

html 문법 구조

```
<!doctype html><!-- DTD 선언 -- >
<html language="ko">
  <head>
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <title>머리 글(제목) </title>
    <style>
      /* 웹 구조 꾸미는 영역 */
    </style>
    <script>
      // 웹 동적반응 작업 영역
    </script>
  </head>
  <body>
    <!-- 웹 구조내용 작업영역 -- >
  </body>
</html>
```

일반적으로 웹문서를 표기하면 3개의 언어로 표현이 가능합니다.

css, javascript, html 3개의 언어는 하나의 웹문서안에 모두 표기가 가능하며, 확장자명은 *.html 입니다.

css는 style이라고 하며, <style> </style>내부에 작성합니다(상황에 따라 예외 존재).

javascript는 script라고 하며 <script> </script>내부에 작성하도록 되어있습니다(일부 예외).

html은 html 문서 자체도 의미하지만 <body></body>내부에 작성되는 부분을 지칭하고 있습니다.

double tag : <code> </code> - 시작과 끝이 존재하며 일정 부분(영역)을 선 할 때 사용하는 기능

single tag : <code/> - 시작과 동시에 끝이 나는 부분, 줄 바꿈, 영역 구분, 이미지 삽입 등의 기능 시에 사용
코드의 끝 부분에 닫기 전에 띄어쓰기와 함께 " / "기호를 붙인다.(시작 부분에 붙이면 동작하지 않음)

<!doctype html> 으로 작성된 부분은 DTD라 불리우며, xhtml, html4, html5 등 html의 형식을 나타냅니다(위코드는 html5).

웹문서(html)는 <html>로 시작하여 </html>로 끝나며 해당 내용(코드) 내부에 작성하도록 되어있습니다.

<html> 부분의 language="ko" 부분은 작성하지 않아도 무방하나 한국어를 기입할경우 작성하는게, 편의성이 있습니다.

크게 head와 body로 구분하며, html코드는 body에 작성하며, 이외의 다양한 기능들은 head 안에 작성하도록 되어있습니다.

<meta> 태그는 사용자의 정보 및 속성에 관하여 기입하는 내용으로 아래와 같은 내용들을 담고있습니다.

- <meta name="subject" content="주 제" />
- <meta name="title" content="제목" />
- <meta name="author" content="작성자" />
- <meta name="keywords" content="키워드1, 키워드2, 키워드3" />
- <meta charset="utf-8" />
- <meta http-equiv="X-UA-Compatible" content="IE=edge" />
- <meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">

위 내용중파란색으로 표기되어있는 부분은 꼭작성하는 것이 좋습니다.

utf-8로 표시되어있는 내용은 언어의 인코딩형식을 utf-8방식으로 처리하겠다는 의미입니다

(웹의 경우는 **euc-kr/utf-8** 두가지 형식 존재)

x-ua... 부분은 브라우저에서 사용하는 버전을 최신으로 작동하겠다는 의미이며, 위내용에서는 익스플로러를 처리해 놓았습니다.

주황색으로 표기되어있는 부분은 반응형웹의 경우 처리되는 부분으로 브라우저의 표현되는 부분과, 각종 기기(디바이스)크기를 1:1로 처리하겠다는 의미를 가집니다.

<style></style> 사이에 작성하는 코드는 **css**(body부분의 코드를 꾸며주는 영역) 코드입니다.

<script></script> 사이에 작성하는 코드는 **javascript**(body부분의 내용을 동적인 형태 및 다양한 기능을 처리하는 영역) 코드입니다.

<body></body> 사이에 작성하는 코드는 **html** 본연의 코드를 작성하는 부분입니다.

각 영역내의 회색으로 표기되어있는 부분은 **주석**으로 실제 웹에서 구현되지 않고, 참조하게 되는 표기방법으로 기억하시면 됩니다.

즉 <style>내에서는 /* */ 로 표기하여사이에 작성하면 웹에서 구동되지 않습니다(여러줄 처리가능).

<script>내에서는 /* */ 와 // 두가지 방법으로 /* */는 <style>과 마찬가지로 여러줄 주석이며, //는 한줄로만 처리되는 주석입니다.

