Basics of HTML-CSS

- Santhosh [220895]

Agenda

DocType

- HTML Structure

- Need for DTD

Box Model

- Box Model

- Block Vs Inline Elements

- Margin Vs Padding

Positioning

- CSS Float

- CSS Position

- zIndex

CSS Specificity

- CSS Selectors

- !important

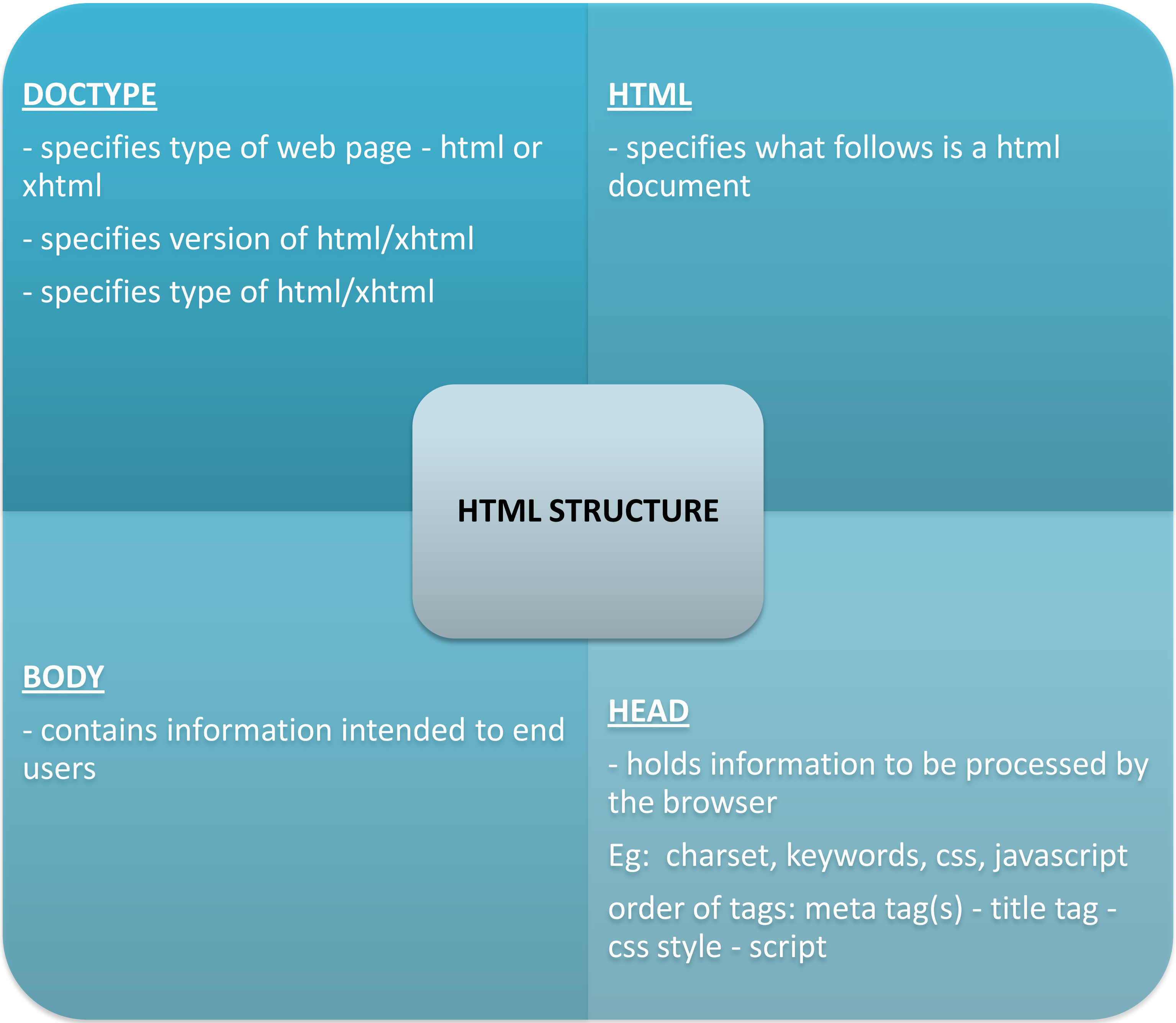
IE Bugs & CSS

Best Practice

DocType

HTML Page Structure

```
<DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>

<html>

        <head>

                <meta info>

                <title>

                        <!-- page title -->

                </title>

                <css styles>

                <scripts>

        </head>

<body>

        <!-- Content -->

</body>

</html>
```



