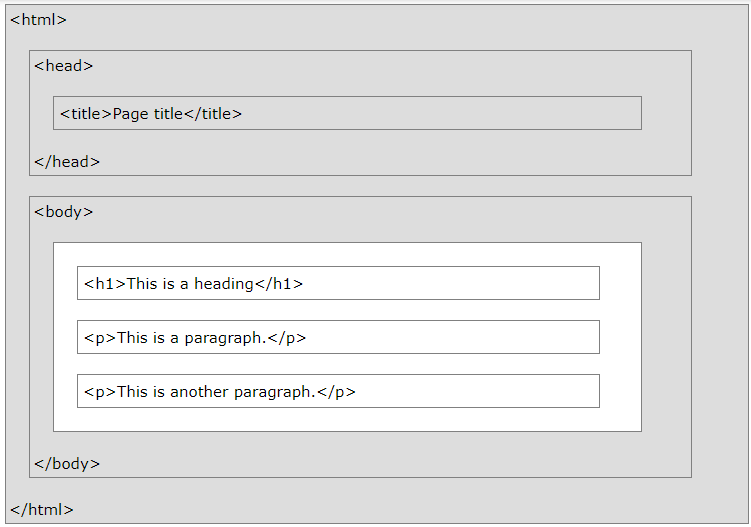
HTML5 정리

20162532 배정식

* **HTML의 목적**1. 웹문서를 만들기 위해서 사용  
  2. 웹문서를 효과적으로 전달하기 위해서 사용 hyper text 이용(하이퍼링크)
* **HTML(Hyper Text Markup Language : Markup tag(내용을 출력하기 위한 지시문)를 사용)**tag : 지시문(명령어) - 요소, 앞에있는 Tag를 start tag 라 하고, 뒤에있는 Tag를 end tag이다.  
  웹 브라우저에서 Tag는 실제로 나타내지는 않지만 컨텐츠를 표현하는 지시문으로 사용  
  **태그의 사용 형태 : <tagname> 내용 </tagname>**
* **기본적인 Tag 종류**  
  <!DOCTYPT html> : 문서타입이 html5 웹문서라는 선언, 제일 윗줄에서 선언해야 한다. 대소문자 상관 x  
  <html> : html 페이지의 최상위 루트에 해당하는 요소, head 와 body 를 사용  
  <head> : 웹문서 자체의 정보를 표현, : html 문서에 대한 정보를 보여주는 태그  
  <title> : 보통 head에 포함되며 웹문서의 제목이 되는 부분  
  <body> : 실제 컨텐츠 중에서 화면에 보여지는 본문 부분, h1과 p를 사용  
  <h1> 제목 </h1> : 헤딩 태그 = 제목처럼 크고 굵은 글씨를 나타낼 때 사용, h1~h6 크기순서  
  <p> 본문 </p> : p 태그 = 내용을 그대로 출력할 때 사용



* **HTML 편집기 (메모장을 이용해서 편집할 수 있다)**  
   : 저장할 때 이름은 제목.htm or 제목.html 으로 저장  
   저장 타입은 모든 파일로 바꾸기  
   인코딩은 UTF-8 : 한글파일이 깨지지않고 정확히 전달
* **HTML 요소 : 시작태그부터 내용 종료태그 까지를 요소라고 한다.**

1. **H1 태그 ( 헤딩태그) : <body> 에 포함**  
   <h1> ~ <h6> 까지 있고 h1 이 가장 크고 순서대로 크기가 작아진다.  
   <h1>은 가장 중요한 제목, <h6>은 비교적 덜 중요한 제목을 입력할 때 사용한다.  
   <h1 style=”font-size:00px;”> 내용 </h1> : 제목의 글자 크기를 원하는 사이즈로 조절 가능하다.  
   # <hr> : 컨텐츠의 영역을 분리하기 위해서 수평선을 그어주기 위해 사용 (빈요소)
2. **P 태그 : <body> 에 포함**  
   <p>가 끝나는 부분과 다음 <p> 사이에는 문단을 나눠주는 공백이 존재한다  
   문단의 내용에서 발생하는 공백들은 많이 쓰더라도 한 개의 공백 문자로 간주해서 출력된다  
   <br> or <br /> : 줄 바꿈 태그 = 끝마침 태그와 내용이 없이 줄만 바꿔주는 태그  
   <pre> 내용 </pre> : 사전형식화 텍스트 => 입력한 형식 그대로 출력할 때 사용
3. **Head 태그 : 웹문서에 대한 데이터 (메타 데이터) 에 대한 컨테이너 태그이다.**
4. **Title 태그 :** 웹 페이지 탭에 나오는 웹문서의 이름을 나타내는 태그이다. (<head> 안에 포함
5. **Style 태그 :** <body> 태그에서 사용하는 태그들의 스타일을 정의하는데 사용 ( <head> 안에서 사용)
6. **Link 태그 :** 외부 CSS 파일을 불러와서 해당 부분에 스타일을 적용할 때 사용  
   <link rel =”stylesheet” href =”파일이름” > : rel 속성에는 파일의 형태를 입력
7. **Meta 태그 :** 페이지 설명, 키워드, 문자, 저자 등 메타 데이터를 정의하는데 사용(광고 효과)  
   <meta charset = “UTF-8”> : 사용하는 문자를 UTF-8 로 지정 ( 한글을 잘 읽는 문자)  
   <meta name =”description” content = “Free Web tutorials”> : 웹페이지의 설명을 정의  
   <meta name =”Keywords” content = “HTML, CSS, XML, JavaScript”> : 검색 엔진을 정의  
   <meta name =”author” content =”작성자 이름”> : 페이지 작성자 이름을 정의  
   # <meta http-equiv = “refresh” content = “30”> : 30초마다 문서를 새로고침 한다.  
   **viewport : 웹 페이지에서 사용자가 볼 수 있는 영역을 말한다. 장치에 따라서 크기가 달라진다**.  
   <meta name =”viewport” content =”width=device-width, initial-scale=1.0”>  
   : viewport 라는 이름으로 설정해 놓은 부분의 크기( 너비 , 확대 정도) 를 설정한다.
8. **Base 태그 :** 사용될 속성의 기본값을 정해 놓을 때 사용   
   ex) <head> 태그에서 사용해서 href 속성, target을 지정해 놓으면   
   <body> 태그에서 사용하는 src 속성, href 속성에서 필요한 값을 지정해 놓은 값으로 사용

* **HTML 속성 : HTML 요소에 추가적인 정보**  
  속성을 사용할 때 시작태그 안에서 사용한다. (name = “value” 형태)