<body>내에서는 <!-- --> 로 표기되는 주석처리이며, <style>, <script>영역내에서는 주석으로 인지되지 않습니다(여러줄 처리가능).

html/body

body내의 코드(태그)는 기본 2가지 형식으로 구분되어 있습니다.

외운다는 것보다 자주보고 익혀주세요.

인라인 태그 가로, 세로 등의 크기 값이 존재하지 않음, 여러개의 인라인태그가 모여있을경우, 가로로 정렬(줄바꿈이 없음)
예외: img(가로, 세로값이 존재하나, 인라인 태그로 되어져 있음)

블록 태그 가로, 세로 값이 존재하며, 세로로 배치되는 형태(줄바꿈이 있음)

| 인라인 태그종류 | 블록 태그 종류1 | 블록 태그 종류2 |
|---|--|--|
| <ul style="list-style-type: none"> • a : 앵커태그(링크) • br : 문단개행(줄바꿈) • span : 인라인형태 그룹핑 엘리먼트 • img : 이미지 삽입(단, 크기존재) • font : 폰트 외양 • i : 이탤릭체 • em : 강조(이탤릭체) • strong : 굵은강조 • input : 입력필드(단, 크기존재) • label : 폼 태그 라벨(단, 크기존재) • legend : 필드셋 제목 • select : 선택가능목록 • q : 한 문장내 짧은 인용문 (" 표기) • cite : 짧은인용 • abbr : 줄임말 • address : 물리적인 주소 • summary : 테이블 요약정보 • td : 테이블 데이터 • iframe : 인라인 하위 윈도우 | <ul style="list-style-type: none"> • body : 페이지 본문 • div : 블록레벨 그룹핑 엘리먼트 • h1~h6 : 머리글 • p : 단락 • ul : 순서가 없는 목록 • ol : 순서가 있는 목록 • li : 리스트 아이템 • dl : 정의목록 • dt : 정의용어 • dd : 설명정의 • pre : 미리 서식을 지정한 텍스트 • blockquote : 블록레벨 인용문 사용 • hr : 가로줄 | <ul style="list-style-type: none"> • form : 입력 폼 • fieldset : 필드셋 라벨 • table : 테이블 • caption : 테이블 설명 • textarea : 텍스트 입력 폼 • th : 테이블 헤더 • tr : 테이블 행 • thead : 테이블 헤더 • tbody : 테이블 바디 • tfoot : 테이블 푸터 • col : 테이블 컬럼 • colgroup : 테이블 컬럼 그룹 • embed : 외부컨텐츠 • object : 개체임베딩 |

기초코드이해1

h1

- h1, h2, h3, h4, h5, h6 : 총 6가지가 존재하며 제목 태그라 불리며, 제목으로 사용함
- h1의 경우 각 문서, 또는 챕터 당 1개의 h1만 존재가 가능하며, 기입 하는 것이 좋음..

ul, ol, li, dl, dt, dd

- list 사용 시에 쓰이는 태그
- ul : 순서가 없는 태그 type=" " disc, auqre, circle
- ol : 순서가 있는 태그 type=" " i, l, a, A, 1
- li : ol, ul 에 속해져 있는 태그
- dl : 제목이 있는 태그
- dt : 리스트 제목 영역
- dd : 리스트 내용 영역

div

- 영역을 지정하는데 사용하는 태그(크기 값을 가지며, 일종의 상자의 개념)

p, span

- 글씨 및 내용을 삽입 할 때 사용하는 태그
- p : 가로, 세로 사이즈 값이 존재하며, 줄 바꿈이 있음(block tag)_문단
- span : 가로, 세로 등의 사이즈가 없음, 줄 바꿈이 되지 않음(inline tag)_부분
- * span tag 안에 p tag가 들어갈 수 없음, p tag 안에는 span tag가 들어갈 수 있음

b, i, u, s, strike

- 강조를 표현하는 태그 접근성으로는 사용할 수 없으며, 해당 내용을 다른 태그로 대체 됨
- b : 굵기 (**strong** 으로 대체)
- i : 기울기 (**em** 으로 대체)
- u : 밑줄 (**ins** 으로 대체)
- strike/s : 취소선 (**del** 로 대체)
- small : 작은 글씨

