Web Designer code of conduct

1. We write our own code
2. We write little code as possible
3. We design for speed and efficiency
4. We separate content from presentation
5. We rigorously indent our code
6. We know that simple solutions work best
7. We verify our solutions
8. We learn, experiment and play constantly

File structure

1. Img folder
2. Css folder
3. Js folder

Web development tool

1. Complexity
2. Security
3. Scalability
4. Cost
5. Programming language
6. Platform support

Ide – integrated development environment

Why ide important

1. Code editing automation
2. Syntax Highlighting
3. Intelligent code completion
4. Refactoring support
5. Local build automation
6. Compilation
7. Testing
8. Debugging

Local ide –

1. Sublime text
2. Visual studio code
3. Notepad++
4. PhpStorm
5. Eclipse

**What are the types of IDEs?**

**Cloud ides**

1. Standardized development environment
2. Platform independece
3. Better performance

**Aws cloud9 -** cloud-based integrated development environment (IDE) that lets you write, run, and debug your code with just a **browser.**

**Wireframe** – skeletal structure of a website or visual guide to present a page schematic, blueprint etc.

Marvel – quickly create and easy design different projects.

Wireframesketcher – quickly create mock ups, wireframes and prototype for desktop.

**API – Application programming interface**

Postman – help web developer to build, test, share and modify api

Docker – deploying applications inside a virtual containers

**Web Development Framework**

Flutter – for building cross platform applications

Reactjs – creating modern and responsive web and mobile app user interface

Vue.js – help programmers to develop web app and mobile app with ease.

Angular – create a wide range of apps

Laravel – single page websites into enterprise level applications.

**Git** - free and open source distributed version control system that’s handle everything from small to very large projects.

**Linus Torvalds** in **2005**, for development of the Linux kernel

**Junio Hamano** has been the core maintainer.

**BENEFITS OF GIT**

* **It’s free (and open-source):**
* **Performance:**
* **Security:**
* **Widely-used**

**SHA1 (Secure Hash Algorithm 1)**

**GIT REPOSITORIES/ Repo** - contains all of the project files and the entire revision history.

**Branch** - is essentially is a unique set of code changes with a unique name.

The **main branch** — the one where all changes eventually get merged back into, and is called **master.**

**PULL REQUESTS** - are a way to discuss changes before merging them into your codebase

**GIT COMMANDS** - are a distributed version control system for tracking changes in any set of files. They were originally designed for coordinating work among programmers who were operating source codes during software development.

**Git clone** - used for downloading the latest version of a remote project and copying it to the selected location on the local machine

**Git fetch** - command downloads commits, files, and refs from a remote repository into your local repo, will get all the updates from the remote repository, including new branches

**Git checkout** - to switch the branch that you are currently working on

**Git init** - you need to use if you want to start a new empty repository or to reinitialize an existing one in the project root

**Git add** - adds a change in the working directory to the staging area. a. It tells Git that you want to include updates to a particular file in the next commit.

**Git commit** - command captures a snapshot of the project's currently staged changes.

**Git commit** and **git add** are two of the most frequently used.

**Git push** - is most commonly used to publish and upload local changes to a central repository

**Git diff –** isto see the unstaged changes on the current branch. You may also compare two branches

**Git pull** - is used to fetch and download content from a remote repository and immediately update the local repository to match that content.

**Git merge** - lets you take the independent lines of development created by git branch and integrate them into a single branch. Also it is the way of putting forked history back together again.

**SEO** – Search Engine Optimization

Is the process of improving your site to increase its visibility when people search for products or services related to your business in Google, bing and other search engine.

**Black Hat SEO** – is the practices that violate the search engines' terms of service.

**Black hat SEO techniques or practices** include:

* **Keyword Stuffing**
* **Link Manipulation**
* **Creating articles, pages, or site landing pages with duplicate content**

**Word of caution**: experience short-term success. The traffic to your site might increase rapidly.

**White Hat SEO -** is a SEO tactics that are in tandem and agreement with the terms and conditions of the search engines

**White hat SEO practice like:**

* **Creating original quality content and services**
* **Mobile-friendly website**
* **Use of clear and keyword-rich meta tags**

**Paid Search -** presented with a list of results; or SERP (the search engine results page) which shows organic results and paid results. Results have a little green box with the word **“ad”** before the listing.