HTML STRUCTURE

DOCTYPE

- specifies type of web page - html or xhtml

- specifies version of html/xhtml

- specifies type of html/xhtml

HTML

- specifies what follows is a html document

BODY

- contains information intended to end users

HEAD

- holds information to be processed by the browser

Eg: charset, keywords, css, javascript

order of tags: meta tag(s) - title tag - css style - script

DTD - Document Type Definition

- Contains Definition for HTML Elements

E.g. [Definition for Paragraph Element]

```
<!--=== Paragraphs ===-->

<!ELEMENT P - O (%inline;)*  -- paragraph -->

<!ATTLIST P

%attrs;  -- %coreattrs, %i18n, %events–

%align;  -- align, text alignment -- >

[- MANDATORY] [O Optional] [(%inline)* Content Model]
```

Types

1. STRICT

   - contains semantic tags.

   E.g. a, addr, h1-h6

2. LOOSE/TRANSITIONAL

   - mostly commonly used DTD

   - contains presentational and semantic tags.

   E.g. b, i, center

3. FRAMESET

   - contains presentational, semantic and frameset tags.

   E.g. frameset, frame

Need for DTD

1. Web page Validation http://validator.w3.org/

2. Consistent Look and Feel across Browsers.

Browser Modes [DOCTYPE Switch]

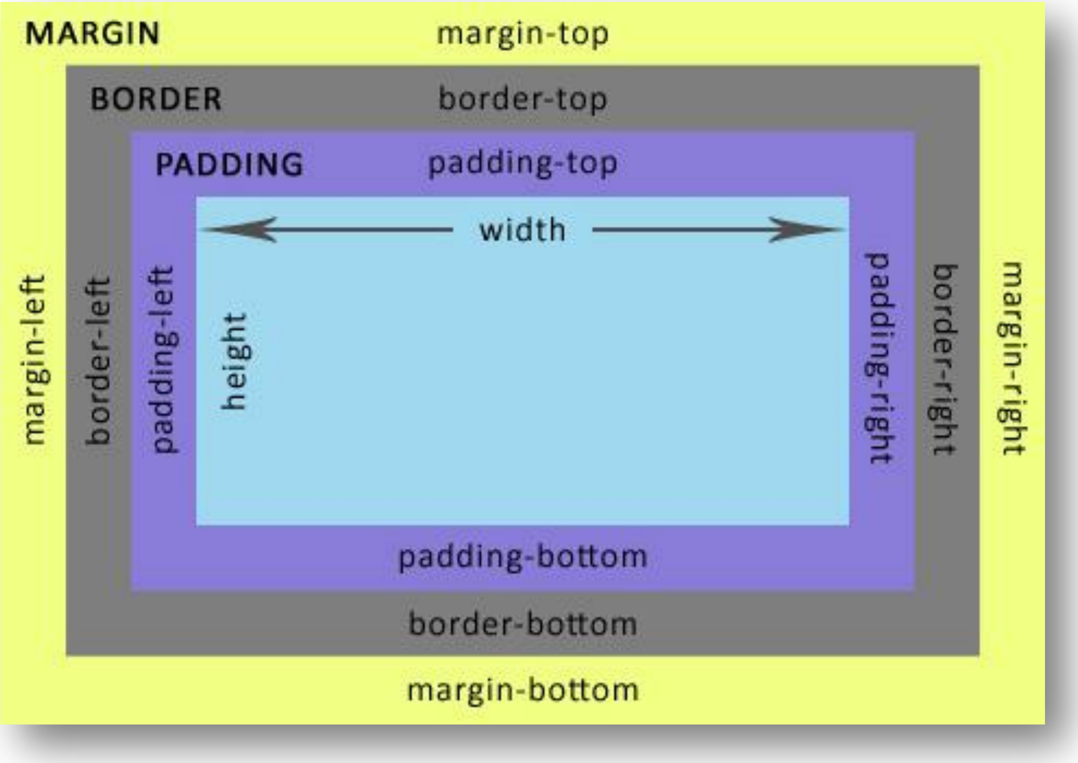
1. Standards-Compliant Mode

- rules for rendering elements are based on W3C Spec.