br, hr

- br : single tag로써
로 표기하며, 문단내에서 줄바꿈을 사용할 때 쓰임
- hr : single tag로써 <hr/>로 표기하며, 여러태그 작성중 화제의 전환이 될 때 쓰임

a

- 링크를 사용하는 태그
- 링크 기능 : #자리에 원하는 위치를 지정하여 사용

img

- 이미지 삽입 기능
-

id, class 이름설정하기

- 이름을 지정하여 지정된 이름에 대한 효과 및 꾸며줄 때 사용함
- id 값의 경우 하나의 이름만 존재가 가능
- 하나의 이름 및 여러 개의 공통된 이름으로도 사용 가능
- 숫자가 앞에 가거나, 띄어쓰기를 사용할 수 없음, 특수 문자 영어 외의 언어, 태그와 동일한 이름 사용할 수 없음.
- 이름이 길 경우 단어 사이에 "_" 를 기입하거나, 단어의 "첫"글자를 대문자로 표기
- 어떠한 태그에도 이름지정이 가능

```

<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>site1</title>
</head>
<body>
  <div id="wrap">
    <div id="header">
      <h1><a href="#">logo</a></h1>
      <div id="nav">
        <h2>menu_list</h2>
        <ul class="gnb">
          <li><a href="#">menu_01</a></li>
          <li><a href="#">menu_02</a></li>
          <li><a href="#">menu_03</a></li>
          <li><a href="#">menu_04</a></li>
          <li><a href="#">menu_05</a></li>
        </ul><!--.gnb-->
      </div><!--#nav-->
    </div><!--#header-->
    <div id="container">
      <h2 class="con_h2">본문</h2>
      <p class="con"> 본문 내용을 담는 내용을 처리하는 부분
      <span> 내용 중 부분 정리하는 경우에는 span 으로 묶어주며, <strong>강조1</strong>, <em>강조2</em>, <ins>강조3</ins> </span>
      </p>
      <p class="con"> 단락을 구분할때 별도로 사용 줄바꿈시에는 <br /> br로 구분하여 처리 </p>
      <p> 내용을 여러형태로 구분하여 처리가능 </p>
    </div><!--#container-->
    <div id="footer">
      <h2 class="footer_h2">정보를 담는 말미</h2>
      <ol class="footer_nav">
        <li><a href="#">회사정보</a></li>
        <li><a href="#">회사위치</a></li>
        <li><a href="#">고객센터</a></li>
      </ol>
    </div>
  </div><!--#wrap-->
</body>
</html>

```

▲ 코드 작성

logo

menu_list

- menu_01
- menu_02
- menu_03
- menu_04
- menu_05

본문

본문 내용을 담는 내용을 처리하는 부분 내용 중 부분 정리하는 경우에는 span 으로 묶어주며, 강조1, 강조2, 강조3

단락을 구분할때 별도로 사용 줄바꿈시에는
br로 구분하여 처리

내용을 여러형태로 구분하여 처리가능

정보를 담는 말미

1. 회사정보
2. 회사위치
3. 고객센터

▲ chrome

logo

menu_list

- menu_01
- menu_02
- menu_03
- menu_04
- menu_05

본문

본문 내용을 담는 내용을 처리하는 부분 내용 중 부분 정리하는 경우에는 span 으로 묶어주며, 강조1, 강조2, 강조3

단락을 구분할때 별도로 사용 줄바꿈시에는
br로 구분하여 처리

내용을 여러형태로 구분하여 처리가능

정보를 담는 말미

1. 회사정보
2. 회사위치
3. 고객센터

▲ firefox

◀ IE(explorer)

기초코드이해2

Table

- 테이블 태그(Table): 표를 표현하거나 제작함에 있어 필요한 태그, 과거 웹페이지 제작형태로 사용되었으나 현재는 표의 기능을 하는곳에 사용되어지고 있습니다.
ex: 주식차트, 경기운영표, 게시판 등

| | |
|-------------------------------|--|
| 테이블을제작함에 있어 설명하는곳은 summary"> | |
| <caption> table 제목 </caption> | |
| <tr> | |
| <th>Month</th> | |
| <th>Savings</th> | |
| </tr> | |
| <tr> | |
| <td>January</td> | |
| <td>\$100</td> | |
| </tr> | |
| </table> | |