1. **Href 속성** : 링크 태그 (하이퍼링크)  
   <a href=”URL”> 화면에 보이는 텍스트 </a> : 앵커 태그 = 텍스트를 클릭하면 URL로 이동   
   => 실제 이동할 링크는 URL 자리에 입력하고 화면에 보일 텍스트는 링크제목에 입력한다.
2. **Src 속성** : 이미지 태그에서 사용 (img 태그는 시작태그만 존재하고 끝마침 태그는 없다 : 빈요소)  
   <img src=”그림파일의 이름” alt=”대체텍스트” width=”너비” height=”높이”>   
   => alt 속성 : 그림이 깨질 경우 대체 텍스트를 보여주기 위해 사용(시각장애 분들을 위해)
3. **Style 속성** : 제목이나 문단의 스타일을 바꿀 때 사용 (CSS의 기능을 사용)  
   형식 : <tagname style=”스타일명령어 : 값;”> ;사용필수  
   <body style=”background-color : 색깔;”> 본문 <body> : 배경색깔을 지정  
   => 전체 배경을 바꾸기 위해서는 바디 태그에 입력해야 하며 한 줄 씩 개별적으로 바꾸고 싶으면   
   헤딩태그나 p태그에 각각 입력한다.  
   <p style=”color : red; font-size : 100px;”> 내용 </p> : 내용의 색깔과 폰트 크기를 조절한다.  
   <p style=”font-family : 글씨체;”> 내용 </p> : 내용이 입력한 글씨체로 바뀐다. Ex) 궁서체, 굴림체  
   <p style=”font-size : 200%;”> 내용 </p> : 입력한 숫자에 따라 내용의 크기를 확대 축소 할 수 있다.  
   <p style=”text-align : center;”> 내용 </p> : 텍스트의 정렬방식을 설정한다. Ex) 가운데정렬
4. **Lang 속성** : 최상위 html 태그에서 사용  
   <html lang=”언어”> : 언어 속성 선언 = 입려한 언어로 속성을 설정
5. **Title 속성** : 내용의 부연설명을 해주기 위해 사용  
   <p title=”설명할 내용”> 보여지는내용 </p> : 내용에 마우스를 대면 입력한 부연설명이 보인다.

# 지시문, 명령문을 입력할 때는 항상 소문자로 입력한다.  
# 태그속에서 URL 이나 값을 입력할때는 항상 “ ” 을 이용해야한다.  
# 뒤에 오는 끝마침 태그가 없는 태그 : empty 요소, 빈 요소라고 한다.

* **HTML 텍스트 형식**

<p> <b> 내용 </b> </p> : 내용의 글씨체를 두꺼운 글씨로 출력  
<p> <strong> 내용 </strong> </p> : 내용의 글씨체를 두꺼운 글씨로 표현 중요한 내용에 사용  
<p> <i> 내용 </i> </p> : 내용의 글씨체를 기울인 글씨로 출력  
<p> <em> 내용 </em> </p> : 내용의 글씨체를 기울여서 표현 강조하고 싶을 때 사용  
<p> <sub> 내용 </sub> </p> : 내용의 글씨체를 아래 첨자로 출력  
<p> <sup> 내용 </sup> </p> : 내용의 글씨체를 윗 첨자로 출력  
<p> <small> 내용 </small> </p> : 내용의 글씨체를 작은 글씨로 출력  
<p> <mark> 내용 </mark> </p> : 내용을 형광 펜처럼 마킹 된 채로 출력  
<p> <del> 내용 </del> </p> : 내용을 중간에 취소 선을 그어서 삭제한 내용으로 표현  
<p> <ins> 내용 </ins> </p> : 내용을 밑줄 그어서 표현 추가된 내용을 정의할 때 사용

* **HTML 인용 및 인용 요소**  
  <blockquote cite="URL"> 내용 </blockquote> : 입력한 사이트로부터 긴 내용을 들여쓰기해서 인용  
  <p> 내용 <q> 내용 </q> </p> : q 사이의 짧은 내용을 “” 이용해서 인용  
  <p>내용 <abbr title="요약할 내용">요약한 약자</abbr></p>: 입력한 내용을 요약하여 약어로 표현  
  <address> 내용 </address> : 내용을 주소로 인지하여 이탤릭체로 표시  
  <cite> 내용 </cite> : 내용을 작품의 제목으로 정의할 때 사용하며 이탤릭체로 표시  
  <bdo dir=”rtl”> 내용 </bdo> : 내용의 출력방향을 결정 right to left 오른 -> 왼( 반대는 ltr)
* **HTML 주석 태그** : 소스 코드 설명 주석을 달기 위해서 사용하는 태그 (실행화면에는 나타나지 않음)  
  <!—내용 --> (html 에서의 주석) , /\* 내용 \*/ (CSS 에서의 주석)
* **HTML 색상**  
  #색상 표현하는 방법: 색상 이름 , RGB, HEX, HSL, RGBA, HSLA  
  # 색상이름 : 색깔의 이름이나 과일의 이름으로도 가능  
  # rgb(빨강, 초록, 파랑) : 빨강 초록 파랑의 채색을 섞어서 사용 (0~255)  
  # rgba(빨강, 초록, 파랑, 투명도) : rgb에 추가로 투명도도 조절할 수 있다. 투명도(0~1.0)  
  # HEX : 해당되는 색상의 16진수 값을 입력하여 사용 (#000000 ~ #ffffff) =>#\_R\_G\_B  
  # hsl(색조, 채도, 밝기) : 입력한 색상 채도 밝기에 의해서 색상을 출력 ( 0~360, 0~100%,0~100%)  
  # hsla(색조, 채도, 밝기, 투명도) : hls에 추가로 투명도도 조절할 수 있다.  
  <p style=”background-color : rgb(0,0,255);”> 내용 </p> : 내용의 배경을 입력한 색상으로 출력  
  <p style=”color : #00ffcf;”> 내용 </p> : 내용의 글자 색을 입력한 색상으로 출력  
  <p style=”color : hsl(50,50%,100%);”> 내용 </p> : 내용의 글자 색을 입력한 색상으로 출력  
  <p style=”border: 15px solid 색상이름;”> 내용 </p> : 내용의 테두리를 15px의 실선으로 출력  
  # solid(실선), dotted(점선), dashed(대쉬기호)
* **HTML 스타일 – CSS**

1. **인라인 -HTML 요소에서 스타일 속성 사용**   
   🡪 style 속성 이용 : <h1 style=”color:색상”> 내용 </h1>
2. **내부 - 섹션 의 <style>요소를 사용하여<head>**   
   🡪 head 섹션 내부에 style 섹션을 열어서 본문의 배경 글자색 등을 미리 설정  
   <head> <style>  
   body {background-color: red;} => 본문 전체의 배경을 설정  
   h1 {color: green;} => h1 태그의 글자 색을 설정  
   p {color: blue;} => p 태그의 글자 색을 설정  
   </style> </head>
3. **외부 -외부 링크를 통해 현재 웹페이지의 스타일을 설정**  
   => <link rel=”파일타입” href=”파일명”> (head 태그안에 사용)   
   # <link rel=”stylesheet” href=”/html/styles.css”> : 폴더와 파일의 위치가 다를 때 사용  
   # <link rel=”stylesheet” href=”styles.css”> : 폴더와 파일의 위치가 같을 때 사용
4. **CSS 테두리**  
   <style> P{ border: 3px solid 색상;} </style> : 테두리 크기와 선 종류, 색상을 입력하여 출력  
   <style> P{ padding: 30px;} </style> : 테두리 내의 여백의 크기를 조절  
   <style> P{ margin: 50px;}} </style> : 테두리 외부의 여백의 넓이를 조절

* **HTML 링크 : 이미지나 텍스트를 클릭하면 다른 웹페이지로 가도록 하는 기능**  
  <a href=”URL”> 내용 </a> : 화면에 보이는 내용을 클릭하면 URL 주소로 이동  
  # href속성은 링크의 대상 주소를 지정