2. Quirks Mode

Box Model

W3C Box Model



HTML Element's Width =

Margin-Left + Border-Left + Padding-Left

+ Width +

Padding-Right + Border-Right + Margin-Right;

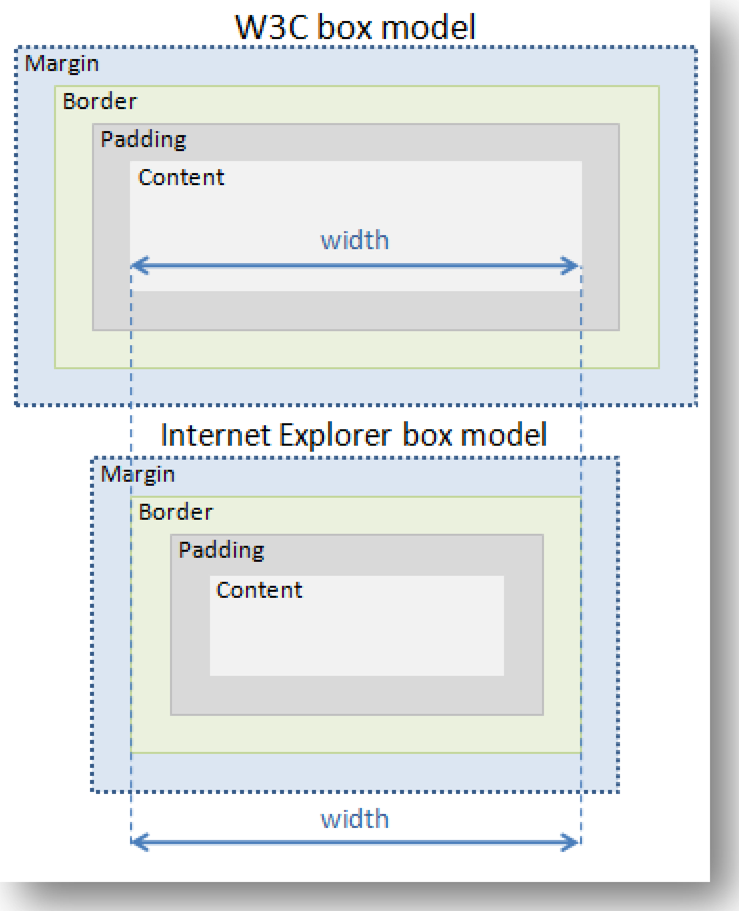
HTML Element's Height =

Margin-Top + Border-Top + Padding-Top

+ Height +

Padding-Bottom + Border-Bottom + Margin-Bottom;

IE Box Model



[IE] HTML Element's Width =

Margin-Left + Width + Margin-Right

[IE] HTML Element's Height =

Margin-Top + Height + Margin-Bottom

- applies only in the absence or invalid DOCTYPE

Block Level Elements

- Used to contain text, inline elements and other block level elements

E.g. div, h1-h6, p, table

Block Level Characteristics

- based on Visual Appearance and CSS Properties

Visual Appearance

- Occupies the available width of the parent container

- Block level elements are stacked vertically

- Always begins in a new line

CSS Properties

Margin	Top	Right	Bottom	Left
Border	Top	Right	Bottom	Left
Padding	Top	Right	Bottom	Left
Width/Height	Width		Height	

🟩 Properties can be set/modified 🟥 Properties cannot be set/modified

Inline Level Elements

- Used to hold text and other inline elements.

E.g. div, h1-h6, p, table

Inline Characteristics

- based on Visual Appearance and CSS Properties

Visual Appearance

- Occupies the enclosing content's width and height

- Inline elements are stacked horizontally

CSS Properties

Margin	Top	Right	Bottom	Left
Border	Top	Right	Bottom	Left
Padding	Top	Right	Bottom	Left
Width/Height	Width		Height	

🟩 Properties can be set/modified 🟥 Properties cannot be set/modified

Inline-Block Elements

- Inherits the Visual Appearance of Inline Elements Visual Appearance and CSS Properties of Block Level Elements

Inline-Block Characteristics

- based on Visual Appearance and CSS Properties

Visual Appearance

- Occupies the enclosing content's width and height

- Inline Block elements are stacked horizontally, until explicit width is specified to occupy the available space

CSS Properties

Margin	Top	Right	Bottom	Left
Border	Top	Right	Bottom	Left
Padding	Top	Right	Bottom	Left
Width/Height	Width		Height	

🟩 Properties can be set/modified 🟥 Properties cannot be set/modified

Margin Vs Padding

Margin

- defines space between border and other outer elements.
- synonymous with cellspacing in tables.
- Margin can take negative values.
- appears outside the element's background-color or background-image.