- 가로값의 영역먼저 표기
- 가로영역 값 내에 세로 영역값 세팅
- 작성의 요령 및 내용이라고 본다면 table 전체를 선언(<table></table>)하고, 제목/의미(summary="표설명" <caption>표제목작성</caption>)를 부여, 행(<tr></tr> 갯수만큼 만들기)인지 선언, 내부에 셀의 갯수(<td></td>)를 표기 순으로 진행(셀의 형식이 제목일 경우에는 <th></th>)합니다.

```
<table> </table>
summary=" "
<caption> </caption>
<colgroup></colgroup>
<col>
<thead></thead>
<tbody></tbody>
<tfoot></tfoot>
<tr> </tr>
<td> </td>
<th> </th>
```

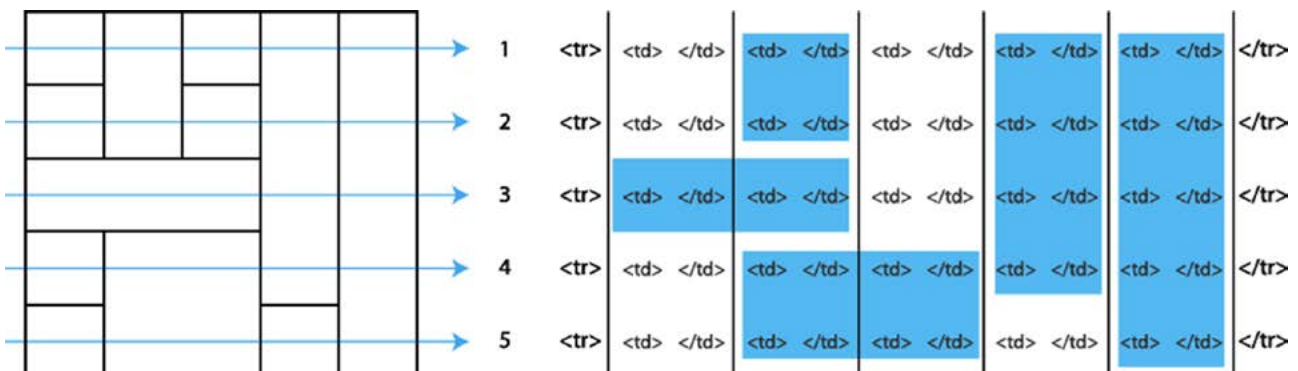
테이블을 사용하기 위해서는 table형식의 구조를 선언한다.
 실제로 보이지는 않으나, 설명을 넣어 이해할 수 있도록 유도(웹 접근성위해 사용)
 표에대한 제목
 열단위로 그룹으로 묶어 사용하는 기능
 열
 표에서 상단영역(head영역)을 설정할때(표에서 마지막부분에 작업하여도 결과물은 최 상단에 배치)
 표에서 상단영역(body영역)을 설정할때(표에서 어느영역에작업하여도 head와 foot영역사이에 배치)
 표에서 상단영역(footer영역)을 설정할때(표에서 첫부분에 작업하여도 결과물은 마지막영역에 배치)
 표작성시 행에 해당하는 영역으로, 표작업은 tr영역을 먼저 생성후 각 기입되는 셀영역입력
 표작성시 셀에 해당하는 영역으로, tr영역 내에서 각기 셀에 내용을 입력
 td영역(셀)영역 작성시 표의 주제(소제목) 영역에 해당하는곳을 th로 작성

```
<table>
  <caption> boxspan test </caption>
  <tr> <td></td> <td rowspan="2"> </td> <td></td> <td rowspan="4"> </td> <td rowspan="5"> </tr>
  <tr> <td></td> <td></td> <td></td> </tr>
  <tr> <td colspan="2"></td> <td></td> </tr>
  <tr> <td></td> <td colspan="2" rowspan="2"> </td> </tr>
  <tr> <td></td> <td></td> </tr>
</table>
```

```
colspan=" "
```

```
rowspan=" "
```

| |
|---|
| th or td영역에 입력하며, 셀과 셀사이를 합칠때 사용함(가로형식) |
| th or td영역에 입력하며, 셀과 셀사이를 합칠때 사용함(세로형식) |



* * colspan, rowspan의 경우 합쳐지는 곳에 존재하는 셀(td, th)영역은 삭제하여야 자리를 정확하게 잡을 수 있습니다.
* * 위 이미지는 style에서 "border-collapse: collapse;"라고 표기하면, 표의 간격이 서로 붙는 성향이 있습니다.

