**WEB 01**

**Element**  
<tag>something goes here</tag>  
  
**빈요소**  
<element/> = <element>

**요소의 중첩**  
<p> something <a>something inside</a> something </p>

**Basic Elements- used only once**  
**<html>** = root element, tells us where the html starts, head and body and all html text must stay within the <html>…</html>

**<head>** = first element inside the html, metadata inside the html, not seen on the website, like the brain, provides information but not seen from the outside

**<body>** = what’s seen on the website, if we input body it shows up on the browser  
<title> = html’s title shown on the tab  
  
<!DOCTYPE html>  
<html>

<head>

<title></title>  
</head>  
   
<body>  
</body>

</html>

**<!DOCTYPE html> DTD 선언문**  
= can be outside of the html bracket, applies to HTML5, HTML4 and before has a more complex <!DOCTYPE html ……….>, erase the content behind html to convert the doc to HTML5

**Text Encoding**  
= text shows up in the user’s language and computer  
**<meta charset= “utf-8”/>** for HTML5, must be above the title, so that it affects the title when the text shows up on the site

**Attribute**  
= applies only on start tags when necessary, not always  
<tag attribute= “value”> *content* </p>  
i.e. <tag ref= <http://www.astellalee.com>> content </a>

<tag attribute=“value”> 내용 </p>

<p class= “value1 value2”> 속성 </p>

**언어 선언**  
<html lang=“ko”>

————————————————————————————————————

**WEB 02\_TEXT**

**HTML TEXT**

**<h1> ………. <h6>**varies in sizes for diff headers, shows up in BOLD  
h1=biggest / h6=smallest  
Goes inside the <body></body>

**<p> </p>**  
내용을 설명하는 요소   
  
**<br/>**line break, replace the “Enter” key

**<hr/>**line that goes across the page

**<address> </address>**when saving the company’s address, email, fax, phone number  
show up in *italics*

**<a>**Hyperlink, appears blue and shows a underline, and turns purple after the link is clicked once.  
<a href=“<https://www.astellalee.com>"> 내용 </a>  
<a href=“#”> 내용 </a>  
  
<a target=“\_blank”> = opens a new tab  
<a target=“\_self”> = opens in the current tab

<a title=“제목”> 내용 </a> = 링크에 대한 설명  
**절대결로**  
http:// 포한되어 있는 경로   
Not safe to use, because the web link could disappear if the creator changed the link

**<a href= “ “> TO A FOLDER**  
입력하고있는 파일을 기준으로 routing  
  
If the files are in the **same file**  
<a href= “index.html>  
  
If the file is **inside another folder**  
<a href= “folder/index.html”>  
  
If the file is **outside** **another folders**  
<a href= “../index.html”>

————————————————————————————————————

WEB 03\_IMAGE

<img/>  
Image

<img src=“img.jpg” alt=“이미지 입니다.” width=”100” height=”auto”/>   
  
src = file name of the image, bring images in from personal file b/c images online change often  
alt = shows up when the image in unable to load, description of the image goes here  
width = pixel  
height = pixel

Keep a “img” folder at the main level

————————————————————————————————————

WEB 04\_LISTS

<ul> <ul/>  
**Unordered list**  
Iist, it doesn’t make the list, but represents a type of list  
Black dot bulleted list format

<ol> <ol/>  
**Ordered list**  
numbered list

<li/>  
L**ist items** <li> MUST goes inside <ul/> or <ol/>

<dl> </dl>  
**Description List**  
another type of list   
NO Bullet of Number  
  
<dt></dt>  
**Desription Term**  
Non hyphenated list   
  
<dd></dd>  
**Description detail**hyphenated list

————————————————————————————————————

WEB 05\_TABLES

<table></table>  
**Tables group**  
Iist, it doesn’t make the list, but represents a type of list  
Black dot bulleted list format

<table border=“1”> 내용 </table>

DO NOT USE “BORDER” b/c tables are now hosted through CSS, not html.  
You can’t make columns (vertical), you can only make rows (horizontal)

<caption>  
**optional**

<tr>  
1. **Hold the Row content <th> and <td>(horizontal)**You can’t make columns (vertical), you can only make rows (horizontal)