1. **로컬 링크**   
   <a href="파일명">내용</a> : 같은 웹사이트에 대한 링크는 파일명을 입력
2. **Target 속성**  
   <a href=”URL” target=”\_blank”>내용</a> : 해당 링크를 새 창이나 탭에서 보여줌(\_self : 현재열린창)
3. **이미지 링크**  
   <a href="이동할 주소"><img src="사진파일의 이름" alt="사진을 대체할 텍스트";> </a>  
    :사진을 클릭하면 입력한 링크주소로 이동
4. **이메일 링크**  
   <a href=”mailto : 이메일주소”> 내용</a> : 내용을 누르면 입력한 이메일 주소로 메일을 전송
5. **링크 버튼**  
   <button onclick="URL=바꿀주소">버튼이름</button> : 버튼을 누르면 파일주소의 링크로 이동  
   # onclick : 마우스로 클릭하는 사건이 발생하면 뒤에 있는 자바스크립트 코드를 실행
6. **링크 색상 :** 보여지는 링크의 스타일을 설정  
   => head 태그 안에 있는 style 태그에 입력하며 가짜요소 a를 이용해서 각각의 스타일을 정한다.  
   **I. 텍스트**   
   Ex) a : link{ : 한번도 누르지 않은 링크에 대하여   
    color: green; : 텍스트 색깔 설정  
    background-color : transparent; : 배경 색깔 설정  
    text-decoration : underline;} : 텍스트 밑에 밑줄 긋기  
   # visited (들어갔었던 링크에 대하여), hover (마우스를 올렸을 때), active (클릭했을 때)  
   # underline : 클릭하는 링크에 밑줄을 생기게 한다.

**II. 버튼**  
Ex) a : link, a : visited { : 버튼을 클릭하지 않았거나, 클릭 했을 때   
background-color : blue; : 배경색상을 설정  
color : white; : 글자 색상을 설정  
padding : 15px 25px; : 테두리 내부 여백의 크기를 설정 ( 위, 옆 순서)  
text-align : center; : 텍스트의 정렬을 설정  
text-decoration : none; : 텍스트의 효과를 설정  
display : inline-block;} : 버튼 외부의 여백을 설정  
a : hover, a : active { : 마우스를 올렸을 때, 클릭 했을 때  
background-color : red;}  
# a 태그를 중복으로 설정할 수 있다.

1. **북마크 링크**

<h2 id=”위치의 이름">내용</h2> : 책갈피 위치 설정  
<a href="#설정해 놓은 위치">링크 텍스트</a> : 링크 텍스트를 누르면 책갈피 위치로 이동

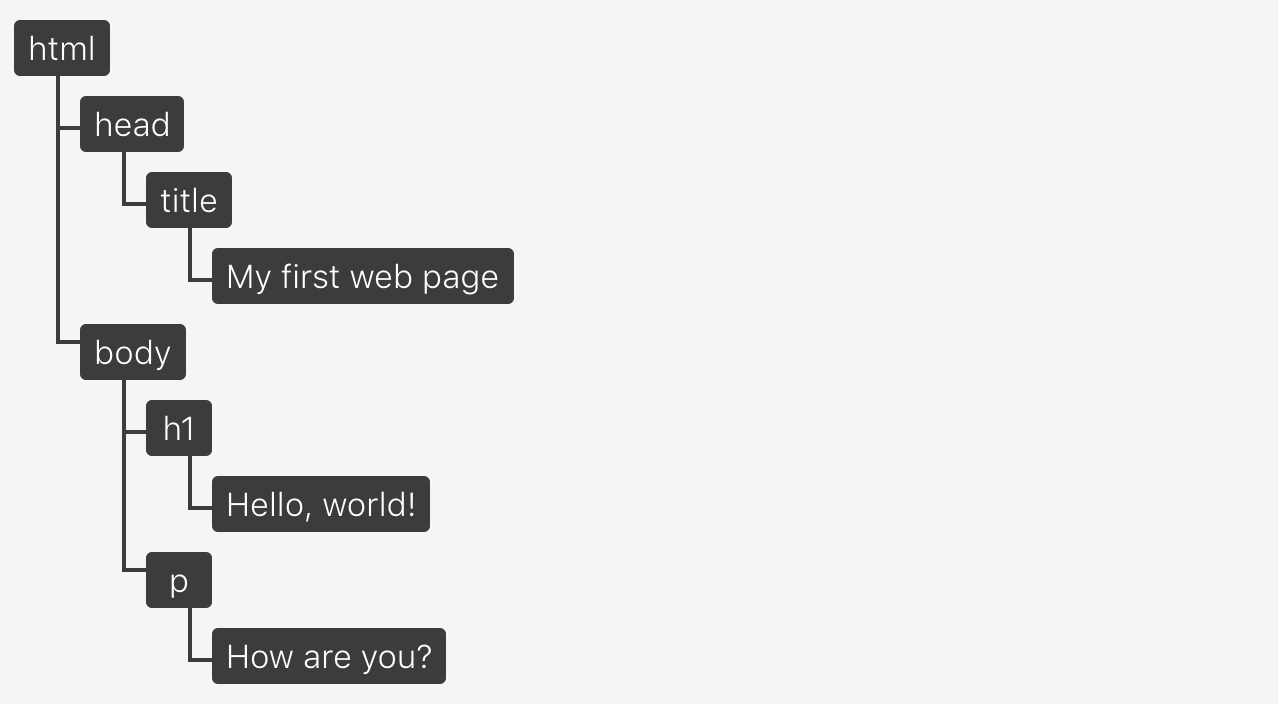
* **HTML 이미지**

1. **<img src=”사진 링크" alt="대체 텍스트" style="width : 넓이px; height : 높이px;"**
2. **<img src=”사진 링크" alt="대체 텍스트"  width="넓이" height="높이";"**  
   :입력한 URL의 사진이 출력.그래픽이 깨졌을 경우 대체 텍스트가 나오고 스타일은 크기를 설정  
   🡪 head 섹션에 스타일을 설정 해놨을 경우 그림의 크기가 바뀔 수 있으므로 (1)번을 사용한다.  
   # 테두리선을 설정하기 위해서는 스타일 태그 안에 border :00; 를 추가한다.  
   # 이미지의 위치를 설정하기 위해서는 스타일 태그 안에 (float : 위치;)를 추가한다.  
   # 폴더의 위치를 참조할 때 /폴더의 위치/를 쓰고 현재 폴더일 때는 ./폴더위치/ 를 사용한다.
3. **이미지 맵 :** 지정한 이미지 영역( usemap, area ) 에 하이퍼링크 기능 ( href) 을 넣을 때 사용  
   <img src=”사진파일" alt="대체텍스트" usemap="#workmap" width="400" height="379">  
     