기초코드이해3

form

- 양방향 통신이 가능한 기능(로그인, 검색, 주문..)이라고 말합니다.
- 조금더 쉽게 표현하자면 로그인, 물건주문, 주문물건 확인 및 체크, 정보수정, 설문조사 등 일반적으로 홈페이지 내용을 보기만 하는 것이 아닌, 사용자/소비자의 의견을 입력하여 보내는 기능이 가능한 것을 말합니다.

```
<form action="http://....." method="get">
  <fieldset>
    <legend>필드1 제목 </legend>
    <!-- 폼에 관련된 내용을 삽입 -->
  </fieldset>
  </fieldset>
  <legend>필드2 제목 </legend>
  <!-- 폼에 관련된 내용을 삽입-->
  </fieldset>
</form>
```

- 아래는 기본 태그/속성에 관한 부분입니다.

| | |
|---------------|--|
| form | form 영역 진행하기 위한 형식 선언 |
| action | 서버와의 연동을 위해 경로지정 |
| method | form 태그 처리 방식 지정 약 4가지 방식이 있으며, 2가지 방식을 중심으로 파악(get, post) |
| method="get" | 오픈 되어진 폼 형식에 주로 사용하며, 가벼우며, 빠르다.(보안에 취약_검색 및 공개 자료) |
| method="post" | 보안 처리 하여 사용되어지며, 무겁고, 느리다.(물품 주문, 로그인 사용 시...) |
| fieldset | form 영역에서 여러개의 form 형식으로 구분되어질 때 사용하는 태그(기본 border 값이 적용된다.) |
| legen | fieldset영역 설정 시 해당 fieldset 영역에서 제목 및 분류 시 사용하는 태그 |
| input | 내용을 입력시에 사용하는 태그 다양한 방식이 존재하며, id, name, value 등이 필요하다. |

```
<input type="text" id="text_name" value="검색창이 만들어져 내용을 확인해보시오." size="10" maxlength="10"
name="text_box" />
```

```
<label for="text_name"> text </label>
```

| | |
|-----------------|--|
| type="text" | 검색처럼 내용을 확인면서 입력 할 수 있는 방식(1줄) |
| type="password" | 내용 입력시 텍스트 내용이 보이지 않게 처리하는 방식 |
| size="20" | 입력창의 가로길이를 20크기만큼 |
| maxlength="10" | 입력가능한 크기값(영문은 10글자, 한글은 5글자 /bit 의 개념) |

```
<input type="radio" name="check" checked="checked" id="radio01"/> <label for="radio01">첫번째</label>
<input type="radio" name="check" id="radio02"/> <label for="radio02">두번째</label>
<input type="radio" name="check" id="radio03"/> <label for="radio03">세번째</label>
```

| | |
|--------------------|--|
| type="radio" | 단, 하나만을 선택시 사용(주로 동의체크용), 취소 기능 없음 |
| type="checkbox" | 여러개의 다중 체크박스사용시 이용(주로 설문조사), 선택 및 취소 가능 |
| name="server_name" | 서버와 자료를 공유하기위해 동일한 조건일경우 동일한 name을 사용(주로, radio, checkbox) |
| <label>btn</label> | input id_name과 동일한 이름을 for문으로 정하면, 서로 연결되어 작동(웹접근성 용이) |
| for="radio01" | input id값과 동일한 이름을 사용하여, 버튼의 대용으로 사용 |
| checked="checked" | 체크 이전에 미리 체크 되어진 상태를 표기 |

```
<input type="submit" name="submit_btn" value="확인하겠습니다."/>
```