<th>  
2. **Table Headline  
Title of the rows(horizontal)**You can’t make columns (vertical), you can only make rows (horizontal)  
Will be **center aligned** to the cell in **bold**

<td>  
3. **Table Details  
Contents of the rows(horizontal)**You can’t make columns (vertical), you can only make rows (horizontal)  
Will be **LEFT** aligned to the cell in normal thickness

**table, th, td** have borders around them, remember when translating it to CSS

<thead>  
T-head  
Groups <tr> table rows

<tbody>  
t-body  
Groups the <td> or <th>  
Can be used multiple times  
<tfoot>  
t-foot  
Groups the <td> or <th>

1.2.3.  
<thead> <tfoot> <tbody>

————————————————————————————————————

WEB 07\_FORMS

<form>  
Form groups  
MUST NOT put elements like <h1> or <p> ONLY <fieldset> under FORM

<fieldset>  
Different form groups i.e. Customer Info / Customer Orders / Customer Churn  
Here you can write in other elements like <h1> <p>

<legend> ….optional  
legend for <fieldset> form group  
  
<label>  
title for the individual content going in the form i.e. ID / PW / Login button  
If there are no content, do not use <label>   
  
**<input/>**  
<input type=“text”/>  
<input type=“password”/>   
When info must be hidden such as the password, we use “Password” input type

<input type=“submit”/> ………전송 버튼 Can be used ONLY when there is input info to send to server  
<input type=“reset”/> ………reset button  
<input type= “button”/> ………creates a button  
<input type= “image”/> ………makes an image become a button (use <img/> to replace ‘image’)

<input type=“file”/> ………opens desktop file browser  
  
<input type=“radio”/> ………radio selector individual names  
<input type=“checkbox”/> ………multi-select box individual names

<input name=“name”/> …groups the category of the button choices. i.e. boy/girl = GENDER  
<input value=“값”/> ……designates the individual names of buttons

————————————————————————————————————

WEB 08\_FORMS

**TEXT BOX <textarea>  
inputing a chunk of text**

<textarea cols=“10”> </textarea>  
cols= column width in word count  
cols = HTML not html5 (in chrome, cols #+1 word count will be added)  
  
<textarea rows=“10”> </textarea>

**Dropdown Form Box**<select> ….. must be written inside the <fieldset>  
  
<optgroup> ….. dropdown options  
<optgroup label=“text”></optgroup>  
  
<option>  
<option value=“값”>값</option> …… goes into <optgroup> brackets

**INLINE & BLOCK  
Inline** - takes up only the space where the text is i.e. input types, links   
**Block** - designated unseen area where the text will go i.e. h1-h6, p,

You can’t put block inside inline. (block is bigger than inline)  
You CAN put inline inside block.

<div>  
Block type- Content grouping that can contain ‘block’ and ‘inline’

<span>  
Inline type- group only the selected contents  
We can use <span> to select specific items (like a single word in a paragraph) to give it a specific CSS.  
  
**&** = must type in **&amp;  
Q&A = Q&amp;A**

——————————————————————

WEB 10\_CSS

**E{font-style: italic;}**   
options: normal, italic, oblique

**E{font-weight: bold;}**   
options: bold, normal  
  
**E{letter-spacing: 5px;}**increase letter spacing size  
  
**E{word-spacing: 5px;}**spacing between words  
  
**E{line-height: 5px;}**line-height spacing

**E{text-indent: 5px;}**paragraph indentation  
  
**E{text-transform: uppercase;}**uppercase, lowercase, capitalize(aka title-case)  
  
**E{text-align: left;}**left, right, center, justify  
  
**E{text-decoration: 위치** 모양 색상**;}**underline, overline, line-through, done  
  
**E{text-decoration:** 위치 **모양** 색상**;}**solid, double, dotted, dashed, wavy  
  
**E{text-decoration:** 위치모양 **색상;}**컬러 고유 명칭, 16 basic colors, rgba, hsla ***i.e. E{text-decoration: underline dotted red;}*E{color:** red**;}**font color  
  
**E{background-color:** red**;}**text’s background color  
  
**E{width: 100px; height: 100px;}**can be just width or height, or both together  
\*CAN NOT apply width/height to inline-items like hyperlinks

DIV / SPAN  
<div>  
Block type- Content grouping that can contain ‘block’ and ‘inline’

<span>  
Inline type- group only the selected contents  
We can use <span> to select specific items (like a single word in a paragraph) to give it a specific CSS.  
Margin cannot be used on SPAN (inline).  
Padding can be used on SPAN.