   <map name="workmap"> : usemap 속성에서 사용 할 위치 저장  
     <area shape="rect" coords="시작좌표,끝좌표" alt="대체텍스트" href="URL"> : rect = 사각형,cords =좌표  
     <area shape="circle" coords="중심좌표,반지름" alt="대체텍스트" href="URL "> : circle = 원형  
   </map> : shape 뒤에 모양을 결정한 후 사진의 좌표에 따라서 좌표값을 입력 ( 왼쪽 위 를 기준으로 x,y)  
   # <area> 태그 안에 해당 영역의 모양, 좌표, 대체 텍스트, URL(클릭 시 해당 링크로 이동) 을 입력
4. **이미지 맵 및 JavaScript**  
   onclick = “함수이름()” : 클릭을 했을 때 함수를 실행 시킨다.  
   <script>  
   function 함수이름 () { : 함수 선언   
   alert(“ 보여줄 메시지”); } : alert = 함수가 실행 되었을 때 페이지에 안내 메시지를 띄운다.  
   </script>
5. **배경 이미지 : body 태그 안에 있는 태그에 사용되어 해당 문단의 배경을 설정**<div style = “ background-image : url (‘ 그림파일의 이름 ‘); “>   
   # style 태그를 head 요소에 있는 <h1> 이나 <p> 에 추가해서 사용할 수도 있다.  
   => body { background-image : url ( ‘ 그림파일의 이름’) ;  
   # 해당 그림의 크기가 작을 경우 body에 선언하게 되면 전체 화면을 채우기 위해 반복된다.  
   background-repeat : 반복되는 그림의 스타일을 설정할 수 있다.  
   => no-repeat : 반복하지 않고 한번만 나옴, repeat – x : x 축 방향으로 반복,  
    repeat – y : y 축 방향으로 반복 repeat – xy : xy 축 방향으로 반복  
   background-attachment : fixed ; => 그림이 화면이 움직이더라도 고정이 되도록 설정  
   background-size : cover; => 크기가 화면에 꽉 차도록 설정  
   background – size : 100% 100% ; => 크기가 회면의 크기에 따라서 달라지도록 설정(가로, 세로)  
   **# <picture> 요소 : 장치 사이즈에 맞게 그림의 크기를 조절해주는 기능**  
   <picture>  
     <source media="(min-width: 650px)" srcset="img\_food.jpg">  
     <source media="(min-width: 465px)" srcset="img\_car.jpg">  
     <img src="img\_girl.jpg">  
   </picture> : 화면의 넓이가 650px 이상이면 food 사진을 보이고,   
    화면의 넓이가 465px 이상이면 car 사진을 보이고,   
    화면의 넓이가 그 이하이면 girl 사진을 보인다.  
   **# img 요소는 항상 picture 요소 마지막에 사용**  
    <source srcset="그림파일. png">   
     <source srcset= “그림파일. jpg">   
     <img src=" 그림파일.gif" alt="대체 텍스트" style="width:auto;">  
   => 해당 브라우저가 인식하는 첫번째 이미지 형식을 출력한다.( picture 요소의 기능)

* **HTML 테이블 : 데이터를 행과 열로 정렬 할 수 있다. => <table>**  
  <tr> : 행렬의 행을 나타내는 태그. 테이블의 값을 입력하려면 항상 <tr> 태그안에 입력해야 한다.  
  <th> : 행렬의 열을 나타내는 태그로 행의 제목을 나타내는 태그. 굵은 글씨 가운데 정렬  
  <td> : 행렬의 열을 나타내는 태그로 제목을 제외한 내용을 나타내는 태그.  
  table, th, td { border: 1px solid black ; } : 테이블 전체, 제목, 내용 모두에 테두리를 씌움  
  # head 태그 안의 style 태그에 선언.  
  border-collapse : collapse ; : 이중 실선을 단일선으로 바꿔준다. ( 테두리선이 겹치는 것을 제거.)  
  th, td { padding : 15px ; } : 테이블 셀의 안쪽 여백을 지정  
  th { text-align: left or center or light; } : 제목 부분 글의 정렬을 지정  
  table { border-spacing : 5px ; } : 테이블의 행과 열 서로 간의 간격을 설정  
  <th colspan="2">내용</th>: 두개의 열을 한 개로 합친다.  
  <th rowspan="2">내용</th>: 두개의 행을 한 개로 합친다.  
  <caption>내용</caption>: 표 위에 캡션을 다는 태그 ( table 태그안에 입력)  
  # 테이블이 여러 개일 경우 특정 테이블의 스타일을 바꾸기 위해서는 id 속성을 이용한다.  
  <table id="t01"> : 스타일을 바꿀 테이블의 이름을 설정  
  #t01 tr:nth-child(even) { background-color : ;} : t01 테이블의 tr 태그중 짝수번째 태그에만 해당  
  #t01 tr:nth-child(odd) {  background-color: #fff ; } : t01 테이블의 tr 태그중 홀수번째 태그에만 해당  
  #t01 th { background-color: black ; } : t01 테이블의 th 태그에만 해당
* **HTML 목록**   
  <ul style="list-style-type:모양 타입;"> : ul =순서번호가 없고 기호로 나열된 항목 (unordered list)  
    <li>내용</li> : 입력한 모양 타입에 따라 항목 앞에 모형을 지정  
    <li>내용</li> </ul> 🡪 ex) disc, circle, square, none   
  <ol type="순서번호 타입"> : ol =순서번호가 있는 항목 (ordered list)  
    <li>내용</li> :입력한 순서번호 타입에 따라 순서번호의 모양을 지정  
    <li>내용</li> </ol> 🡪 ex> 1, A, a, I, i  
  <dl>  
    <dt>제목</dt> : dt의 내용은 제목처럼 dd 의 내용은 들여쓰기해서 내용처럼 출력  
    <dd>내용</dd> </dl>  
  <ul>    
  <li>항목</li>  
    <li>항목  
      <ul>   
   <li>세부항목</li> </ul> : 항목내에 세부항목을 중첩으로 들여쓰기로 출력  
    </li>  
    <li>항목</li> </ul>  
  <ol start="50"> : 순서번호의 시작번호를 50으로 설정  
  <li>내용</li> </ol>
* **HTML Block 요소 : 새줄의 전체부분을 차지 (줄바꿈O)**  
  <div>내용</div> : 입력한 내용의 스타일링을 할 때 사용 ( 문단이나 문장 단위 )  
  🡪 div 요소 : 컨테이너의 역할을 수행 = 다른 HTML 요소들이 들어올 수 있음  
  ex) div, dl, dt, h1~h6, hr, li, ol, p, table, ul
* **HTML Inline 요소 : 새 줄에서가 아니라 그 줄에서 필요한 만큼만 차지 (줄바꿈 x)**  
  <span>내용</span> : 입력한 내용의 스타일링을 할 때 사용 ( 단어 단위)  
   🡪 span 요소 : 컨테이너의 역할을 수행 = 다른 HTML 요소들이 들어올 수 있음  
  # 추가적으로 style, class, id 등 을 함께 사용가능  
  ex) a, bdo, br, button, I, em, b, img, map, q, script, span, sub, sup, strong
* **Class 속성 : 비슷한 종류의 여러 개의 요소를 묶을 때 사용, 묶어서 스타일링 가능**  
  Class 의 이름표를 지정하고 헤드 태그에서 스타일링하기위해 불러올 때 앞에 . 을 붙여준다.  
  🡪 ex) .Class1 : Class1 이라는 이름으로 지정한 부분에 모두 해당  
  한 요소에 클래스를 여러 개 지정할 때는 클래스명 사이에 공백을 띄우면 된다.  
  서로 다른 태그여도 같은 클래스를 지정할 수 있다.
* **Id 속성 : 유일무이한 한 개의 속성만 묶을 때 사용**  
  Id 의 이름표를 헤드 태그에서 스타일링하기위해 불러올 때 앞에 # 을 붙여준다.  
  🡪 ex) #지정이름 : 지정해 놓은 이름이 해당하는 부분만 적용  
  # 클래스 속성과는 다르게 id 속성은 여러 개의 텍스트를 지정할 수 없다.
* **HTML Iframe : 하나의 웹페이지 안에서 또 다른 웹페이지를 나타낼 때 사용**<iframe src="URL" height="높이" width="넓이"></iframe>  
  : 입력한 URL, 사이즈에 맞는 영상이나 웹페이지가 삽입  
  <iframe src="URL" style="border:설정값;"></iframe>  
  : 삽입된 웹페이지의 테두리를 설정 ex) none, 2px solid red  
  <iframe src = “처음에 보여줄 URL” name =” 설정할 이름 “</iframe>  
  <p><a href = “ 이동할 URL “ target = “ 설정한 이름 “> 텍스트 </a> </p>  
  : <p> 태그로 인해 화면에 보여지는 텍스트를 클릭하면 iframe 의 화면의 링크가 바뀐다.
* **HTML 자바 스크립트**<script> function 함수이름() {  
  document.getElementById(“ 설정해 놓은 Id 속성 “).함수로서 사용할 기능;} </script>  
  : 함수를 사용하면 설정해 놓은 Id 속성이 있는 부분에 함수의 기능을 적용할 수 있다.  
  # 사용할 수 있는 기능 : innerHTML (텍스트를 변환), style (스타일 변환), src (링크의 변환)  
  <noscript> 태그 : 설정해 놓은 스크립트 기능을 사용할 수 없는 장치일 경우 대체 텍스트사용
* **HTML 파일 경로 : 루트 폴더가 기준이 된다.  
  절대파일 경로 : 파일의 전체 URL 이 경로가 된다.**<img src =” URL / image / 파일” : “파일”은 URL 웹사이트의 “image” 폴더안에 존재  
  # URL : http ( 프로토콜) , www (서버), w3schools.com (도메인)  
  <img src = “/image/파일”> : “파일”은 현재 웹의 루트(최상위)에 있는 “image” 폴더안에 존재  
   **상대파일 경로 : 현재 페이지를 기준**<img src = “파일”> : “파일” 은 현재 페이지와 같은 폴더에 존재  
  <img src = “image/파일”>: “파일”은 현재 폴더에 있는 “image” 폴더안에 존재  
  <img src = “../image/파일”> : “파일”은 이전 폴더에 있는 “image” 폴더안에 존재  
  <img src = “./파일”> : “파일”은 현재 폴더안에 존재 ( /은 폴더를 표현)
* **HTML 레이아웃 : 레이아웃을 설정할 때 table 을 만들 필요가 없다.**HTML은 웹 페이지의 다른 부분을 정의하는 몇 가지 의미 요소를 제공합니다.