| | |
|---------------|------------------------------------|
| type="submit" | 확인버튼 |
| type="reset" | 취소버튼 |
| type="button" | 버튼 |
| type="image" | 이미지로 버튼 대체 src="경로" 기능 사용, |
| alt="submit" | type="image" 용 (이미지 확인버튼 생성시) 설정용. |
| value="내용입력" | 버튼 이름 |
| type="file" | 첨부파일 사용시 사용(첨부파일 버튼명은 변경이 불가능함) |

```
<textarea name="text_area" id="text_area" cols="50" rows="10">내용을 삽입해서 확인하면 되는용도. </textarea>
```

| | |
|------------------------|--|
| <textarea> </textarea> | 여러줄의 내용을 기입할 경우(type="text"와 유사) 주로 댓글 기능에 사용
<textarea>의 경우 약관과 같은 내용을 삽입할 경우 tab버튼 또는 space 버튼 모두 인식되므로,
내용을 확인후에 사용하여야한다. |
| cols="50" | 가로 상자의 크기값 |
| rows="10" | 세로의 크기값 |

```
<select name="korea">
  <option value="seoul"> 서울 </option>
  <option value="daejeon"> 대전 </option>
  <option value="daegu"> 대구 </option>
  <option value="busan"> 부산 </option>
  <option value="gwangju"> 광주 </option>
</select>
```

<select> 여러개의 사항 중에 하나를 선택 할 경우(전화 번호)
 <option> 여러개의 내용을 표기
 selected="selected" 기본 첫 사항을 표기하나, selected 내용이 기입 될 경우 선택된 사항이 보이게 설정

```
<select name="font" id="font">
  <optgroup label="color">
    <option value="red">Red</option>
    <option value="green">Green</option>
  </optgroup>
  <optgroup label="size">
    <option value="11px">11px</option>
    <option value="12px">12px</option>
  </optgroup>
</select>
```

optgroup

- 펼침 목록 상자(Select Box)의 항목(Option)들을 의미있게 그룹핑하는데 사용되는 요소<select>
 여러개의 사항 중에 하나를 선택 할 경우(전화 번호)
 <option> 여러개의 내용을 표기
 selected="selected" 기본 첫 사항을 표기하나, selected 내용이 기입 될 경우 선택된 사항이 보이게 설정

기초코드이해4

css(style)1 :

head 영역에 대부분 작성하여, body영역 내에 존재하는 구조를 꾸며(모양, 위치, 색상 등.)주는 역할을 한다.

```
<head>
  <title> .... </title>
  <style type="text/css">
    /* 내용삽입 */
  </style>
</head>
```

표기방법 :

```
<p id="id_name"> </p>
<p class="class_name"> </class>
<code> </code>
```

```
#id_name { }
.class_name { }
code { }
```

```
<div id="id_name"></div>
작성방법 :
```

div#id_name or #id_name { } *div와 id 값 같은 위치에 있을 경우 붙여 표기

```
<ul> <li> aaa </li> </ul>
<ul> <li><span> aaa </span></li> </ul>
<ul> <li> aaa </li> </ul>
<ul> <li><span> aaa </span></li> </ul>
```

```
ul li { }          * 자손 선택자라고 칭하며, ul 안의 li를 선택(띄어쓰기로 인식)
ul span { }        * 자손 선택자라고 칭하며, ul 안의 span를 선택(띄어쓰기로 인식)
ul > li { }        * 자식 선택자라고 칭하며, ul 안의 li를 선택(>로 인식, 바로 밑의 코드선택)
ul > span { }      * 자식 선택자일 경우, ul 자식은 span이 아닌 li이므로 선택불가능
                   ( ul > li > span {} 표기는 가능)
```

{ } 표기는 내부에 style 속성을 표기하도록 되어있습니다.

즉, 배경색: 파란색, 글씨색:노랑 등의 표기를 하도록 되어있습니다.

