HTML 기초

1. HTML이란?

* HTML은 HyperText Markup Language의 약자입니다.
* 웹 페이지는 HTML 문서라고도 불리며, HTML 태그들로 구성됩니다.

각각의 HTML 태그는 웹 페이지의 디자인이나 기능을 결정하는데 사용됩니다.

1. HTML 태그(tag)

* HTML 태그는 태그 이름을 꺾쇠 괄호(<>)로 감싸서 표현합니다.
* HTML 태그는 보통 시작 태그(start tag, opening tag)와 종료 태그(end tag, closing tag)의 한 쌍으로 구성됩니다.
* 종료 태그는 시작 태그와 전부 똑같지만, 태그 이름 앞에 슬래시(/)가 존재합니다.
* 태그에 따라 시작 태그만 있고 종료 태그가 없는 태그도 존재합니다.
* (br, hr, input, meta) - 종료태그가 없는 태그는 <br/> 이렇게 표기하기도 한다.

1. HTML 작성

* HTML 문서는 윈도우의 메모장, 리눅스의 vi와 같은 기본 에디터로도 작성할 수 있습니다.
* HTML 문서의 작성을 마친 후에 확장자를 .html로 저장하면 웹 브라우저에서 바로 확인할 수 있습니다.

1. HTML 버전

* 인터넷의 발전에 따라 HTML 버전 또한 다음과 같이 발전을 거듭하고 있습니다.

|  |  |  |
| --- | --- | --- |
| 버전 | 년도 | DOCTYPE 및 설명 |
| HTML 1.0 | 1991 | 팀 버나스리(Tim Berners-Lee)가 발표한 최초의 HTML |
| HTML 2.0 | 1995 | 국제 표준으로 제정된 최초의 HTML |
| HTML 3.2 | 1997 | W3C에 의해 제정된 최초의 HTML |
| HTML 4.01 | 1999 | <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  "http://www.w3.org/TR/html4/strict.dtd"> |
| XHTML 1.0 | 2000 | <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"> |
| HTML 5 | 2014 | <!DOCTYPE html> 🡪 html5 언어로 해석을 하는구나! |

* HTML의 최신 버전인 HTML5에 대한 더 자세한 정보를 원한다면, W3C 공식 사이트를 방문하여 확인할 수 있습니다.
* W3C HTML5 : W3C Recommendation 28 October 2014 => <https://www.w3.org/TR/html5/>

1. W3C

* W3C는 World Wide Web Consortium의 약자입니다.
* W3C는 월드 와이드 웹(WWW)을 위한 표준을 제정하고 관리하는 중립적인 기관입니다.
* 이러한 W3C가 관리하는 대표적인 웹 표준은 다음과 같습니다.
  1. HTML
  2. CSS
  3. DOM
  4. SVG
  5. XHTML
  6. XML – 태그가 붙음/ 데이터 베이스를 대체해서 사용 함

( 워드 – doc,xls -> 검색x 내부 내용 / 비오픈

Docx, xlsx, hwpx – 검색을 용이하게 – 오픈형

#### 주석( Comment)

* 주석(comment)이란 개발자가 작성한 해당 코드에 대한 이해를 돕는 설명이나 디버깅을 위해 작성한 구문을 의미합니다.
* 주석은 다른 HTML 코드와는 달리 웹 브라우저에 의해 표현되지 않습니다.
* 문법 : < ! -- 주석내용 -- > ctrl + /.

#### 예약어 ( = 엔티티 Entity)

* HTML에는 미리 예약된 몇몇 문자가 있으며, 이러한 문자를 HTML 예약어(reserved characters)라고 부릅니다.
* 이러한 HTML 예약어를 HTML 코드에서 사용하면, 웹 브라우저는 그것을 평소와는 다른 의미로 해석합니다.
* 따라서 HTML 예약어를 기존에 사용하던 의미 그대로 사용하기 위해 별도로 만든 문자셋을 엔티티(entity)라고 합니다.

|  |  |  |
| --- | --- | --- |
| 엔티티 문자 | 엔티티 이름 | 설명 |
|  | &nbsp; | 줄 바꿈 없는 공백 |
| < | &lt; | 보다 작은 |
| > | &gt; | 보다 큰 |
| & | &amp; | AND 기호 |
| " | &quot; | 큰따옴표 |
| ' | &apos; | 작은따옴표 |

