mark>	Defines marked/highlighted text
<nav>	Defines navigation links
<section>	Defines a section in a document
<summary>	Defines a visible heading for a <details> element
<time>	Defines a date/time

HTML Multimedia

Multimedia on the web is sound, music, videos, movies, and animations.Multimedia comes in many different formats. It can be almost anything you can hear or see, like images, music, sound, videos, records, films, animations, and more. Web pages often contain multimedia elements of different types and formats. Multimedia elements (like audio or video) are stored in media files. The most common way to discover the type of a file, is to look at the file extension. Multimedia files have formats and different extensions like: .wav, .mp3, .mp4, .mpg, .wmv, and .avi.

The HTML <video> Element

To show a video in HTML, use the `<video>` element:

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

The `controls` attribute adds video controls, like play, pause, and volume.

It is a good idea to always include `width` and `height` attributes. If height and width are not set, the page might flicker while the video loads.

The `<source>` element allows you to specify alternative video files which the browser may choose from. The browser will use the first recognized format.

The text between the `<video>` and `</video>` tags will only be displayed in browsers that do not support the `<video>` element.

To start a video automatically, use the `autoplay` attribute. Add `muted` after `autoplay` to let your video start playing automatically (but muted).

The HTML <audio> Element

To play an audio file in HTML, use the `<audio>` element:

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

The `controls` attribute adds audio controls, like play, pause, and volume.

The `<source>` element allows you to specify alternative audio files which the browser may choose from. The browser will use the first recognized format.

The text between the `<audio>` and `</audio>` tags will only be displayed in browsers that do not support the `<audio>` element.

To start an audio file automatically, use the `autoplay` attribute. Add `muted` after `autoplay` to let your audio file start playing automatically (but muted).

What is HTML Canvas?

The HTML <canvas> element is used to draw graphics, on the fly, via JavaScript.

The <canvas> element is only a container for graphics. You must use JavaScript to actually draw the graphics.

Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

A canvas is a rectangular area on an HTML page. By default, a canvas has no border and no content.

Always specify an id attribute (to be referred to in a script), and a width and height attribute to define the size of the canvas. To add a border, use the style attribute.

Here is an example of a basic, empty canvas:

```html
<canvas id="myCanvas" width="200" height="100" style="border:1px solid
#000000;">
</canvas>
```

We need Javascript.

```html
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.moveTo(0,0);
ctx.lineTo(200,100);
ctx.stroke();
</script>
```

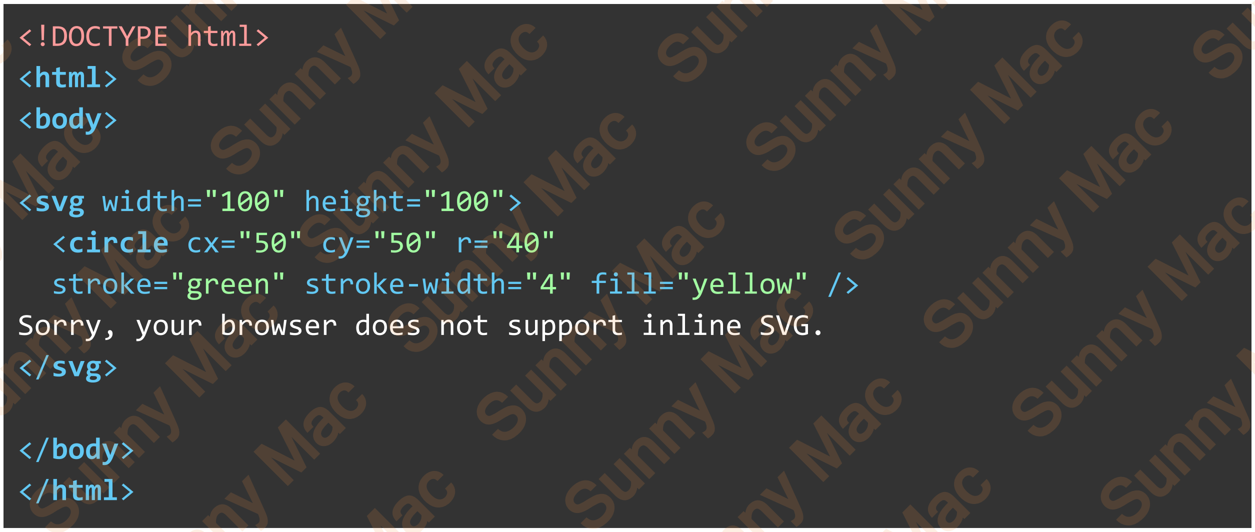
What is SVG?

