HTML Fundamentals

What is HTML?

HTML Layout:

HTML layouts provide a way to arrange web pages in well-mannered, well-structured, and in responsive form or we can say that HTML layout specifies a way in which the web pages can be arranged. Web-page layout works with arrangement of visual elements of an HTML document.

What are Meta Tags in HTML?

`meta` tags live within the `head` tag of the HTML document.

The `head` tag is used for configurating the HTML file.

You use the `head` tag to add a title to the webpage, link to a CSS stylesheet, and define more information about the HTML document.

`meta` tags represent metadata. They are essentially used for defining and describing data about data, and are used to add extra information to the data inside the webpage.

There are many `meta` tags. Some of them help improve the SEO (Search Engine Optimisation) of your website, making sure that the content of your site is relevant to what people are searching for.

How to define the character set of a website

`<meta charset="UTF-8">` defines the character set that will be used in the site.

`UTF-8`, which stands for 8-bit Unicode Transformation Format, is the standard character encoding used with the latest version of HTML, which is HTML5.

This line should be included in every single webpage created, as it ensures that every character from every language in the world is displayed properly in every browser.

By using the universal `UTF-8` as the character set, characters from non-latin languages will not be distorted.

The Google Chrome browser has automatically set the encoding to `UTF-8`, so you won't have to worry about that when designing for this browser. But you still need to include `<meta charset="UTF-8">` in every HTML file in case this feature is not supported by other browsers.

For example, look at what happens in the Safari browser when this line is not added and I write a heading in a non-latin language, such as Greek:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h1>Γεία σου κόσμε!</h1> <!-- Hello world! -->
</body>
</html>
```

How to let Microsoft's Internet Explorer know which rendering view to use

You use the `http-equiv="X-UA-Compatible" content="IE=edge"` meta tag to choose and define the version of Internet Explorer in which the web page will be renedered.

Always choose the latest one, which is `IE=edge`.

There are many versions of Microsoft's browser. In the past the different advances caused headaches to web designers and web developers alike, who worked on making sure websites were usable on legacy browsers.

This tag will ensure that the website will not be rendered as an older version of Internet Explorer, which tend to be buggy.

How to adjust viewport settings

Nowadays, it is important that all sites look good on all devices, especially mobile phones.

So, you need to include the `meta name="viewport" content="width=device-width, initial-scale=1.0"` tag in every HTML file.

`viewport` refers to how the site is displayed on different screen sizes, and how much visual area a user has available.

Each device has a different viewport. For example, mobile devices have a smaller one and desktop computers have a larger one.

`content="width=device-width` is the first step to making sure that websites look good on mobile devices.

It prevents a site that is viewed from a mobile device from looking like it would on a laptop – that is small and far away zoomed out.

This ensures that the HTML will adjust to the width of the device's screen.

`initial-scale=1.0` sets how the webpage scales,and sets the initial zoom when the page is first loaded by the browser.

Additional Meta tags:

How to add a description of your webpage

Using a meta description tag for your page helps search engines figure out and rank your website against other websites. It's used primaraly for SEO (Search Engine Optimization) purposes.

```
<meta name="description" content="Our mission: to help people learn to code for free. We accomplish this by creating thousands of videos, articles, and interactive coding lessons - all freely available to the public.">
```

How to add the name of the website's author

Another useful `meta` element to include is the author's name.

```
<meta name="author" content="Quincy Larson">
```

How to add Title of website

```
<meta name="og:title" content="My Website">
```

How to add Image of website

```
<meta name="og:image" content="https://www.example.com/image.jpg">
```

How to add Keyword in website

```
<meta name="keywords" content="web development, html, CSS, javascript">
```

Async/Defer in HTML

async:

The `async` attribute means that a script is completely independent:

- The browser doesn’t block on `async` scripts (like `defer`).
- Other scripts don’t wait for `async` scripts, and `async` scripts don’t wait for them.
- `DOMContentLoaded` and async scripts don’t wait for each other:
  - `DOMContentLoaded` may happen both before an async script (if an async script finishes loading after the page is complete)
  - …or after an async script (if an async script is short or was in HTTP-cache)

In other words, `async` scripts load in the background and run when ready. The DOM and other scripts don’t wait for them, and they don’t wait for anything. A fully independent script that runs when loaded.

```
<p>...content before scripts...</p>

<script>
  document.addEventListener('DOMContentLoaded', () => alert("DOM ready!"));
</script>

<script async src="https://javascript.info/article/script-async-defer/long.js"></script>
<script async src="https://javascript.info/article/script-async-defer/small.js"></script>

<p>...content after scripts...</p>
```

Async scripts are great when we integrate an independent third-party script into the page: counters, ads and so on, as they don't depend on our scripts, and our scripts shouldn't wait for them:

```
<!-- Google Analytics is usually added like this -->
<script async src="https://google-analytics.com/analytics.js"></script>
```

defer:

The `defer` attribute tells the browser not to wait for the script. Instead, the browser will continue to process the HTML, build DOM. The script loads "in the background", and then runs when the DOM is fully built.

```
<p>...content before script...</p>

<script defer src="https://javascript.info/article/script-async-defer/long.js?speed=1"></s
cript>

<!-- visible immediately -->
<p>...content after script...</p>
```

In other words:

- Scripts with `defer` never block the page.

- Scripts with `defer` always execute when the DOM is ready (but before `DOMContentLoaded` event).

🗣️ Please Check the following sub-pages:

Semantic Elements

Anchor Tags

Lists

Div/Span