#### 문자셋(Character set)

* 웹 브라우저가 HTML 문서를 정확하게 나타내기 위해서는 해당 문서가 어떠한 문자셋으로 저장되었는지를 알아야 합니다.
* 따라서 HTML 문서가 저장될 때 사용된 문자셋에 대한 정보를 <head>태그 내의 <meta>태그에 명시합니다.
* HTML4에서 UTF-8의 경우 : <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
* HTML5에서 UTF-8의 경우 : <meta charset="UTF-8">
* 위의 두 예제는 해당 HTML 문서가 UTF-8 문자셋을 사용하여 저장되었음을 웹 브라우저에 알려줍니다.
* Html : <meta charset="UTF-8"> 문자셋 정의를 안해주면 한글이나 기타 언어 표시가 깨진다.
* Css : @charset ‘utf-8’;

▶ 문자셋의 종류

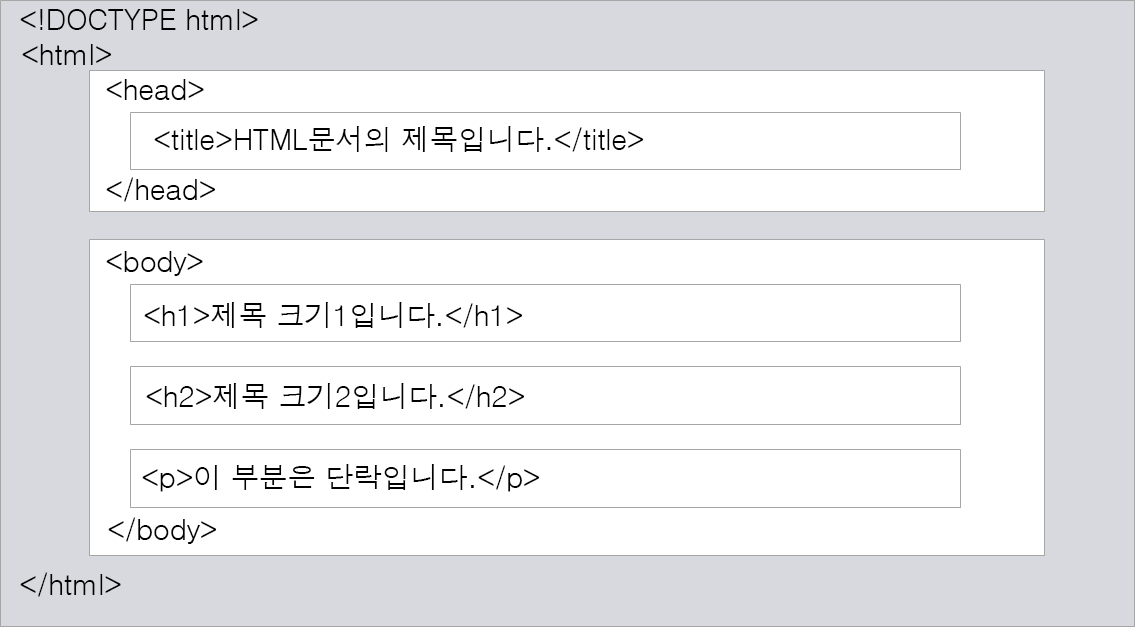
1) ASCII : 가장 처음 만들어진 문자셋으로, 인터넷에서 사용할 수 있는 127개의 영문자와 숫자로 구성

2) ANSI : 윈도우즈에서 만든 문자셋으로, 총 256개의 문자 코드를 지원

3) ISO-8859-1 : 256개의 문자 코드를 지원하는 HTML4의 기본 문자셋

4) UTF-8 : 세상에 있는 거의 모든 문자를 표현. 유니코드 문자를 지원하는 HTML5의 기본 문자셋

HTML 기본 구조



1. <!DOCTYPE html> : 현재 문서가 HTML5 문서임을 명시합니다.
2. <html> : HTML 문서의 루트(root) 요소를 정의합니다.  – 시작점, 뿌리 = 루트요소
3. <head> : HTML 문서의 메타데이터(metadata)를 정의합니다.