Usage:

- To provide space between adjacent elements
- To position elements.

Padding

- defines space between border and the element's content.
- synonymous with cellpadding in tables.
- negative values are not applicable to padding.
- appears inclusive of the element's background-color or background-image.

Usage:

- To provide space within the element's border and the content

CSS Positioning:

CSS FLOAT | CSS POSITION | CSS ZIndex

- HTML Elements flow from Top to Bottom, Left to Right.

- To Change the normal flow, and to position elements as required, CSS provides FLOAT, POSITION and ZIndex properties.

- Float(s) are used for horizontal positioning (i.e) to the extreme left or to the extreme right.

- Position(s) are used for horizontal(left, right), vertical(top, bottom) and positioning at a specific location.

- ZIndex is used for positioning along z-axis, commonly referred to as overlapping.

CSS Float

Float Value(s)

- left, right, none, inherit

Usage

- Wrapping text around Images
- Creating multi column layout
- Form Layout
- Navigation

Impact

- Parent Container do not enclose floating children
- Elements floated, becomes BLOCK Level

Issues

- In IE, parent container enclose(s) floating children.
- Double Margin Bug: When the margin of a floated element goes in the same direction as the float, the margin value gets doubled in IE6. fix: use display:inline to the floated element.

CSS Clear

- When an element is floated, subsequent elements will flow along the floated element. CSS Clear is used clear the float and retain normal flow.

- CSS Clear used to enable Parent Container to enclose floating children.

- Clearing floats is possible by using CSS Clear or Overflow or Height property.

- Floats can also be cleared using overflow property; To do so, specify overflow: hidden; to the Parent Element containing floating children.

- height property can also be used to clear floats; we must provide height: xx px; to the Parent Element containing floating children.

CSS Position

Position value

- static, relative, absolute, fixed
- IE6 doesn’t support fixed positioning.

Usage

- Positioning elements relative to Parent Container or Browser Window.
- Element(s) are positioned using TOP, LEFT, BOTTOM, RIGHT values.

position: static

- Default value of position property
- Elements follows the Document flow.

position: relative

- Mostly used to in combination with position:absolute, to relatively position the absolute element.
- Commonly used position value for zIndex.

position: fixed

- Position(s) elements relative to the browser window

E.g. Fixed Footer at the Browser's Bottom

position: absolute

- Used to position element(s) relative to parent container

E.g. Quick Shop Buttons

Impact

- Absolute and Fixed elements are BLOCK Level.
- Absolute and Fixed elements are removed from document flow, so enclosing parent can not contain these elements.

- When positioning, it's important to provide minimum 2 values.

E.g. TOP LEFT or TOP RIGHT or BOTTOM LEFT or BOTTOM RIGHT

ZIndex

Value

- Takes any positive integer

- zIndex values are provided in multiples of 1000

Usage

- To position Elements along the z-axis.

- zIndex works only with positioned elements [absolute, relative , fixed]

- Visually identified through overlapping elements.

Working

- position:relative is commonly used along with zIndex for correct rendering.

- When two non-positioned or positioned elements overlap, the latter element would come on top.
- Positioned elements will always overlap on Non-Positioned elements.

- When two positioned elements[p1, p2] with different zIndex[p1=1000, p2=500] overlap, the element with greater zIndex comes on top.

- In the above case, Element p1 and its Child Elements will always remain on top of Element p2 and its Child Elements [even if children's of p2 have highest zIndex values]

CSS Specificity

CSS Selectors

- When 2 or more declarations apply to the same element, then declaration with the most specific selector will be applied.
- When 2 rules have the same weight, the last rule declared in the stylesheet will be applied.

Order of Precedence - Highest to Least

Inline Style	ID	Class	Element
1000	100	10	1

E.g.