|  |  |
| --- | --- |
| HTML5 의미 요소 | * <header>-문서 또는 섹션의 헤더를 정의합니다 * <nav>-탐색 링크를 위한 컨테이너를 정의합니다 * <section>-문서에서 섹션을 정의합니다 * <article>-독립적인 기사를 정의합니다 * <aside>-사이드 바와 같이 컨텐츠를 제외하고 정의합니다. * <footer>-문서 또는 섹션의 바닥 글을 정의합니다 * <details>-추가 세부 사항을 정의합니다 * <summary>- <details> 요소의 제목을 정의합니다 |

W3.CSS 에서 스타일 시트를 가져와서 레이아웃에 적용 시킬 수 있다.

* **HTML 반응형 웹 디자인 : 웹사이트의 크기를 장치에 따라서 적절하게 보여주는 웹   
  # 주로 viewport 를 사용**반응형 이미지 : max-width, max-height 를 사용 하여서 이미지의 크기 비율을 조절할 수있다.  
  ex) style = “width:100%” (x) => style = “ max-width:100%; height : auto;”  
  반응형 텍스트 : font-size : 10vw(viewport width) 를 이용하여 텍스트의 크기를 조절한다.  
  미디어 쿼리 : 조건을 적용해서 화면의 크기에 따라 쿼리를 다르게 출력  
  @media screen and (max-width:800px) { : 화면의 너비가 800px 보다 작다면  
  .left, .main, .right { width:100%; } : left, main, right 클래스를 너비 100% 로 보여준다.  
  # 스크린으로 보여지는 것과 프린트를 보여주는 것을 다르게 출력할 수 있다.
* **HTML 컴퓨터 코드**code 태그를 이용해서 변수에 값을 저장할 수 있다.  
  kbd 태그를 이용해서 키보드로 입력한 값을 표현할 수 있다.  
  samp 태그를 이용해서 컴퓨팅 시스템의 출력을 표현할 수 있다.  
  var 태그를 이용해서 변수에 값을 저장할 수 있다.
* **HTML 의미 요소 : 브라우저가 개발자 모두에게 의미를 전달하기 위해 사용**비의미적 요소 : <div>, <span> 은 내용에 대헤서 표현 x  
  시멘틱 요소 : <form>, <table>, <article> 은 내용에 대해서 표현 O  
  # div 태그에 저장해서 쓰지않고 각각의 태그를 이용해서 표현한다. Ex) header, nav, aside  
  <section> 요소 : 제목을 가지고 있는 주제별 컨텐츠 그룹을 나타낸다. (문단)  
  <article> 요소 : 웹사이트의 나머지 부분과는 구분이 되는 독립적인 컨텐츠를 표현  
  # section 과 article 은 서로를 포함하거나 중복해서 사용할 수 있다.  
  <header> 요소 : section , article 의 헤더( 제목 ) 을 표현한다.  
  <footer> 요소 : 바닥글 ( 꼬리말 ) 을 표현할 때 사용한다.  
  <nav> 요소 : 다른 웹 사이트로 이동할 수 있는 링크의 집합을 표현할 때 사용  
  <aside> 요소 : 문단의 주제와는 관련이 없지만 추가 설명을 해주기 위해 사용  
  <figure>, <figcaption> 요소 : 그림이나 표에 사용하는 캡션을 표현할 때 사용.  
  # 시멘틱 태그는 웹사이트를 적절하게 공유할 수 있도록 하기 위해서 사용한다.
* **HTML 스타일 가이드 및 코딩 규칙 : 다른 개발자들이 쉽게 이해할 수 있도록 규칙을 따른다  
  (1)** 문서 시작시에 항상 <!DOCTYPE html> 을 사용한다.  
  (2) 모든 태그를 사용할 때 소문자를 사용한다.  
  (3) 모든 태그를 사용할 때 항상 끝마침 태그를   
  (4) 이미지를 입력할 때 항상 너비, 높이를 지정하는게 좋다.  
  (5) 등호 주위에 공백을 사용해도 정상적으로 인식할 수 있다.  
  (6) 좌우 스크롤은 사용에 불편하므로 행을 입력할 때 너무 긴 값을 입력하는 것은 좋지않다.   
  (7) 목록이나 테이블을 형성하는 것이 아니라면 들여쓰기를 하지 않는게 좋다.  
  (8) 검색엔진에서 적절하게 검색되게 하기 위해서는 title 요소를 항상 입력해야 된다.  
  (9) html, head , body 태그를 생략없이 항상 입력  
  (10) 문서를 작성할 때 언어 및 문자 인코딩을 항상 설정해야 한다.  
  (11) 뷰 포트를 이용해서 장치의 크기에 따라서 이미지 및 텍스트를 적적하게 보여준다.  
  (12) 주석을 사용할 때 여러 줄에 걸친 내용은 들여쓰기를 이용해서 보기 쉽게 표현한다.  
  (13) 외부 스타일 시트, 자바 스크립트를 사용할 때는 간단한 링크 구문을 이용한다.  
  (14) 외부 링크나 이미지를 사용할 때에는 대소문자를 정확하게 입력해야 한다.  
  (15) 파일마다 확장자를 정확하게 입력해야 한다.
* **HTML Entities : <, > 의 부등호를 사용하려면 태그와 혼동되기 때문에 지정된 기호를 사용한다**< : &lt ; or &#60; 로 표현  
  > : &gt ; or &#62 ; 로 표현  
  non-breaking space : 내용이 길어져서 다음 문단으로 갈 때 단어가 분리되지 않도록 사용  
  # 실제 공백이 필요한 경우 &nbsp 를 사용해서 공백을 유지할 수 있다.
* **HTML 기호 : 내용으로 기호를 사용하고 싶을 때 사용 ( 기호, 단위 그리스문자 등등)**&euro; , &sum; 등등 다른 태그에 포함되지 않고 내용처럼 사용가능
* **HTML 이모티콘 : 이모티콘은 이미지 파일이 아니라 UTF-8 알파벳 문자이다.**<head> 태그 안에 <meta charset = “UTF-8”> 을 선언 후 &#해당숫자 로 사용가능  
  # 외부 스타일 시트를 가져와도 @charset “UTF-8”; 을 선언 후 사용
* **HTML 인코딩 (character sets) : 여러가지 문자세트가 있지만 UTF-8 은 거의 모든 문자를 포함**
* **HTML URL (Uniform Resource Locators)   
  :웹 주소는 IP주소로 표현하지만 사람들이 기억하기 쉽게 URL 로 표현을 한다.  
  URL 인코딩은 ASCII 문자로만 가능하며 다른 문자가 들어올 경우 변환하여 전송**https : www. W3schools. Com /html /default.asp   
  scheme : //prefix. Domain: port /path /filename  