- 메타데이터(metadata)란 HTML 문서에 대한 정보(data)로 웹 브라우저에는 직접적으로 표현되지 않는 정보를 의미합니다. – 외부데이터 연결할 때

- 이러한 메타데이터는 <title>, <style>, <meta>, <link>, <script>, <base>태그 등을 이용하여 표현할 수 있습니다.

- <meta> : 문서의 정보를 알려줌

- <style> : 스타일시트(서식) html 문서 내에서 지정

- <link> : 외부 스타일시트 문서(.css)파일을 링크 걸 때(연결)

- <script> : 자바스크립트/제이쿼리를 작성하거나 직접적용 or 외부 파일( .js) 연결할 때

1. <title> : HTML 문서의 제목(title)을 정의하며, 다음과 같은 용도로 사용됩니다.

- 웹 브라우저의 툴바(toolbar)에 표시됩니다.

- 웹 브라우저의 즐겨찾기(favorites)에 추가할 때 즐겨찾기의 제목이 됩니다.

- 검색 엔진의 결과 페이지에 제목으로 표시됩니다.

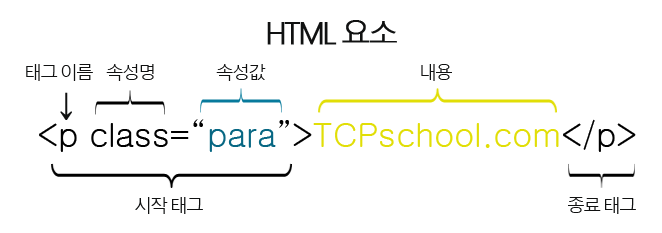
5) <body> : 웹 브라우저를 통해 보이는 내용(content) 부분입니다.

1. <h1> ~ <h6> : 제목(heading)을 나타냅니다.
2. <p> : 내용/단락(paragraph)을 나타냅니다.

HTML 요소 구조

#### HTML 요소 구조

* HTML 요소(element) = 태그는 여러 속성을 가질 수 있으며, 이러한 속성(attribute)은 해당 요소에 대한 추가적인 정보를 제공합니다. – html 속성 : attribute = attr / 스타일시트 속성 = css
* 또한, HTML 요소는 시작 태그로 시작해서 종료 태그로 끝납니다.



* 속성은 HTML 요소 중에서도 언제나 시작 태그 내에서만 정의되며, 속성 이름과 속성값(value)으로 표현됩니다.

##### ▶ 문법

<태그이름 속성이름="속성값"></태그이름>

#### 속성 이름은 언제나 소문자로 작성하자!

* HTML5 표준에서는 속성 이름에 대소문자를 구분하지 않고 있습니다.
* 하지만 W3C에서는 속성 이름을 사용할 때 될 수 있으면 소문자로 작성하도록 권장하고 있습니다.
* 또한, XHTML에서는 속성 이름을 더욱 엄격하게 소문자로만 사용해야 합니다.

#### 속성값은 언제나 따옴표로 감싸자!

* HTML5 표준에서는 속성값에 따옴표 사용을 강제하지 않습니다.
* 하지만 속성값을 따옴표로 감싸지 않으면, 다음과 같은 예상치 못한 오류가 발생할 수 있습니다.

ex) good - <img src="quotes.jpg" alt="이미지가 없어요">

bad - <img src="quotes.jpg" alt=이미지가 없어요>

* 위의 예제와 같이 속성값에 띄어쓰기가 들어가게 되면, 반드시 따옴표를 사용해야 정확한 값을 저장할 수 있습니다.
* 속성값을 감쌀 때는 보통 큰따옴표("")가 사용되며, 작은따옴표('')도 사용할 수 있습니다.
* Html – “” / css- ‘ ’ , “ ”

HTML – 텍스트, 목록, 표 태그

▶ 블록태그 – 줄바꿈

|  |  |
| --- | --- |
| 태그 | 설명 |
| <h1> <h2>…<hn> | 제목 표시 <h1> ~ <h6> |
| <p> | 단락 만들기(한 덩어리) |
| <br> | 줄 바꾸기 – 닫는 기호x |
| <hr> | 한 줄 넣기(수평) – 닫는 기호x |
| <blockquote> | 인용문 넣기 – 인용 내용 문단이 들여쓰기 됨 – 줄바꿈0 |
| <pre> | 입력한 그대로 화면에 표시 (소스에 포함된 모든 내용이 입력한 그대로 보여짐) |
| <div> | 박스(네모)를 만들어 주는 태그 |