<div></div>  
CANNOT just include text inside div.   
**<div><p>** blah blah blah **</p></div>**  
————————————————————————————————————

WEB 11

**Margin**- outside of padding  
E{margin: 50px;} - same 50px spacing all around the border  
E{margin: 50px 20px;} - top-bottom / left-right  
E{margin: 50px 20px 10px 30px;} - top-right-bottom-left (clockwise)

E{margin: 50px;}  
E{margin: 50px;}  
E{margin: 50px;}  
E{margin: 50px;}

E{margin: auto;} - centers to the page

**Border** between Margin and Padding  
E{border: 50px solid red;} - width-style-color  
  
E{border-width: 50px;} - same width all around  
E{border-color: red;}   
E{border-style: solid;} - solid, dotted, dashed

E{border-top: 50px solid red;}  
E{border-bottom: 50px solid red;}  
E{border-left: 50px solid red;}  
E{border-right: 50px solid red;}

**Padding**- boarder around content  
E{padding: 50px;} - same 50px spacing all around the border  
E{padding: 50px 20px;} - top-bottom / left-right  
E{padding: 50px 20px 10px 30px;} - top-right-bottom-left (clockwise)

E{padding-top: 50px;}  
E{padding-left: 50px;}  
E{padding-right: 50px;}  
E{padding-bottom: 50px;}

**Content**- width and height are only for the content space————————————————————————————————————

WEB 12

**LIST BULLET**

{

list-style-type: disc;

}

disc -black dot

circle -outlined dot

square

decimal

lower-alpha -alphabet list (a,b,c,d,e,...)

upper-alpha

lower-roman -roman numerals

upper-roman -roman numberals

none

**LIST POSITION**

{

list-style-position: outside;

}

outside -stays the same

inside -indented list

{

list-style: disc inside;

}

Only when both are applicable, use the {list-style: ---- ----;}

\* {margin: 0; padding: 0; list-style:none;}

We must do this to get rid of the edge borders

**Parent/ Child relationship in coding**

**Selecting all the elements under an element**

ul li {property: value;} ....... all the li with property/value elements under ul

**Selecting ONLY the child under the parent**

ul > li {property: value;} ..... only the li with the property/value under ul

————————————————————————————————————

WEB 13

**div, li {property: value;}** ……. Selects all div and li

**Inline** …….only covers the select area of the element

…….can not create padding and margin around inline items

**Block** …….covers the horizontal width of the element

표현 속성  
**E{display: inline;}**  
…….. inline, block, inline-block, none

i.e. <a></a> is an inline item, so you can’t give it another {display: inline;} characteristic   
  
You can make <div></div> OR <p></p>, which have BLOCK characteristic, an {display: inline;} character to make it work like an INLINE element.   
  
**display: none;** ………makes the entire element disappear

————————————————————————————————————

WEB 14\_Link states

a**:link {property: value;}** …….before link is clicked / only used with ‘a’

a**:visited {property: value;}** …….the way link looks after it’s clicked / only used with ‘a’

**p:hover {property: value;}** ……. Link on hover / can be used with ‘class’  
i.e. div.box2:hover > a {

background-color: yellow;

}

**p:active {property: value;}** ……. When link is clicked

i.e. div.active{  
width:   
background: color;  
….  
…

}

Input:focus {property: value;}……………form box, when selected

————————————————————————————————————

WEB 15\_배치 속성

E**{position: static;}** …….”position” = static, **relative, absolute, fixed**

**Static and relative** — can be the same

**Relative** ——- to its current position, parent with relative position can contain it’s children

**Fixed** —— like a fixed navigation, relative to the entire screen space, must give it a position

**Absolute** —— relative to the parent element, other elements will not see its space as an element, it can be pushed anywhere onto behind other elements (relative to the sac