  scheme : 인터넷 서비스 유형 (http , https)  
  prefix : 도메인의 접두사 ( www)  
  domain : 인터넷 도메인 이름 ( w3schools.com)  
  port : 호스트의 포트 번호 ( http 의 경우 기본값 80 )  
  path : 서버의 경로 ( 생략된 경우 사이트의 루트 값 )  
  filename : 문서 또는 파일의 이름  
  http ( HyperText Transfer Protocol ) : 일반적인 웹 페이지  
  https ( Secure HyperText Transfer Protocol ) : 안전한 웹 페이지
* **HTML XHTML ( eXtensible HyperText Markup Language ) => HTML을 다른 데이터 형식과 함께 사용할 수 있도록 확장성과 유연성을 높이기 위해 사용  
  : XML 애플리케이션으로 정의된 HTML 버전, HTML 보다 더 엄격한 규칙을 준수해야 하고 모든 주요 브라우저에서 지원된다.**# HTML 과의 차이점  
  xmlns 속성은 <html> 태그의 필수 속성이다.   
  <html> , <head> , <title> , <body> 는 필수 입력 요소이다.  
  태그는 항상 닫혀 있어야 하고 올바르게 중첩되어 있어야 한다. <br />  
  요소 및 속성은 항상 소문자여야 하고 속성 값은 항상 인용하여야 한다. 속성 최소화는 금지  
  # W3C Validator 로 HTML 문서가 규칙을 잘 준수했는지 검사할 수 있다.
* **CSS ( Cascading Style Sheets ) :** HTML 요소가 화면에 표시되는 방법을 설정  
  #여러 웹 페이지의 레이아웃을 한번에 제어할 수 있다. 외부 스타일 시트는 CSS파일에 저장  
  HTML 은 웹페이지를 포맷하기위한 태그가 없기 때문에 CSS 를 사용
* **CSS 구문  
   h1 { color : blue ; font-size : 12px ; }  
  selector** **property value property value**선택기 : 스타일을 적용할 대상 태그의 이름  
  선언문 : property : value ; 의 형태로 선언문과 선언문을 구분할 때는 ; 를 사용  
  property : Css 속성이름 , Value : 값 property : Value 의 형태로 사용
* **CSS 선택기 (선언문)**해당 태그의 이름을 입력하거나 id, class 속성의 이름을 입력.   
  웹 페이지의 모든 요소에 적용하려면 \* 을 사용한다.  
  id : #id의 이름 , class : .class의 이름 형태로 사용   
  # p.class의 이름 , p#id의 이름 : 형태로 사용하면 p 태그 중에 해당되는 태그에만 설정  
  여러 개의 태그에 대한 정보를 동시에 설정할 수 있다. 태그사이에는 , 로 구분
* **CSS 를 추가하는 방법 : 외부 Css, 내부 Css, 인라인 Css 세가지의 방법이 있다.**외부 CSS : 미리 만들어 놓은 외부 CSS 파일을 가져와서 적용 (style 태그 사용하지 않음)  
  <link rel = “stylesheet” type =”text/css” href = “ 파일이름 “ > 의 형식으로 사용  
  # 외부 CSS 파일은 HTML 태그가 없이 중괄호로만 입력한다.  
  내부 CSS : <head> 안에 <style> 태그를 이용해서 각각의 스타일을 적용  
  인라인 CSS : 적용하려는 태그안에 style 속성을 입력해서 해당태그에만 적용  
  **#여러 개의 스타일 타입을 적용시켰을 때는 인라인 스타일 속성이 가장 먼저 적용되고   
  그 다음에는 head 태그 중에서 가장 밑에 있는 스타일 속성이 적용된다.**
* **DOM (the Document Object Model) : HTML, XML 문서의 프로그래밍 인터페이스이다.  
  인터페이스 : 두 개체를 서로 연결해주는 중간 매개체 역할**=> 문서의 구조화된 표현을 제공하며 프로그래밍 언어가 DOM 구조에 접근할 수 있는 방법을 제공하여 문서구조 , 스타일, 내용 등을 변경할 수 있게 돕는다.  
  웹서버에 저장된 HTML 문서를 정상적으로 출력하기 위해서는 클라이언트로 로딩하는 순간 웹브라우저가 HTML 문서를 읽고 DOM 을 구조적으로 생성한다. => JavaScript 가 정상 작동  
    
    
    
    
    
    
    
    
    
  웹서버에 저장된 HTML 문서를 정상적으로 출력하기 위해서는 클라이언트가 HTML 문서를 로드하는 순간 문서를 읽고 구체적으로 구조화된 DOM 을 생성한다.  
  그때 DOM 은 html 태그를 root 노드로 하는  
  노드 트리 구조로 생성한다.

[ DOM ]

* **CSS 조합 : 선택기 간의 관계를 지정할 때 사용**공백문자 (후손 선택기) : A B => A의 후손 노드인 B를 선택한다. (포함하고 있는 모든 요소)  
  ex) div p { : div 요소 내의 모든 p 요소를 선택  
  자식 선택기 : A> B => A의 자식 노드인 B를 선택한다. (포함하고 있는 바로 밑 요소)  
  인접한 형제 선택기 : A + B => A의 바로 밑에 있는 형제인 B를 선택한다.  
  => 함께 포함되어 있는 요소 중 가장 인접한 요소  
  일반 형제 선택기 : A ~ B => A의 형제인 B를 선택한다.  
  => 함께 포함되어 있는 모든 요소
* **CSS pseudo class ( 의사 클래스 ) : 요소의 상태를 표현해줄 때 사용한다.**ex) 마우스를 가져갔을 때 , 한번 방문 했을 때, 초점을 맞췄을 때   
  형식 : selector : pseudo-class { property : value; }  
  a 태그의 의사 클래스  
  **=> a : link : 링크를 방문하지 않은 상태   
  => a : visited : 링크를 방문한 상태  
  => a : hover : 링크위에 마우스를 올렸을 때의 상태  
  => a : active : 링크를 클릭 했을 때의 상태**  
  # 의사클래스와 CSS 클래스를 함께 사용했을 때   
  => a . highlight : hover : a 태그 중에서 highlight 태그인 대상만 선택  
  # 모든 HTML 요소에서 의사 클래스를 사용할 수 있다.  
  # div : hover p : div 요소 위에 마우스를 올렸을 때 p 태그를 선택한다.  
    