- SVG stands for Scalable Vector Graphics
- SVG is used to define graphics for the Web
- SVG is a W3C recommendation

The HTML <svg> Element

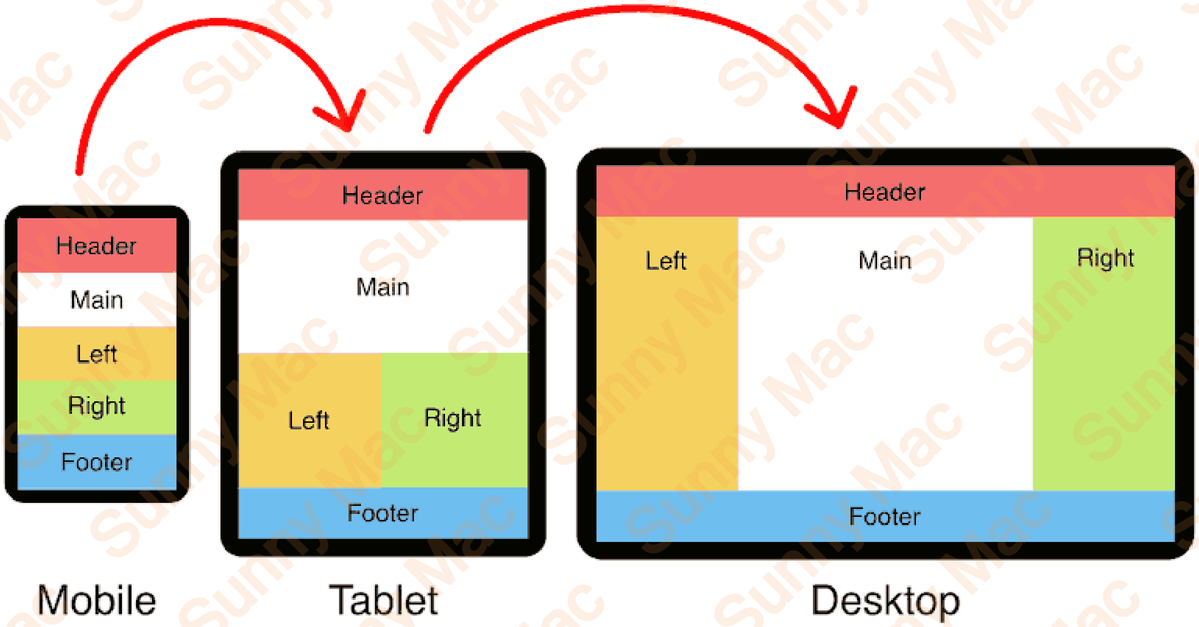
The HTML <svg> element is a container for SVG graphics.

SVG has several methods for drawing paths, boxes, circles, text, and graphic images.



```
<!DOCTYPE html>
<html>
<body>

<svg width="100" height="100">
  <circle cx="50" cy="50" r="40"
  stroke="green" stroke-width="4" fill="yellow" />
Sorry, your browser does not support inline SVG.
</svg>

</body>
</html>
```

Holy Grail Layout 3- Column



The "holy grail" website layout refers to a specific type of web design layout that is often sought after by web designers and developers. It comprises a header section at the top, a footer section at the bottom, and three columns in the middle, with the main content flanked by two sidebars.The sidebars are for navigation menus, advertisements, or other types of content. A responsive holy grail layout collapses into a single column of content areas for mobile devices, and on tablet, the sidebars appear side-by-side below a full-width main content column.

The HTML <form> Elements

The HTML <form> element can contain one or more of the following form elements:

- <input>
- <label>
- <select>
- <textarea>
- <button>
- <fieldset>
- <legend>
- <datalist>
- <output>
- <option>
- <optgroup>

The <input> Element

One of the most used form elements is the <input> element.

The <input> element can be displayed in several ways, depending on the type attribute.

```
<label for="fname">First name:</label>
<input type="text" id="fname" name="fname">
```

HTML Input Types

Here are the different input types you can use in HTML:

`<input type="text">` defines a single-line text input field.

`<input type="password">` defines a password field.

`<input type="submit">` defines a button for submitting form data to a form-handler. The form-handler is typically a server page with a script for processing input data. The form-handler is specified in the form's `action` attribute.

`<input type="reset">` defines a reset button that will reset all form values to their default values.

`<input type="radio">` defines a radio button.Radio buttons let a user select ONLY ONE of a limited number of choices.

`<input type="checkbox">` defines a checkbox. Checkboxes let a user select ZERO or MORE options of a limited number of choices.

`<input type="button">` defines a button.

The `<input type="color">` is used for input fields that should contain a color. Depending on browser support, a color picker can show up in the input field.

The `<input type="date">` is used for input fields that should contain a date. Depending on browser support, a date picker can show up in the input field.

The `<input type="datetime-local">` specifies a date and time input field, with no time zone. Depending on browser support, a date picker can show up in the input field.