이때 {배경색: 파란색; 글씨색: 노랑;}의 형태로 지시뒤에는 : 표기, 답뒤에는 ; 표기를해줍니다. 가독성을 위해 띄어쓰기를취합니다.

ex: **code{ background-color: blue; color: yellow; }** 입니다.

```
<style> 영역에 작성 시
<body>
  <p id="p_id">내용</p>
</body>
```

```
<style>
  p#p_id{ background-color : red ; color : yellow ; width : 300px ; height : 50px ; font-size : 30px ; }
</style>
```

* 태그 및 속성 선택 {지시할 내용 : 시행할 내용; 지시할 내용 : 시행할 내용; 지시할 내용 : 시행할 내용; 지시할 내용 : 시행할 내용}

style 내용 :

크기 단위: px(pixel 절대 크기 값) pt(point 절대 크기 값) em(상대 크기값, 기본16px) %(상대 크기값, 기본16px)

색상 단위: #000000 or #000 (0~9,a~f =>16진수) , 색상 명

size		
width	가로	* 단위 반드시 표기
height	세로	* 단위 반드시 표기

background		
background-color:	배경 색	webcolor(#xxxxxx), 색상 명을 표기
background-image:	배경 이미지	url("경로/파일 명.파일 형식"); 표기
background-size:	배경 이미지 사이즈	배경 이미지 크기 비율 조정
background-repeat:	배경이미지 반복	repeat-x(가로 방향 반복), repeat-y(세로 방향 반복), no-repeat(반복 안함)
background:	배경에 관련 된 부분 일괄처리 및 color, image, repeat 영역에 대하여, 별도 표기 없이 사용 가능	

css(style)2 :

font

font-family	글자 서체 명	띄어쓰기가 있는 경우 " " 기호로 감쌀 것.
font-size	글자크기	기기의 크기에 대한 비율에 맞춰 제작되어지므로 "em" 단위를 사용.
color	글자색	webcolor(#xxxxxx), 색상 명을 표기
font-weight	글씨굵기	thin, normal, bold, bolder, 100~900 등 표기(일부 효과 없는 경우 있음)
font-style	글씨 기울기	normal, italic 에 대한 내용이 기본 형태.
line-height	행간	글 줄과 글 줄의 사이의 간격
letter-spacing	자간	글자와 글자 사이의 간격
- font	전체(font)	font: 기울기 굵기 크기/행간 서체명, 서체명, "서체 명";

align

vertical-align	위치	top, middle, bottom, baseline
text-align	글씨정렬	left, center, right
text-indent	들여쓰기	오른쪽 방향 +, 왼쪽방향 -

margin

-외부여백(개체간의 간격을 만들때 주로 사용/ *여러개의 사용시 더블마진이라는 문제점 발생)

margin-top: 10px;	상단 여백을 10px
margin-right: 10px;	우측 여백을 10px
margin-bottom: 10px;	하단 여백을 10px
margin-left: 10px;	좌측 여백을 10px

margin: 10px;	상,하,좌,우 모두 10px 여백
margin: 10px 20px;	상,하 10px 좌,우 20px 여백
margin: 10px 20px 30px;	상 10px 좌,우 20px 아래 30px 여백
margin: 10px 20px 30px 40px;	시계방향으로(상단부터) 10px, 20px, 30px, 40px
margin: 0 auto;	상,하 여백없음(0일경우 단위 없어도됨) 좌,우 auto(일반적으로 중앙)

padding

- 내부여백(개체 자체의 크기값은 변하지 않으며, 옷을 입듯이 감싸는 영역이 변함, 별도의 색은 지정 불가.)

padding-top: 10px;	상단 여백을 10px
padding-right: 10px;	우측 여백을 10px
padding-bottom: 10px;	아래 여백을 10px
padding-left: 10px;	왼쪽 여백을 10px

padding: 10px;	상,하,좌,우 모두 10px 여백
padding: 10px 20px;	상,하 10px 좌,우 20px 여백
padding: 10px 20px 30px;	상 10px 좌,우 20px 아래 30px 여백
padding: 0px 20px 30px 40px;	시계방향으로(상단부터) 10px, 20px, 30px, 40px

border

- 외곽선(주변영향, padding와 비슷한 효과를 보일 수 있음 별도의 색 지정가능.)