▶ 인라인 태그 - 한줄로 표시 – a, span, input, 가상요소

|  |  |
| --- | --- |
| 태그 | 설명 |
| <strong>  <b> | 굵게 |
| <em>  <i> | 이탤릭체(기울임) |
| <q> | 인용문 - 내용이 따옴표로 묶임 / 줄바뀜x |
| <mark> | 노랑색 형광펜 그어 놓은 효과 적용됨 |
| <span> | 줄 바꿈 없이 한 달락 묶기 |
| <ruby>  <rt> | 덧말 넣기 - 동아시아 글자 표시/ |
| <sub> | 위 첨자 |
| <sup> | 아래 첨자 |
| <s> | 취소선 |
| <u> | 밑줄 |
| <abbr> | 약자 표시 ex) <abbr title=”Internet Of things”> IoT </abbr> |

▶ 목록을 만드는 태그

|  |  |  |
| --- | --- | --- |
| 태그 | 설명 | |
| <ul> 기호 목록  <li> 목록 요소 | <ul type=”circle”>  <li> 내용> </li>  </ul> | Type 속성 값   * Disc : ● * Circle : ○ * Square : ■ * None : 없음 |
| <ol> 숫자 목록  <li> 목록 요소 | <ol type=”A”>  <li> 내용> </li>  </ol> | Type 속성 값   * 1 : 숫자 * A : 알파벳 대문자 / a : 소문자 * I : 로마자 대문자 / I : 소문자 |
| <dl> 제목 있는 목록  <dt> 제목  <dd> 내용 | <dl>  <dt> 제목</dt>  <dl> 내용 </dl>  </dl> | |

▶표 만드는 태그

|  |  |
| --- | --- |
| 태그 | 설명 |
| <table>  <tr>행  <td>내용셀  <th>제목셀  <thead>제목영역  <tbody>내용영역  <tfoot>요약영역 | 예시  <table>  <tr>  <th> 제목 </th>  <td> 내용 </td>  <td> 내용 </td>  </tr>  </table> |
| 속성-colspan | 열 합치기 ex) <td colspan=”합칠 셀 개수”> |
| 속성-rowspan | 행 합치기 ex) <td rowspan=”합칠 셀 개수”> |
| <caption> 표제목 | * 테이블 태그 바로 다음에 써준다. * 표 상단 중앙 정렬 되어 표시 * 제모 여러 줄 표시 가능 /텍스트 꾸밀 수 있음 |
| <figure>  <figcation>  설명문 | <figure>  <figcaption> 표설명 </figcaption> 🡪 요소 위에 써주면 위에 위치,  <table>~~~~ </table> 요소 아래에 써주면 아래에 위치함  </figure>   * Figure와 figcaption 한 셋트 * 중앙정렬 x |
| 설명문  Id 속성으로 연결 | <p id=”summary”> 문단 </p>  <table aria-describedby=”summary”> id가 summary인 것을 설명문으로 인식 |