**Organic Search** - are the unpaid results that appear on a search engine results page after a query.

**Factor:**

* **Backlinks**
* **Domain authority**
* **Relevance.**

**Best Practices**

**CREATE UNIQUE, ACCURATE PAGE TITLES**

* Indicate page titles by using title tags
* Page title contents are displayed in search results
* ACCURATELY DESCRIBE THE PAGE'S CONTENT
* USE UNIQUE DESCRIPTIONS FOR EACH PAGE
* USE BRIEF, BUT DESCRIPTIVE TITLES
* ACCURATELY SUMMARIZE THE PAGE CONTENT

**Description Meta** **Tags** - Google might use them as snippets for your pages

**IMPROVE THE STRUCTURE OF YOUR URLS**

* Simple-to-understand URLs will convey content information easily
* URLs are displayed in search results
* USE WORDS IN URLS
* CREATE A SIMPLE DIRECTORY STRUCTURE
* PROVIDE ONE VERSION OF A URL TO REACH A DOCUMENT

**MAKE YOUR SITE EASIER TO NAVIGATE**

**Navigation -** helping visitors quickly find the content they want

* Plan out your navigation based on your homepage
* Ensure more convenience for users by using ‘breadcrumb lists’
* Allow for the possibility of a part of the URL being removed’
* Prepare two sitemaps: one for users, one for search engines

**2 TYPES OF SITEMAPS**

**HTML SITEMAPS** - Designed for humans

**USE BRIEF, BUT DESCRIPTIVE TITLES** - Designed for crawlers

**HTML (Hypertext Markup Language**) - is the only markup language for creating web pages. It provides some titles, headings, paragraphs, lists, tables, embedded images, etc.

- Publishing **text-based and multimedia information** on the **World Wide Web**.

- **Straightforward Computer Coding Language**. It was developed in the 90s.

HTML uses '**tags'** to create web documents.

**The first language of web designing**

HTML was created by **Sir Tim Berners-Lee** in late **1991**

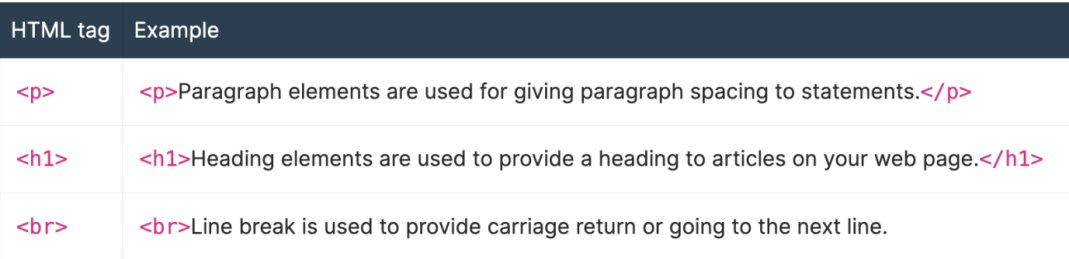
**HTML 1.0** was released in **1993** with the intention of sharing information that can be readable and accessible via web browsers.

**HTML 2.0, published in 1995**, which contains all the features of HTML 1.0 along with a few additional features, which remained the standard markup language for designing and creating websites until **January 1997**

**HTML 4.01 -** widely used and was a successful version of HTML.

**HTML 5** can be said for an extended version of HTML 4.01, which was published in the year **2012.**

**HTML Doctype Declaration** refers to a **Document Type Definition (DTD).**

* XML document format representing allowed elements in a web page
* It is a directive that tells the web browser about the HTML version and standard in which the current page is written; this helps different web browsers parse the web page correctly.1

**DOCTYPE DECLARATION**

* Should be on the first line at the top of all other content on the web page.
* **Concise, easy to use, reliable, and case-insensitive.**
* In **HTML5,** only the doctype declaration is required to enable standards mode for web pages.

**HTML TAGS** - hidden keywords or commands incorporated in HTML. It define how your browser will display the content and format of the web page.

**Two sections: opening and a closing portion**

* All HTML documents begin with **<! DOCTYPE html>**
* HTML documents begins with **<html> and </html>**
* HTML code that **is written inside the <head> and </head> tags** of an HTML document are not visible in the web browser.
* Only the content of **<body> and </body> tags will be displayed in browser.**

**HTML Tag**

The **<head>** tag is another important tag used to add the header in HTML. It is used to gives various additional information about the web page along with description and title to your page. The <title> tag is written within this <head> tag.

**HTML Body Tag**

HTML <body> tag is used to give the body, i.e., the visible section of the HTML document. All formatting and writing of content are done in this section within the opening <body> and the closing

</body> tag.