border-width:	선의 굵기(px, %, bold, thin, thick, medium)
border-color:	선의 색상(webcolor, 색상명, transparent) *transparent(투명)
border-style:	선의 모양(solid, dotted, dashed, double, ridge, groove, inset, outset, none)
border-collapse:	선의 간격(주로 표작성시 collapse를 사용)

border: 굵기 모양 색상;	*4면전체를 (순서는 크게 상관 없음) 순으로 한꺼번에 기입
border-top: 굵기 모양 색상;	상단의 외곽선표기
border-right: 굵기 모양 색상;	우측의 외곽선표기
border-bottom: 굵기 모양 색상;	아래의 외곽선표기
border-left: 굵기 모양 색상;	좌측의 외곽선표기
- border: 1px solid #fff;	border: 굵기 모양 색상;

outline

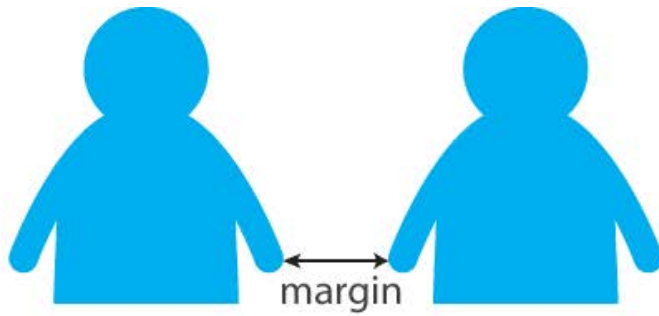
- 외곽선(주변영향없음)

outline-width:	선의 굵기(px, %, bold, thin, thick, medium)
outline-color:	선의 색상(webcolor, 색상명, transparent) *transparent(투명)
outline-style:	선의 모양(solid, dotted, dashed, double, ridge, groove, inset, outset, none)
outline:	*4면전체를 (순서는 크게 상관 없음) 순으로 한꺼번에 기입
- outline: 1px solid #fff;	outline: 굵기 모양 색상;

상,하,좌,우 특정 위치 별도로 지정 불가능

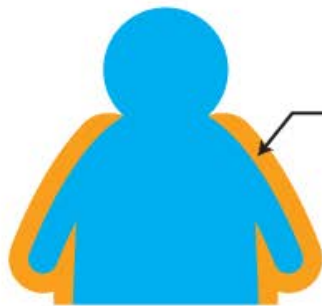
특정 브라우저에서 표기되기에, reset css에서 삭제하기 위해 많이 처리함.

border와 사용법은 같으나 위에 명기된 내용처럼 한쪽면만 사용이 불가능.



주변의 다른 개체와의
거리와 연관있음(외부여백)

값이 늘어날경우 주변에
영향력을 미침(위치변동)



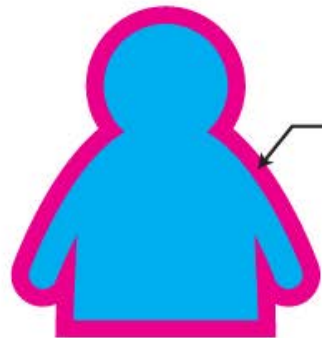
padding

스스로의 크기값이 늘어남(패딩 옷 입듯!)

실제색상은 다를 수 없음

단,본질의 크기는 불변!!(내부여백)

값이 늘어날경우 주변에 영향력을 미침(주변개체 위치이동)

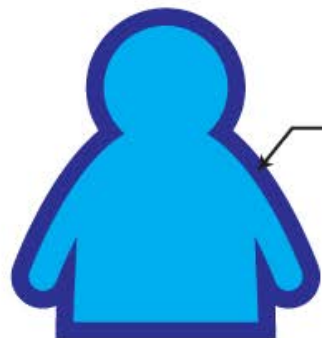


border

개체 주변의 테두리선이 생김

개체와의 색상을 다르게 주거나 투명하게 처리가능

값이 늘어날경우 주변에 영향력을 미침(변개체 위치이동)



outline

개체 주변의 테두리선이 생김

개체와의 색상을 다르게 주거나 투명하게 처리가능

단, 부분 삭제처리 불가능

크기값이 늘어나도 주변에 영향을 미치지 않음

* 모아놓고 보면 요런형태!!!!

----- 보이지 않는 영역

