

HTML | CSS Basic Structure



- **HTML Structure [Source Code]**
- **CSS Structure [Cascading Styles]**
- **DIV or ID Tags and Classes**
- **The BOX MODEL**
- **INLINE Elements and FLOATS**

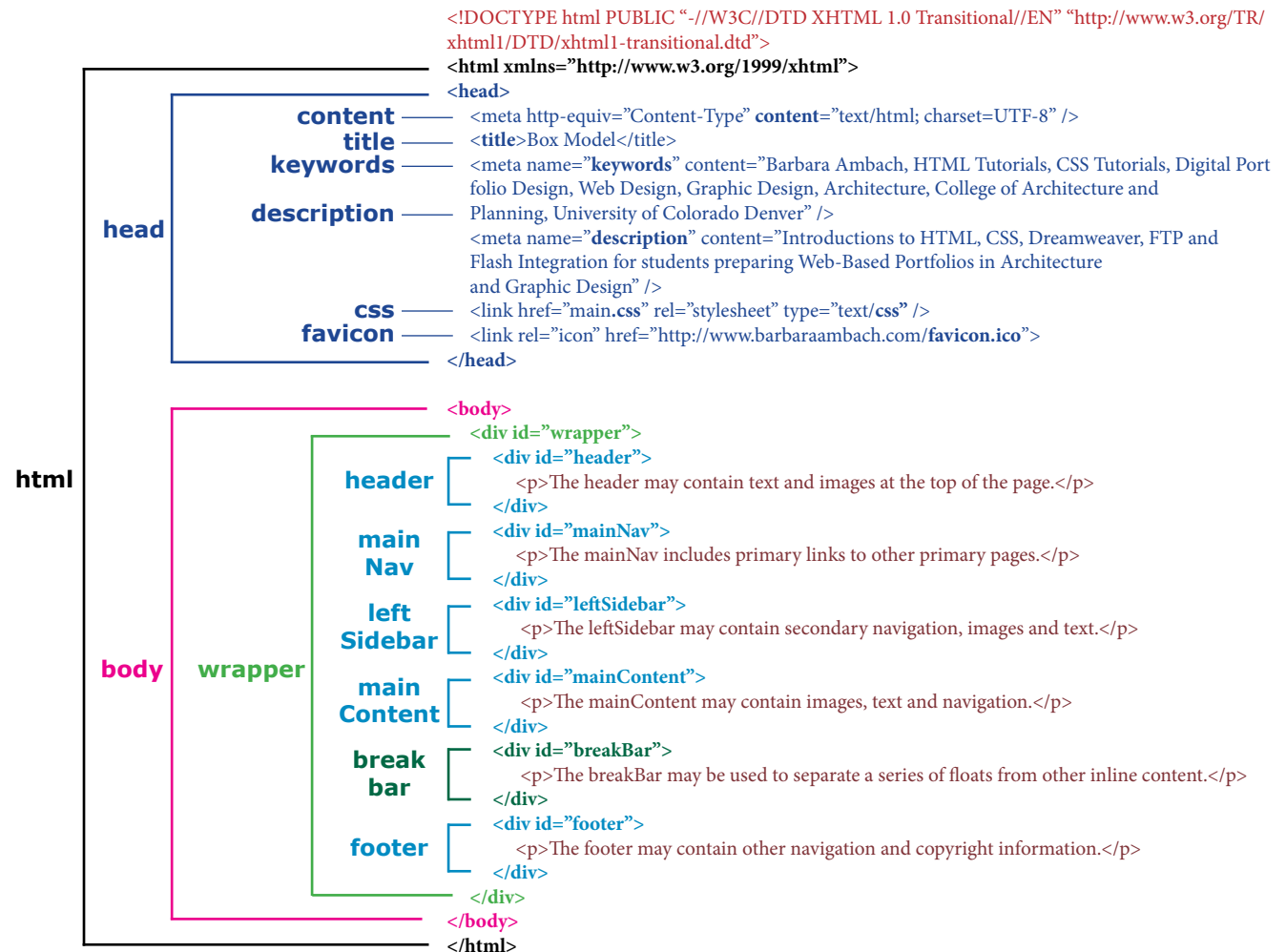
- **HTML Structure [Source Code]**

- **CSS Structure [Cascading Styles]**
- **DIV or ID Tags and Classes**
- **The BOX MODEL**
- **INLINE Elements and FLOATS**

HTML [Hypertext Markup Language] is the code read by a browser and defines the overall page structure. The HTML file or web page [.html] is made up of a **"head"** and a **"body"**. The **"head"** of the document contains the page title, meta tags, links to CSS, Javascript and other items referenced within the page. The **"body"** may be seen as the extents of the browser window itself and contains all of the elements that make up the contents of the page.