폼 관련 태그

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | 속성 | 설명 | | 비고 |
| 1 | <form> | 폼태그 | | <form> ~ </form> 태그 사이에  form 요소와 관련된 태그를 넣는다. |
| 2 | 속성 | method | | 사용자가 입력한 데이터를 서버에 넘길 때 어떻게 넘겨줄 것인가?  get : 입력한 내용 드러남, 데이터 제한 있음(265byte~4096byte)  post : 입력한 내용 드러나지x, 내용 길이 제한x |
| 3 | name | | 폼 이름 지정, 여러 개의 폼을 만들 경우 구분하기 위함. |
| 4 | action | | 폼 태그 안에 내용들을 처리해줄 서버 프로그램 지정  Ex) <form action=”reg.php”>  폼 내용을 reg.php 파일에 값을 넘겨서 처리해라 |
| 5 | autocomplete | | 입력했던 내용 자동 완성  ex) <form autocomplete=”off”> / 기본값 : on |
| 6 | <fieldset> | 폼의 영역 묶기 | | |
| 7 | <legend> | <flideset> 태그를 묶은 영역에 제목 달기 | | |
| 8 | <label> | 레이블 | | 폼 요소에 레이블(텍스트 붙이기  ex) <label> 아이디 <input type=”text”> </label>  <label for=”user”> 아이디 < /label>  <input type=”text” id=”user> |
| 9 | <input> | 입력항목 태그 | |  |
| 10 | 속성 | type | | 1. text : 텍스트 입력칸 2. password : 비밀번호 3. search : 검색창 4. tel : 전화번호 5. email : 이메일 6. url: 링크-인터넷 주소 7. hidden : 화면에 보이지 않지만 서버에 함께 넘겨야 할 값 있을 때   --------------------------------------------------------------------   1. radio : 라디오버튼 – 여러 개 중 한 개만 선택 2. checkbox : 체크박스 – 여러 항목 선택 가능   --------------------------------------------------------------------   1. submit : 전송버튼 2. reset : 리셋버튼 3. button : 버튼 4. file : 파일첨부 5. image : 이미지 버튼 / <input type=”image” src=” 이미지.jpg” >   --------------------------------------------------------------------   1. number : 숫자 조절할 수 있는 화살표 넣음(스핀박스) 2. range : 숫자 조절할 수 있는 슬라이드 막대   함께 지정가능한 속성 - max=”최대값” / min=“최소값” / step=“증감값”  --------------------------------------------------------------------   1. date : 날짜 선택 2. time : 시간 선택 3. datetime: 날짜+시간 |
| 11 | name | | 요소이름 |
| 12 | value | | 서버로 넘길 요소의 값 / 또는 기본값 설정 |
| 13 | autofocus | | 페이지 열었을 때 커서 위치시키기 |
| 14 | placeholder | | 힌트 표시 ex) placeholder =”아이디” |
| 15 | redolny | | 읽기 전용 (입력 x) |
| 16 | required | | 필수 입력칸 |
| 17 | max / min / step | | 최소값 / 최대값 / 증감값 |
| 18 | size | | 텍스트 필드 길이 |
| 19 | maxlength/minlength | | 최대 글자수 / 최소 글자수 |
| 20 | checked | | 라디오/체크박스 체크 기본값 |
| # | 태그 | | 설명 | 비고 |
| 21 | <select>  <option>-요소 | | 드롭다운목록 | <select>  <option value =”값”> 내용 </option>  <option value =”값”> 내용 </option>  <option value =”값” selected> 내용 </option>  </select> |
| <select>속성 | | size  multifle | 화면에 표시될 드롭다운 목록의 개수  여러 개의 옵션 선택 가능 – ctrl키 누름 |
| <option>속성 | | selected | 기본으로 선택되어 있는 옵션 |
| 22 | <datalist>  <option>-요소 | |  | <input type=”text” list=” 데이터목록 id “>  <datalist id=”데이터목록” >  <option value =”값”> 내용 </option>  <option value =”값”> 내용 </option>  <option value =”값”> 내용 </option>  </datalist> |
|  | <option>속성 | | value  label | 서버에 넘겨질 값  브라우저에 표시할 레이블 / 지정하지 않으면 value값을 레이블로 사용 |
| 23 | <textarea> | | 여러줄 입력하는 텍스트 영역 | <textarea name=“intro” cols=”60” rows=”5”> 내용 </textarea> |
| 속성 | | name | 요소 이름 |
| cols | 가로 너비 - 문자개수 |
| rows | 세로 높이 - 줄 개수 |
| 24 | <button> | | 버튼 넣기 |  |

이미지태그

문법 : **<img src=”이미지경로” alt=”대체텍스트”>**

* alt : 이미지가 뜨지 않을 경우 나타날 대체텍스트
* src : 이미지 경로
* 웹 페이지에서 주로 사용되는 이미지의 종류는 다음과 같습니다.

|  |  |  |
| --- | --- | --- |
| JPEG 이미지 | GIF 이미지 | PNG 이미지 |
| jpg | gif | png |