Markup

```
<p id="chapter" class="subsection">
        This is some random content for display.
</p>
```

CSS Declartion

```
p{ color: red;}                  => specificity value = 1;
#chapter{ color: blue; }         => specificity value = 100;
p.subsection{ color: green; }    => specificity value = 1 + 10 = 11;
```

- Here, color attribute of #chapter gets applied, as #chapter has highest specificity value.

!importance

- Used to Override CSS Specificity

- CSS rule with !important takes priority over other matching rules[external css, styles within head section, inline CSS]

E.g.

Markup

```
<p id="chapter" class="subsection">

        This is some random content for display.
</p>
```

CSS Declartion

```
p{ color: red; !important;}          => specificity value = 1;
#chapter{ color: blue; }             => specificity value = 100;
p.subsection{ color: green; }        => specificity value = 1 + 10 = 11;
```

- Here, color attribute of p tag gets applied, since it takes highest priority, overriding any specificity value(s).

IE Bugs & CSS

IE Bugs

- In IE6, DropDown doesn't consider zIndex values and will always come on top of other elements. Possible fix would be to hide the dropdown, or replace it with a input box, or use IFrame.

- Empty DIV takes space in IE. Use line-height:0px; height: 0px

- PNG Images is not supported by IE6. Use HTC fix.

- In Form Layouts, ensure to use <div class="clear"></div>, when float needs to cleared.

- When anchor element is made block level, the clickable region is limited only to the text content. Use a transparent Background Image for the anchor to fix it

Common CSS Used

-Only Block Level Elements can be centered using, margin: 0 auto; width: xx px;

- Inline Level elements can be centered only by applying text-align: center; to its parent container.

- To achieve min-height across browsers, use the following code,
min-height: 300px; height: auto !important; height:300px;

- Vertical centering can be achieved if height and line-height are set to the same value. Used commonly with input elements

- When specifying height of an element to be 100%, ensure the height of its parent is explicitly set.

- If an element has its width set to "100%", it shouldn't have any margins, padding, or borders; if not, it will overflow its parent.

- To hide/show elements, display or visible properties can be used.
display: none; hides the element and removes the space, while visibility: hidden; hides the element but retains the space.

Best Practice

- Use HTML tags for disseminating information and CSS for styling.

- Avoid presentational tags such as b, i, center, font; prefer CSS.
E.g. use 'font-weight: bold;' instead of <b>; use 'font-style: italic' instead of <i>;

- Always start a web page with a valid DOCTYPE.

- Use lowercase for html tags, attributes, values.

- Attributes must be in key/value pairs and the values must be enclosed in quotes.
E.g. <select><option selected="selected">-value-</option></select>

- Every form control should have a corresponding label.
E.g. <label for="input-name">Name<label/> : <input type="text" id="input-name" name="input-name" />

- Follow the Content Model. E.g. UL, OL

- Use reset.css, to set the properties of html tags as required, ensuring cross browser consistency.

- Indent Code as required and provide valid comments as necessary.

Reference

http://www.w3.org/QA/2002/04/valid-dtd-list.html

http://www.w3schools.com/tags/tag_doctype.asp

http://www.thehandcoder.com/markup/the-difference-semantic-markup-vs-structural-markup/

http://www.redmelon.net/tstme/box_model/

http://www.cssnewbie.com/margins-and-padding-use/

http://stackoverflow.com/questions/2189452/when-to-use-margin-vs-padding-in-css

http://www.iraqtimeline.com/maxdesign/basicdesign/principles/prinmargins.html

http://css-tricks.com/all-about-floats/

http://dev.opera.com/articles/view/35-floats-and-clearing/

http://dev.opera.com/articles/view/36-static-and-relative-positioning/

http://dev.opera.com/articles/view/37-css-absolute-and-fixed-positioning/

http://css-tricks.com/absolute-positioning-inside-relative-positioning/

http://www.impressivewebs.com/a-detailed-look-at-the-z-index-css-property/

http://coding.smashingmagazine.com/2009/09/15/the-z-index-css-property-a-comprehensive-look/

http://css-tricks.com/specifics-on-css-specificity/

http://css-tricks.com/ie-css-bugs-thatll-get-you-every-time/

http://www.vanseodesign.com/css/visibility-vs-display/