The `<input type="email">` is used for input fields that should contain an e-mail address.Depending on browser support, the e-mail address can be automatically validated when submitted.Some smartphones recognize the email type, and add ".com" to the keyboard to match email input.

The `<input type="image">` defines an image as a submit button.The path to the image is specified in the `src` attribute.

The `<input type="file">` defines a file-select field and a "Browse" button for file uploads.

The `<input type="hidden">` defines a hidden input field (not visible to a user).A hidden field lets web developers include data that cannot be seen or modified by users when a form is submitted. A hidden field often stores what database record that needs to be updated when the form is submitted.

The `<input type="month">` allows the user to select a month and year. Depending on browser support, a date picker can show up in the input field.

The `<input type="number">` defines a numeric input field. You can also set restrictions on what numbers are accepted with `min and maxu` attribute.

The `<input type="range">` defines a control for entering a number whose exact value is not important (like a slider control). Default range is 0 to 100. However, you can set restrictions on what numbers are accepted with the `min`, `max`, and `step` attributes

The `<input type="tel">` is used for input fields that should contain a telephone number.

The `<input type="time">` allows the user to select a time (no time zone). Depending on browser support, a time picker can show up in the input field.

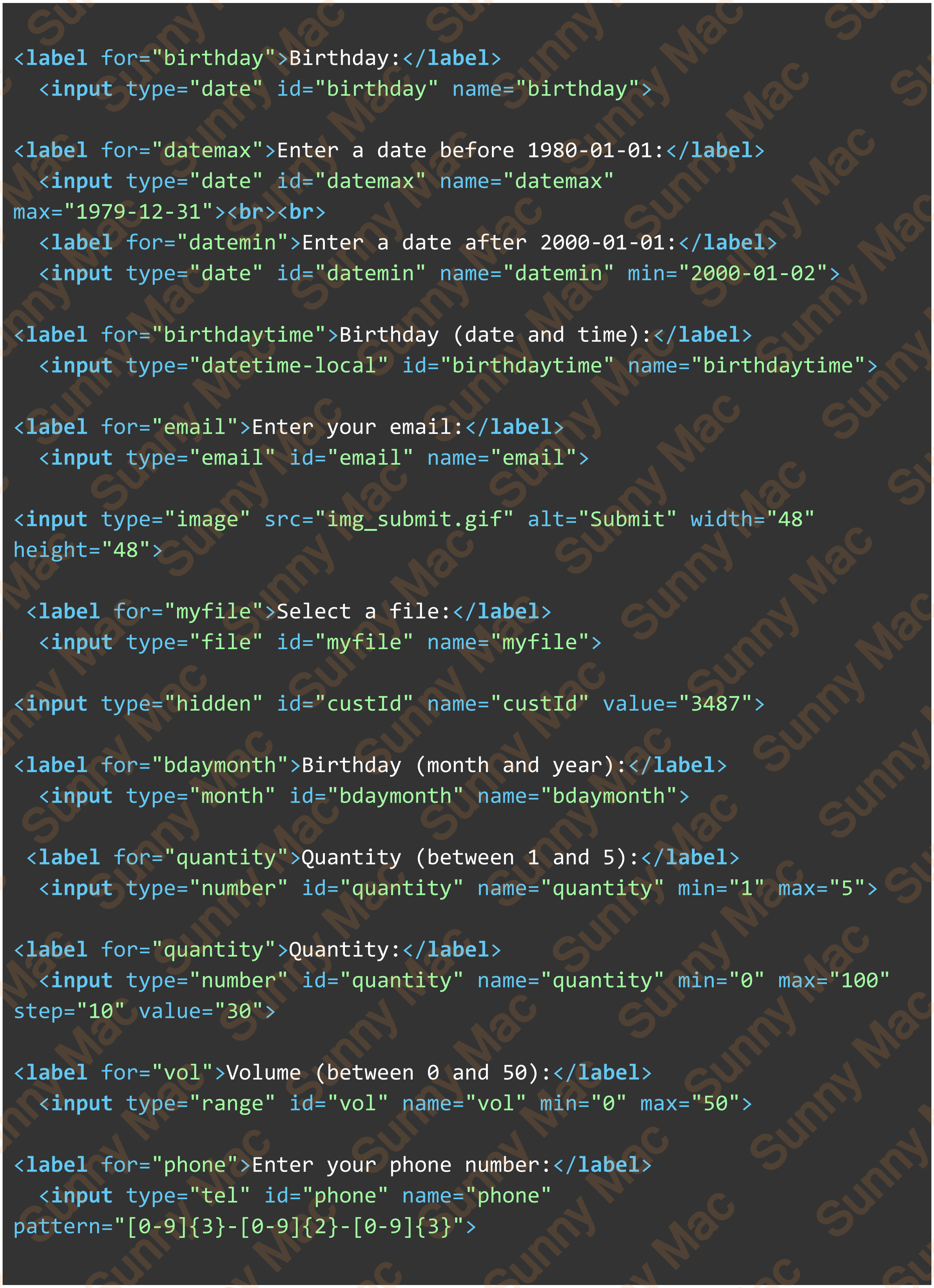
The `<input type="url">` is used for input fields that should contain a URL address. Depending on browser support, the url field can be automatically validated when submitted. Some smartphones recognize the url type, and adds ".com" to the keyboard to match url input.