텍스트 관련 CSS

|  |  |  |  |
| --- | --- | --- | --- |
| # | 속성 | 설명 | 비고 |
| 1 | font-family | 글꼴 | font-family : 글꼴이름, ‘글꼴 이름’; |
| 2 | font-size | 글자크기 | font-size : 16pt / 3em / 0.8em / 120% |
| 3 | font-weight | 굵기 | font-weight : normal / bold / 100~900 사이의 수치 |
| 4 | font-style | 기울임 | font-style : normal / italic |
| 5 | font | 폰트 속성 한번에 지정 | font : “맑은 고딕” bold italic **16px/25px**  글자크기 / 줄간격 |
| 6 | color | 글자색 |  |
| 7 | text-decoration | 텍스에 줄 긋기 | text-decoration : none / underline / overline / lingh-through  없음 / 밑줄 / 윗줄 / 취소선 |
| 8 | test-shadow | 텍스트 그림자 | text-shadow : 7px 3px 5px red ;  가로거리 세로거리 번짐정도 색상 |
| 9 | letter-spacing | 글자 간격 | letter-spacing : 0.2em  word-spacing : 0.5em  ( **em 단위로 지정**해야 글꼴 사이즈에 따라 자간이 보기 좋게 유지 된다.) |
| 10 | word-spacing | 단어 간격 |
| 11 | text-align | 정렬 | text-align : left / right / center / **justify – 양쪽정렬** |
| 12 | text-indent | 들여쓰기 | text-indent : 16px / 5% |
| 13 | line-height | 줄간격 | line-height: 30px / 0.7 / 200% |
| 14 | list-style-type | 목록-불릿 옵션 | list-style-type : disc / circle / square / none |
| 15 | list-style-image | 목록-불릿 이미지 | list-style-image : url ( ‘그림.png ‘ ); |
| 16 | list-style-positon | 목록-들여쓰기 | list-style-position : outside / inside |
| 17 | list-style | 목록 속성 한번에 지정 |  |

▶ 단위

* px : 고정 - 픽셀
* % : 상대 – 백분율 – 부모 요소 기준으로 % 결정 (주로 너비/높이 지정에 사용)
* em : 상대 – 부모요소 기준
* rem : 상대 – body 기준 ( body의 폰트 기본크기 16pt)
* vw : 상대 – 브라우저(뷰포트)의 너비의 1/100 = 1vw ( ex. 브라우저 너비 : 800px이면? 1vw = 8px)
* vh : 상대 – 브라우저(뷰포트) 의 높이의 1/100 = 1vh (ex. 브라우저 높이 : 750px이면? 1vh=7.5px)

링크 태그

1. **문법**

- <a href=”링크 주소” target =”\_blank” > 링크 </a>

1. **target 속성**

* <a>태그의 target 속성은 링크로 연결된 문서를 어디에서 열지를 명시합니다.

|  |  |
| --- | --- |
| target 속성값 | 설명 |
| \_blank | 링크로 연결된 문서를 새 창이나 새 탭에서 염. |
| \_self | 링크로 연결된 문서를 현재 프레임(frame)에서 염. (기본설정) |
| \_parent | 링크로 연결된 문서를 부모 프레임(frame)에서 염. |
| \_top | 링크로 연결된 문서를 현재 창의 가장 상위 프레임(frame)에서 염. |
| 프레임(frame) 이름 | 링크로 연결된 문서를 지정된 프레임(frame)에서 염. |

#### 링크의 상태(state)

* HTML 링크의 상태는 다음과 같이 네 가지로 구분할 수 있습니다.

|  |  |
| --- | --- |
| 링크의 상태 | 설명 |
| link | 아직 한 번도 방문한 적이 없는 상태 (기본설정) |
| visited | 한 번이라도 방문한 적이 있는 상태 |
| hover | 링크 위에 마우스를 올려놓은 상태 |
| active | 링크를 마우스로 누르고 있는 상태 |

- 기본적으로 링크가 걸린 텍스트는 밑줄, 파란색으로 변경됩니다.

- visited 상태의 링크는 밑줄, 보라색으로 변경됩니다.

- active 상태의 링크는 밑줄, 빨간색으로 변경됩니다.

#### 페이지 책갈피 ( 앵커 )

* <a>태그의 name 속성을 이용하면 간단한 책갈피를 만들 수 있습니다.
* 우선 책갈피를 통해 가고 싶은 위치에 <a>태그를 만들고 name 속성을 작성합니다.
* 그 다음에 작성한 name 속성값을 이용하여 다른 <a>태그에서 링크를 걸면 됩니다.