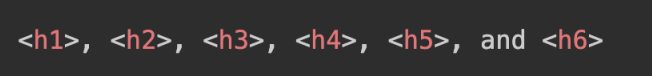
**HTML element** is the collection of start tag

**HTML tag** (either opening or closing) is used to mark the start or end of an element

**NESTED ELEMENTS** - you can write an element, and before closing that element, you can start and finish another element within that outer element

**HEADING ELEMENTS OR ELEMENTS**

Html allows six sizes for heading that uses elements name in format of <hn> where n starts from 1-6.



**HTML ATTRIBUTES**

Attribute is a **property name used to provide supplementary information about HTML elements**

Some common examples of HTML attributes are **id, class, align, etc.**

**HTML ATTRIBUTES CONSIST OF TWO PARTS:**

* **Names of attributes** and their values are **not case-sensitive.** But according to the World Wide Web Consortium (W3C), it is recommended to use lowercase names and values.
* **The value** defines the value which you want to assign to the property and is set within quotations.

**FOUR ESSENTIAL ATTRIBUTES THAT YOU CAN APPLY TO ALMOST ALL HTML ELEMENTS:**

**ID Attribute** - is to assign a unique identity to any element. It also provides a unique identifier, which ultimately makes it impossible to identify an element.

**Title Attribute** – gives a recommended title for your element.

**Class Attribute** – by combining an element through a stylesheet (css) and identifying its class element.

**CSS – Cascading Style Sheet**

**Style Attribute** – gives a chance for specifying the rules for CSS in your element.

**HTML FORMATTING**

**Formatting** is one of the crucial sections of every web development. It plays an essential role in making content attractive, readable, and appealing.

* Can be defined as the appearance of your documentation or presentation of your HTML code in a meaningful and more beautiful way.
* Mainly done to make the layout attractive.

**HTML COMMENTS**

* Essential part of HTML; it helps to provide the details of what is written in the HTML source code.

**HTML HEAD TAG**

* Is a container for metadata (information about the document, such as its title, scripts, and style sheets) that is not directly related to the page's content

HTML LISTING TAG

ul – unordered list

ol – ordered list

<dl> - definition list

* <dd> - definition description
* <dt> - definition terminology

Start attribute – providing starting point or value of your list.

div – division tag

href – hypertext reference

hspace – horizontal spacing all around the image

vspace – vertical spacing all around the image

**HTML Button Tag Attributes**

TYPE

NAME

VALUE

DISABLED

ONCLICK

FORM

FORMACTION

FORMENCTYPE

FORMNOVALIDATE

FORMMETHOD

FORMTARGET

**Html common attributes**

CLASS

ID

ALT

SRC

HREF

WIDTH AND HEIGHT

**Html video tag attributes**

SRC

AUTOPLAY

MUTED

CONTROL

WIDTH AND WEIGHT

**Html body tag attributes**

BGCOLOR

TEXT

Onload

Unonload

Link

Vlink

Alink

Class

Id

Rowspan

Colspan

HTML PHRASE LIST

* Emphasis tag
* Strong tag
* Marked tag
* Abbreviation tag
* Quotation tag
* Short quote tag
* Special termination / definition tag
* Code tag
* Keyboard tag
* Address tag

HTML IMG TAG

* Img
* Src
* Alt
* Border
* Width and weigth
* Vspace and hspace

HTML TABLE TAG

* BORDER
* HEADING
* ROWSPAN AND COLSPAN
* <td> - merge two or more columns into a single column
* Cell padding & cell spacing
* Table background

**Html common EVENT attributes**

* Onload
* Onclick
* Onmouseover
* Onfocus
* Onsubmit

**HTML WEB F ORM**

* CONTACT FORM
* REGISTRATION FORM
* SIGN IN FORM
* LEAD GENERATION FORM
* ORDER FORM
* SURVEY FORM
* SEARCH FORM
* EMAIL FORM

**HTML FORM ELEMENT**

**2 IMPORTANT ATTRIBUTES**

* ACTION
* METHOD

**HTTP METHOD**

* **Get**
* **Post**
* **Put**
* **Patch**
* **Delete**

**HTML FORM INPUT TYPES**

* TEXT
* PASSWORD
* EMAIL
* RADIO
* CHECKBOX
* FILE
* IMAGE
* HIDDEN
* SUBMIT
* RESET
* BUTTON

HTML5 FORM INPUT TYPES

* NUMBER
* TEL
* RANGE
* URL
* SEARCH
* COLOR
* DATE
* TIME
* DATETIME – LOCAL
* MONTH
* WEEK

LABEL – “FOR” ATTRIBUTE