  CSS 첫번째 자식 클래스 : first-child 속성을 사용  
  ex) p:first-child : 첫번째 자식인 p 요소를 선택한다.  
  ex) p:last-child : 마지막 자식인 p 요소를 선택한다.  
  ex) p i : fist-child : p의 후손노드인 I 중 첫번째를 선택한다.  
  ex) p : first-child I : 첫번째 p 노드의 후손인 I 모두를 선택한다.  
  lang 의사 클래스  
  q : lang(no) { quotes : “~” “~” ; }   
  < q lang = “no”>내용 </q> : 내용 앞 뒤로 ~ ~ 를 출력한다.
* **CSS 의사 요소 : 요소의 첫 글자 또는 첫 줄, 요소 내용 전후를 선택할 때 사용**형식 : selector : : pseudo – element { property : value ; }  
  ex) p :: first-line : p 요소의 첫번째 줄을 선택한다.  
  # font – variant : small – caps; : 글꼴을 대문자 또는 소문자로 바꿔서 표현 ( 작은 대문자로)  
  **대문자 or 소문자 작은 대문자**ex) p :: first-letter : p 요소의 첫번째 글자를 선택한다.  
  ex) p.intro :: first – letter : p 요소의 intro 라는 이름을 가진 요소의 첫번째 글자를 선택한다.  
  # 의사 요소 여러 개를 중복해서 사용하게 되면 중첩되서 적용된다.  
  ex) h1 :: before : h1 요소의 앞을 선택한다. ( 다른 컨텐츠를 추가할 때 사용 )   
  ex) h1 :: after : h1 요소의 뒤를 선택한다. ( 다른 컨텐츠를 추가할 때 사용 )  
  ex) :: selection : 마우스로 선택한 영역에 속성을 적용한다.  
  # :: selection 은 color, background, cursor, outline 에 적용할 수 있다.
* **CSS 속성 선택기 : 특정 속성을 가지고 있는 요소를 선택한다.  
  [attribute] 의 형식으로 사용한다.**ex) a [ target ] : a 태그 중 target 속성을 가진 태그만 선택한다.  
  ex) a [ target = “\_blank”] : a 태그 중 target 속성의 값이 \_blank 인 태그만 선택한다.  
  ex) [ title ~= “flower” ] : “flower” 라는 텍스트를 포함하고 있는 title 요소만을 선택한다.  
  ex) [ class |= “top” ] : “top” 라는 텍스트로 시작하는 요소만을 선택한다.   
  # top 으로 시작하는데 단독으로 써 있거나 – 으로 구분이 되어 있는 경우에만 선택된다.  
  ex) [ class ^= “top” ] : 형식에 상관없이 top 으로 시작하는 요소를 모두 선택한다.  
  ex) [ class $= “test” ] : 형식에 상관없이 test 로 끝나는 요소를 모두 선택한다.  
  ex) [ class \*= “te” ] : 형식에 상관없이 te 라는 값을 포함하면 모두 선택된다.  
  ex) input [ type = “text” ] : input 의 타입이 text 일 경우 선택한다.
* **CSS 주석 : 코드에 영향을 주지는 않지만 다른 개발자에게 해당 코드를 설명하기 위해 사용**/\* 내용 \*/ 의 형태로 주석을 사용
* **CSS 색상 : RGB, HEX, HSL, RGBA, HSLA 를 이용하여 색상을 표현한다.**
* **CSS 배경색상 : 색상과 같은 형태로 생삭을 표현**opacity 속성 : 투명도를 조절하는 속성으로 0~ 1 의 값을 갖는다
* **CSS 배경 이미지 : 화면 배경을 이미지로 채울 수 있다.**background-image : url(“ 이미지 파일 이름 “); : 이미지 파일의 이름을 입력하면 배경이 출력  
  background-repeat : repeat – 방향 : 배경 이미지가 반복되는 방향을 정할 수 있다.( x, y , xy)  
  **# 반복을 없애고 싶을 때는 방향 값에 no-repeat 를 입력한다.**background-position : right top : 배경 이미지의 위치를 조절할 수 있다.
* **CSS 배경 attachment : 배경 이미지를 스크롤 할지 고정할지를 선택**background-attachment : fixed : 배경 이미지를 스크롤에 상관없이 고정한다.  
  background-attachment : scroll : 배경 이미지를 스크롤에 따라서 움직이게 한다.
* **CSS 배경 단축표현 : 코드를 단축하기 위해서 하나의 속성에 값을 입력한다.**background : #ffffff url(“ 이미지파일 이름”) no-repeat right-top) : 공백을 이용해서 구분한다.
* **CSS 테두리 : border 속성을 사용해서 테두리의 스타일을 설정할 수 있다.  
  (중요) # 1개의 값을 입력할 때 모든 방향에 적용된다.  
  # 2개의 값을 입력할 때 (위, 아래쪽) 순서로 적용된다.  
  # 3개의 값을 입력할 때 (위, 측면, 아래쪽) 순서로 적용된다.  
  # 4개의 값을 입력할 때 (위, 오른쪽, 아래쪽, 왼쪽) 순서로 적용된다.**dotted : 점선 테두리 dashed : 네모 점선 테두리 solid : 실선  
  double : 이중 실선 none : 테두리가 없음 hidden : 숨겨진 테두리  
  border-style : 원하는 방향의 테두리에 대한 스타일을 설정할 수 있다.  
  border-width : 테두리의 두께를 조절할 수 있다.  
  # 값을 입력하거나 thick (얇음) 등 텍스트를 입력해서 설정한다.  
  border-color : 테두리의 색상을 설정할 수 있다.  
  border-면방향-style : 입력한 면의 방향의 스타일을 설정할 수 있다.
* **CSS 테두리 단축표현 : 코드를 단축하기 위해서 하나의 속성에 값을 입력한다.**border : 5px solid red : 공백을 이용해서 구분한다.  
  border-left : 왼쪽 테두리만을 지정할 수 있다.
* **CSS 둥근 테두리 : border-radius 속성을 이용해서 둥근 테두리를 설정할 수 있다.  
  # border-radius : 5px : 5px 만큼 사각형을 깎아서 둥글게 만든다.**
* **CSS 바깥쪽 여백 (margin) : 외부 여백 공간의 스타일을 설정할 수 있다.  
  # 테두리와 마찬가지로 4방향 여백을 각각 설정할 수 있다.**
* **CSS 여백 축소 : 상단과 하단이 서로 겹칠때에만 적용할 수 있다. ( 좌우 x)**# 겹치는 두 값 중 큰 값만 적용시킨다.  
  margin : 20px 0 0 0; : 상단에만 여백을 둔다
* **CSS 안쪽 여백 (padding) : 내부 여백 공간의 스타일을 설정할 수 있다.  
  # 테두리와 마찬가지로 4방향 여백을 각각 설정할 수 있다.**width 속성과 padding 속성을 함께 사용하면 측면 여백과 너비를 더한 값이 총 크기이다.
* **CSS 높이와 너비 : height , width 속성을 이용하여 크기를 설정한다.  
  # 실제 크기는 여백을 제외한 값이다.**최대 너비 설정 : max-width 를 사용하면 장치의 크기가 바뀔 때 적용되는 최대크기를 설정.
* **CSS 박스 모델 : 모든 HTML 요소는 상자로 간주해서 스타일과 레이아웃을 설정한다.  
  # 구성요소 : content(내용), padding(내부여백), border(테두리), margin(외부여백)**
* **CSS Text : 색상 크기 정렬 등 텍스트의 스타일을 설정한다.  
  # text-align : justify : 영어 입력일 경우 해당 범위를 넘어가면 줄을 바꿔서 출력  
  # text-decoration   
  => none (링크일 경우 밑줄 지움) , overline( 윗줄), underline ( 밑줄), line-through ( 취소선)  
  # text-transform : 대소문자 변환  
  => uppercase (대문자) , lowercase (소문자), capitalize (첫 글자만 대문자)  
  # white-space : 공백문자의 표현  
  => nowrap (공간이 부족해도 줄이 바뀌지 않음) wrap ( 공간이 부족하면 줄이 바뀜)  
  # text-shadow : 텍스트의 그림자 효과를 설정한다.  
  => text-shadow : 가로 세로 흐림효과 색상**color : 색상 background-color : 배경색상 text-align : 정렬 direction : 방향 (rtl)  
  vertical-align : 수직 정렬 text-decoration : 장식 text-transform : 변환  
  text-indent : 첫 줄 들여쓰기 letter-spacing : 글자 간의 간격 line-height : 줄 간격  
  word-spacing : 단어와 단어 사이의 간격 white-space : 공백문자 text-shadow : 그림자효과
* **CSS 폰트 : font-family 속성을 사용하여 텍스트의 글씨체를 설정  
  sans-serif , serif, monospace 로 구분한다.**font-family : “첫번째 글씨체”, 두번째 글씨체, 세번째 글씨체 : 앞에부터 없으면 순서대로 진행  
  font-style : normal(보통) , italic(이탤릭체), oblique(기울인글꼴), bold(굵은글씨)  
  font-size : em ( 배수 ) , % 가급적 상대적 크기로 사용
* **CSS 폰트 단축  
  font : 20px Arial, sans-serif; : 20px Arial 체가 없으면 sans-serif 체를 사용**
* **HTML Form : 사용자의 입력을 수집하고 서버로 보내고(submit) 다시 받을 때 사용**<form> 태그 : 텍스트 필드, 확인란, 제출버튼 등 양식요소를 포함한다.  
  **입력요소**   
  **<input> 요소** => name = value 의 값이 서버로 전송된다.  
  : id(이름표) , name(서버에게 전달해주는 변수) , value(화면에 보이는 값) 속성을 함께 사용  
  # name 속성 : 생략된 경우 서버에 데이터를 전송할 수 없다.  
  <input type = “text”> : 텍스트를 입력 받을 수 있는 상자가 만들어진다. (<label> 태그와 사용)  
  <input type = “radio”> : 많은 선택 중 하나만을 선택하기 위한 라디오 버튼이 만들어진다.  
  <input type = “submit”> : 제출 단추가 만들어 진다. 폼 핸들러로 데이터 양식 전송  
  # 다른 타입과 다르게 <input> 태그와 value 속성을 한번 더 사용하고  
  # <form action =”데이터를 전송할 서버의 페이지 명”> 을 사용  
  <label> 요소 : Input 상자 바로 위에 입력요소의 정보를 표현하기 위해서 사용 (캡션과 유사)  
  => 텍스트 상자를 클릭하기 어려울 때 label 부분을 누르면 텍스트 상자가 선택된다.  
  # for 속성을 함께 사용한다. For 속성은 id 태그 값을 읽어온다.  
  동작속성 action : 데이터를 서버로 보낼 때 데이터를 받는 서버를 정의할 때 사용  
  대상속성 target : 제출된 결과가 새 탭 또는 현재 탭에서 사용할지 정의할 때 사용  
  => \_self : 현재 탭에서 출력 , \_blank : 새 탭에서 출력  
  method 속성 : 데이터를 제출한 서버 주소필드에 입력한 데이터 값을 표현할 때 사용  
  => get : 입력했던 사용자의 입력 정보들을 함께 출력, post : 웹 사이트의 URL 만 출력  
  # 서버에 제출할 때 기본 method 속성 값은 get 이다.  
  # 보안에 민감하지 않고 북마크를 남길 데이터를 전송할 때는 get 방식을 사용  
  # password 등 민감한 데이터를 전송할 때는 post 방식을 사용 (북마크 불가능)  
  **<select> 요소 : 목록상자가 나와서 그중 한 개를 선택 할 수있다.**<option> 태그와 함께 사용 : 목록상자의 값을 지정할 때 사용  
  # selected 속성을 사용하면 처음에 보여지는 값을 설정할 수 있다.  
  # size 속성을 사용하여 보여지는 목록의 개수를 설정할 수 있다.  
  # multiple 속성을 사용하면 중복 선택을 할 수 있다.  
  <textarea> 요소 : 여러 줄의 텍스트를 입력할 때 사용  
  <button> 요소 => onclick=”alert ( 경고창), prompt (텍스트입력창), confirm(확인취소 버튼)  
  <fieldset> 요소 : 폼의 테두리를 생성해서 그룹화 해주는 역할  
  <legend> 요소 : 그룹화 했을 때 제목을 생성  
  <datalist> 요소 : 목록상자와 텍스트 입력 모두 사용가능  
  <output> 요소 : 식을 설정하고 input 태그를 이용해서 수식을 작성할 수 있다.
* **HTML JavaScript 사용 예  
  HTML 그래픽 : 웹페이지에 그래픽을 그릴 때 사용**<canvas> 요소 : 캔버스 상자에 경로, 상자, 원, 텍스트를 그려서 이미지로 추가한다.  
  <svg> 요소 : 벡터 기반으로 그림을 그리고 XML 형식으로 정의한다. (그림이 깨지지 않음)