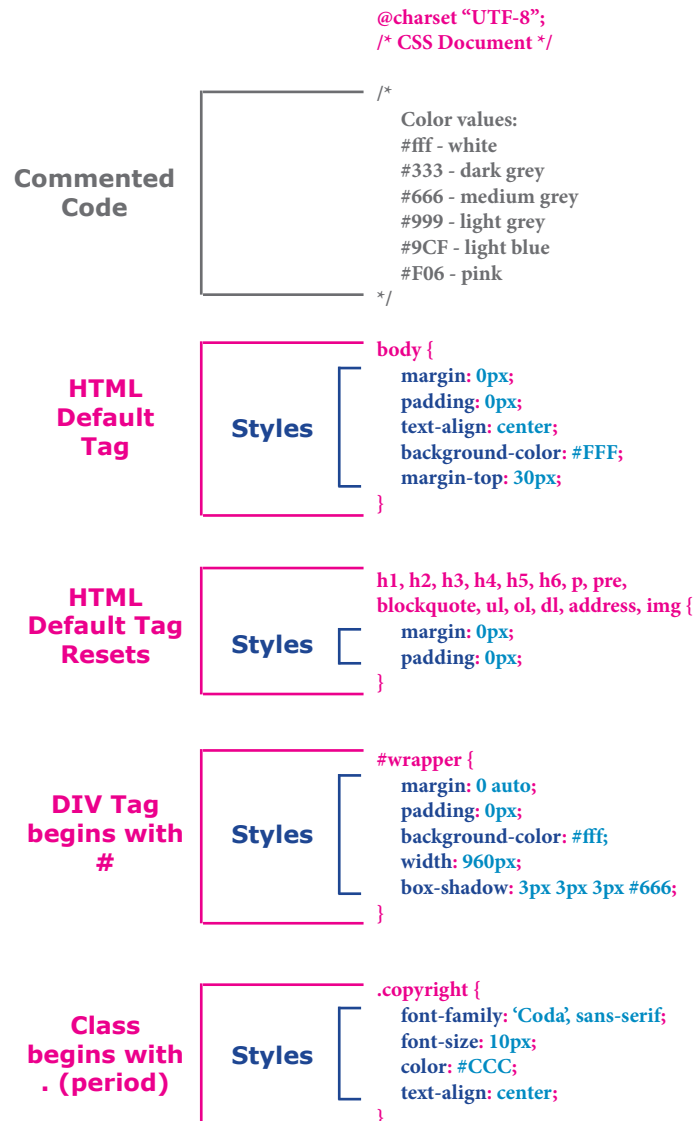
The HTML Structure is a series of nested elements similar to a **"Box of Chocolates"**. The box is the "wrapper" for all of the chocolates. Each chocolate has an individual container or "div". Each "div" contains a series of elements and/or "classes", such as; milk or dark chocolate, filling, nuts and fruit.



- HTML Structure [Source Code]
- DIV or ID Tags and Classes
- The BOX MODEL
- INLINE Elements and FLOATS

● CSS Structure [Cascading Style Sheet]

CSS [Cascading Style Sheet] defines the look of the content within the HTML DIV Tags and Classes.
The CSS is "linked" to the HTML file [Format - Attach Style Sheet - browse for .css file]



Commented code is grey and begins with a `/*` and ends with a `*/`

A generic "comment" may be added or a rule may be temporarily "commented-out" so that it is not read or rendered by the browser.

CSS Resets remove the HTML browser defaults for margins, padding and many other attributes that you may wish to have more control over by overwriting them with your own CSS Rules. These Resets avoid conflicts between the browser's default values and the CSS Rules you write.

- **HTML Structure [Source Code]**
- **CSS Structure [Cascading Styles]**
- **The BOX MODEL**
- **INLINE Elements and FLOATS**

● **DIV or ID Tags and Classes**

Styles for DIV Tags and Classes are written in the CSS [Cascading Style Sheet].

DIV or ID Tags:

A DIV or ID Tag will have a CSS Rule beginning with a # ...
i.e.: #wrapper, #header, #sideBar, #mainContent, #footer.

Each DIV or ID may be used only once in the document and is used for primary structural elements [or DIVISIONS within the page].

Each DIV must be added to the page structure by INSERTING a DIV Tag in the appropriate place in the Source Code:

At Insertion Point = wherever the cursor is placed within the source code ...

Wrap Around Selection = wraps the div around previously selected content in the source code ...

After Start of Tag = after the start tag of a previously placed div ... `<div> ✓ </div>`

After Tag = after the closing tag of a previously placed div ... `<div> </div> ✓`

Classes:

A **Class** will have a CSS Rule beginning with a . (period) ...
i.e.: .title, .img, .block, thumb, etc. etc.