E**{z-index: 1;}** ……. Arrangement IN FRONT or BEHIND an element relative to its current position (relative, absolute, fixed)

**i.e. E {**  
 **position: relative;**

**top: 50px;**

**}**

**————————————————————————————————————**

WEB 16\_Layout (float and clear)

E**{float: left;}** …… = left, right, none / for responsive web

**Left:** 1-2-3-4

4-3-2-1**:Right**

**Float and Block**   
- If float is on top of block, the block has a tendency to go above/on top of the Float

* If block is on top of float, then float properties won’t affect anything above it

**Float and Inline**

* in-line items will surround the float items

**Floats**

* Elements with float, will stay in their own places
* If non-float elements, are w/ float elements, then they will ignore each other’s properties

**Float Only**

* when positioning multiple items. Left, right relationship. Don’t use when just making on element, with no left right relationship items

**Float and Clear** ……… elements go overlap float, when given clear property  
- Clear and float always go together

- *Give an element Both float/clear, when you want it the first thing on the left aligned*  
**E{clear: both;}** ………..left, right, **both**

**Float: comes first, then position:**

When finishing a float, create a new “div or br“ element with css / clear:both

**Html: <br/> | CSS: br {clear: both;}**

**Html: <div class=“c”> </div> | CSS: div.c {clear: both;}**

**————————————————————————————————————**

WEB 17\_Layout (image links)

**Background-“combined”**

**E {background: red url('') no-repeat center/25px scroll;}**

{background: BGcolor ImgURL Repeat Position/Size Attachment;}

**Background-image**

**E {background-image: url('route');}**

**E {background-image: url('route'), url(‘route’);}** .......................images shown in the order listed

**Background-repeat**

**E {background-repeat: repeat;}**

* no-repeat, repeat-x, repeat-y

**Background-position**

**E {background-position: top left;}**

* top, bottom, left, right, center, px, %

- **Top** = top center of the box

- **Bottom** = bottom center of the box

- **Left** = left aligned (in the vertical center of the box)

- **Right** = right aligned (in the vertical center of the box)

- **Center** = center of the box

**- px** = must give both x-axis and y-axis value

- **%** = is for responsive web

**Background-size**

**E {background-size: auto;}**

- auto, px, cover, contain

- background-size: 50px ……………… width only (height auto)

- background-size: 50px 50px ……….. width / height

- “cover”…………………………………. image.cropped to the box's width/height

- “contain” ……………………………… contains image w/i the box

**Background-attachment**

**E {background-attachment: scroll;}**

- scroll, fixed

**Background-“combined”**

**E {background: red url('') no-repeat center/25px scroll;}**

{background: BGcolor ImgURL Repeat Position/Size Attachment;}

**————————————————————————————————————**

WEB 18\_visibility, cursor, overflow, border-collapse, layout

**E{visibility: hidden;}**

**E{cursor: default;}**

- default, pointer, text

**E{overflow: visible;}**

- visible, hidden, scroll

- applies to the parent element

- when a image is bigger than the parent box

- visible: show the full image, doesn't consider parent box size

- hidden: hides the areas that overflow out of the parent box

- scroll: allows users to move vertically and horizontal within the parent box

**E{overflow-X: visible;}**

- only applies to the x-axis when the image overflows

\*used in conjunction with overflow-Y\*

**E{overflow-Y: visible;}**

**E {border-collapse: collapse;}**

- collapse, separate

- merges the multiple borders around each element in a table

- collapse: merges the lines

- separate: shows all the lines around each elements

**LAYOUT**

1. **header** id=“header" / h1.logo / ul.infolink

- h1.logo (class= logo) , is a link, and goes in as a img

- h1.logo > a(link back to homepage) > img

**2. contentsArea** id= "contentsArea"

- ul.infolink > li > a

**3. footer** .................. id="footer"

- div.nav > ul.mainMenu > a

HW: website

header: float and block and text alignments