**HTML 미디어 : 소리, 음악, 비디오, 영화, 애니메이션 등을 나타낼 때 사용**

1. The Internet :  
   전 세계의 네트워크를 상호 연결한 통신망  
   TCP/IP 프로토콜(통신규약)에 따라 정보 전송   
   다양한 인터넷서비스 제공 : 이메일, 파일전송, www,…
2. WWW (World Wide Web)  
   인터넷에서 웹문서를 전달하기 위한 서비스  
   HTTP 프로토콜 이용 HTML 문서를 교환 : 하이퍼링크, URL,…..  
   클라이언트 : 정보(HTML문서) 요청  
   서버 : 미리 정보를 저장, 요청에 응답 (실제로 일하는 곳)
3. 플러그인  
   웹 브라우저의 기능을 확장하는 프로그램  
   대표적인 플러그인 : 어도비 플래시, 마이크로소프트의 ActiveX(금융서비스) ,C++ 등  
   웹 표준인 자바스크립트의 중요성이 커지고 있음
4. 클라이언트 (사용자, 고객)  
   클라이언트 S/W는 PC의 크롬 등 웹 브라우저를 사용  
   1. http 프로토콜을 이용하여 고객은 원하는 웹페이지(URL)를 요청   
   2. 웹서버로부터 전달된 웹페이지를 읽고 해석하여 화면에 보여줌  
   3. 웹페이지는 html, css, javascript 언어로 작성됨
5. 서버 (일꾼)  
   웹서버 S/W는 서버 컴퓨터의 아파치, IIS 등 소프트웨어를 사용  
   1. http 프로토콜을 이용하여 고객이 요청한 웹페이지를 전달함  
   2. 일반적으로 고객의 요청에 따라 정보를 처리, 가공할 필요가있음  
   3. 이 경우 자바, c#, JSP, PHP, ASP, Node.js 등의 언어를 이용하여 서버에서 정보를 처리  
   4. 클라이언트에게 전달할 웹페이지(html, css, javascript)를 생성하고 요청한 웹페이지를 전달함