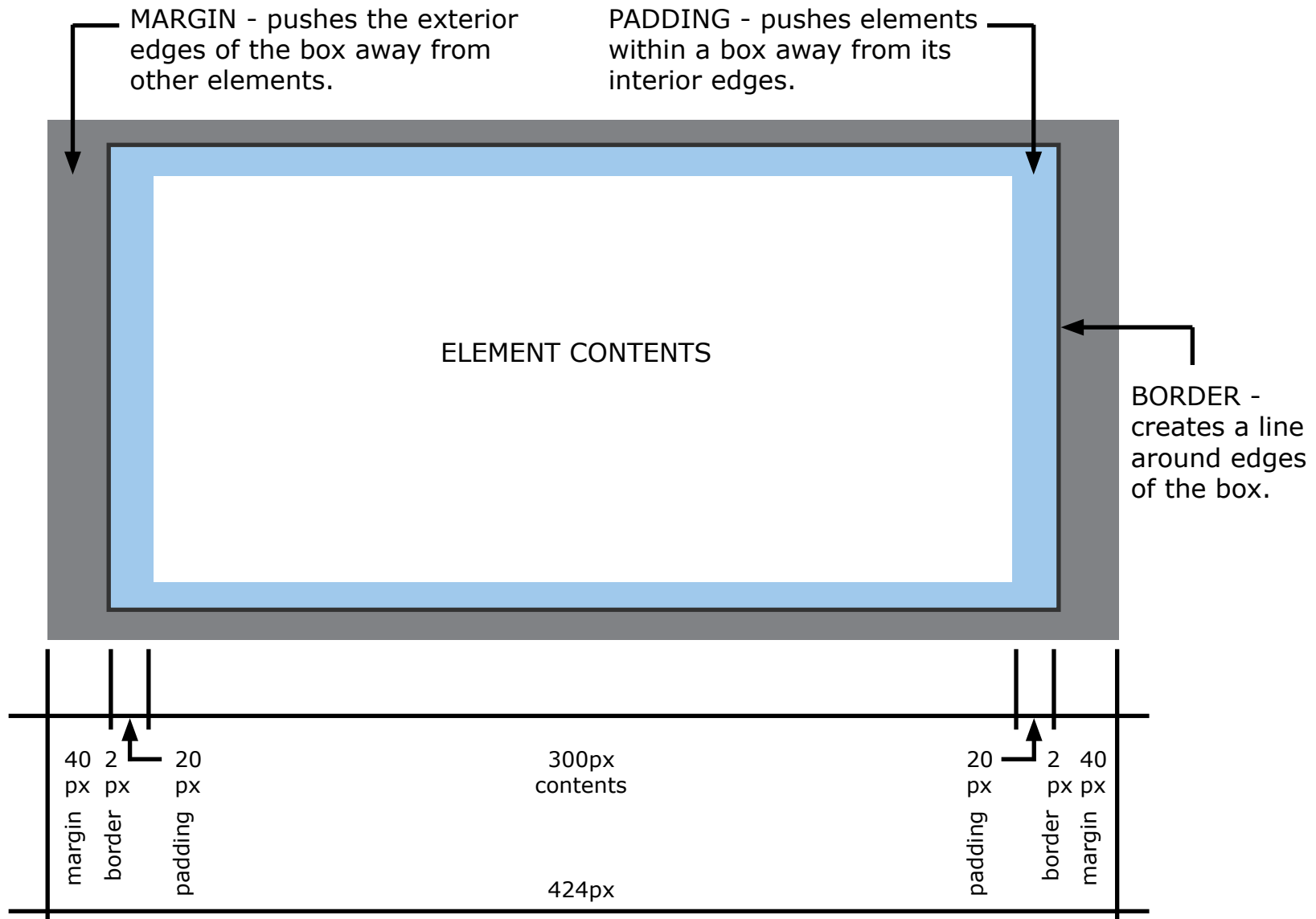
Each Class may be applied to multiple elements in the document to add more specific style attributes.

They may be used to define the specific "look" of various types of text, colors, images, etc. within a div.

• The BOX MODEL

- HTML Structure [Source Code]
- CSS Structure [Cascading Styles]
- DIV or ID Tags and Classes
- INLINE Elements and FLOATS

An element [div or class] may be given margins, padding and borders determining the amount of area it takes up on the page. All of these add to the overall pixel dimensions that the element will occupy.



- **HTML Structure [Source Code]**
- **CSS Structure [Cascading Styles]**
- **DIV or ID Tags and Classes**
- **The BOX MODEL**

● **INLINE Elements and FLOATS**

Elements may be added to the page with **INLINE** structure or with **FLOATS**.

INLINE Elements are placed within “normal” document flow.

This page structure maintains the vertical organization of elements as they are added to the page one after the other.

FLOATED Elements are removed from “normal” document flow.

This page structure allows elements to “float” above the page and, as a result, other elements move up to take the place of where they used to be. Floated elements may be placed next to one another horizontally.

Elements may be “floated” left or right.

Elements Floated “left” align to the furthest left position possible.

Elements Floated “right” align to the furthest right position possible.

Clearing Floats:

A CSS Rule of `clear: left;` or `clear: right;` keeps the next element from floating on that side of a previously floated element.

A CSS Rule of `clear: both;` returns the element back to “normal” document flow and pushes the next Element below the previously “floated” Elements.

A Class of `.clear` with a CSS Rule of `clear: both;` may be added to an element to force no other elements to float next to it or below it.