The `<input type="week">` allows the user to select a week and year. Depending on browser support, a date picker can show up in the input field.

The `placeholder` attribute specifies a short hint that describes the expected value of an input field (e.g. a sample value or a short description of the expected format). The `required` attribute is a boolean attribute.When present, it specifies that an input field must be filled out before submitting the form. The autocomplete attribute specifies if browsers should try to predict the value of an input field or not. You can also specify which type of value you expect in the input field.

Here is Example using all input types :

```html
<!DOCTYPE html>
<html>
<body>

<h2>The input Element</h2>

<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" required  placeholder="Your
name"><br>

<label for="username">Username:</label><br>
  <input type="text" id="username" name="username" autocomplete><br>


input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language"
value="JavaScript">
  <label for="javascript">JavaScript</label>

<input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
  <label for="vehicle1"> I have a bike</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label for="vehicle2"> I have a car</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
  <label for="vehicle3"> I have a boat</label>

<input type="button" onclick="alert('Hello World!')" value="Click Me!">

<label for="favcolor">Select your favorite color:</label>
  <input type="color" id="favcolor" name="favcolor">
```



```html
<label for="birthday">Birthday:</label>
  <input type="date" id="birthday" name="birthday">

<label for="datemax">Enter a date before 1980-01-01:</label>
  <input type="date" id="datemax" name="datemax"
max="1979-12-31"><br><br>
  <label for="datemin">Enter a date after 2000-01-01:</label>
  <input type="date" id="datemin" name="datemin" min="2000-01-02">

<label for="birthdaytime">Birthday (date and time):</label>
  <input type="datetime-local" id="birthdaytime" name="birthdaytime">

<label for="email">Enter your email:</label>
  <input type="email" id="email" name="email">

<input type="image" src="img_submit.gif" alt="Submit" width="48"
height="48">

 <label for="myfile">Select a file:</label>
  <input type="file" id="myfile" name="myfile">

<input type="hidden" id="custId" name="custId" value="3487">

<label for="bdaymonth">Birthday (month and year):</label>
  <input type="month" id="bdaymonth" name="bdaymonth">

 <label for="quantity">Quantity (between 1 and 5):</label>
  <input type="number" id="quantity" name="quantity" min="1" max="5">

<label for="quantity">Quantity:</label>
  <input type="number" id="quantity" name="quantity" min="0" max="100"
step="10" value="30">

<label for="vol">Volume (between 0 and 50):</label>
  <input type="range" id="vol" name="vol" min="0" max="50">

<label for="phone">Enter your phone number:</label>
  <input type="tel" id="phone" name="phone"
pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}">
```

```
<label for="appt">Select a time:</label>
  <input type="time" id="appt" name="appt">

<label for="homepage">Add your homepage:</label>
  <input type="url" id="homepage" name="homepage">

<label for="week">Select a week:</label>
  <input type="week" id="week" name="week">
  <input type="submit" value="Submit">
<input type="reset" value="Reset">
</form>

</body>
</html>
```

The <select> Element

The <select> element defines a drop-down list. The <option> element defines an option that can be selected. By default, the first item in the drop-down list is selected. To define a pre-selected option, add the selected attribute to the option.Use the size attribute to specify the number of visible values. Use the multiple attribute to allow the user to select more than one value.

```
<label for="cars">Choose a car:</label>
<select id="cars" name="cars" size="3">
  <option value="volvo" selected>Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

The <textarea> Element

The <textarea> element defines a multi-line input field (a text area).The rows attribute specifies the visible number of lines in a text area. The cols attribute specifies the visible width of a text area.

```
<textarea name="message" rows="10" cols="30">
The cat was playing in the garden.
</textarea>
```

The <fieldset> and <legend> Elements

The `<fieldset>` element is used to group related data in a form. The `<legend>` element defines a caption for the `<fieldset>` element.

```
<form action="/action_page.php">
  <fieldset>
    <legend>Personalia:</legend>
    <label for="fname">First name:</label><br>
    <input type="text" id="fname" name="fname" value="John"><br>
    <label for="lname">Last name:</label><br>
    <input type="text" id="lname" name="lname" value="Doe"><br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```