<a href="#bookmark"><p>제목 3으로 갑시다!!!</p></a>

...

<h2><a name="bookmark"></a>제목 3</h2>

색상/배경 관련 CSS

|  |  |  |  |
| --- | --- | --- | --- |
| # | 속성 | 설명 | 비고 |
| 1 | background-color | 배경색 | background-color : rgb (255, 255, 0)  background-color : rgba (255, 255, 0, 0.5) 빨강/초록/파랑/알파값(투명도)  background-color : #ffcc00  background-color : #fc0 🡪 #ffcc00을 #fc0으로 줄여 쓸 수 있다.  background-color : hsl ( 색상, 명도, 채도)  background-color : hsla ( 색상, 명도, 채도, 알파값) |
| 2 | background-clip | 배경적용 범위 | border-box : 박수의 가장 외곽-테두리까지 적용 ---- 기본  padding-box : 박스의 패딩범위까지 적용  contents-box : 박스의 내용 부분에만 적용 |
| 3 | background-origin | 배경이미지 기준 조점 | border-box : 박스의 가장 외곽-테두리부터 시작  padding-box : 박스의 패딩범위부터 시작 ---- 기본  contents-box : 박스의 내용범위부터 시작 |
| 4 | background-image | 배경이미지 | background-image : url ( ‘ img/bg.jpg ‘ ) ; |
| 5 | background-rept | 반복방법 지정 | background-rept : repeat / repeat-x / repeat-y / no-repeat |
| 6 | background-position | 배경이미지 위치 조절 | background-position : 수평위치 수직위치   * 수평 : left / center / right * 수직 : top / center / bottom * ex) background-position : left bottom ; |
| 7 |  |  |  |
| 8 | background-attachment | 배경이미지 고정 | scroll : 화면과 함께 스크롤 – 기본값  fixed : 배경이미지 고정 |
| 9 | background-size | 배경이미지 사이즈 | auto : 원래 배경 이미지 크기만큼 표시  contain : 요소 안에 배경이미지 다 표시되도록 확대/축소  cover : 요소 안에 배경이미지가 꽉 차도록   * background-size : 200px 150px ; 가로/세로 크기 직접 지정 * background-size : 60% 40% ; |
| 10 | background | background : url( ‘ img/bg.jpg’ ) no-repeat fixed right bottom ; | |

▶ 색상

* rgb : 빨강 / 녹색 / 파랑
* hsl : 색상 / 명도 / 채도
* 16진수 : #ff00c9 : 빨강 / 녹색 / 파랑 – 각 색상별로 0~255 수치를 정하고 16진수로 변환하여 표기
* 투명도 : 0~1 사이의 값 지정함 ex) rgba ( 255, 128, 192, 0.8) = 80% 불투명

▶ 그라데이션

* 선형 : linear-gradient( 각도 to 방향, 색상값 , 색상값, 색상값 … )
* 원형 : radial-gradient( 최종모양 크기 at 위치 , 색상값, 색상값, 색상값 … )

.bg {

background : blue;

background : linear-gradient( to right bottom, blue, white ) ; 오른쪽 아래 방향으로 그라데이션

background : linear-gradient( 45deg , blue, white ) ; 45도 기울여서 선형 그라데이션

background : radial-gradient ( ellipse, red, yellow) ; - 타원형 기본값

background : radial-gradient ( circle at 10% 10%, red, yellow) ; - 원형

}

박스모델 CSS

|  |  |  |  |
| --- | --- | --- | --- |
| # | 속성 | 설명 | 비고 |
| 1 | display | 화면 배치 방법 결정 | none : 화면에 보이지 않음 (공간차지x)   * visibility : hidden – (화면보이지x , 공간차지0)   block : 블록레벨 – 위/아래로 요소가 쌓임 = 줄 바뀜.  inline : 인라인레벨   * 한 줄로 나열됨, 브라우저 너비가 모자라면 다음 줄로 바뀜. * **width, height, margin, padding 적용 X**   **inline-block : 인라인 레벨 – width, height, margin, padding 적용 0**  table : table 처럼 표현  table-row : tr처럼 표현  table-cell : td처럼 표현ㄴ |
| 2 | width | 너비 | 고정형 px , 가변형- %, vw  고정형 px , 가변형- %, vh   * margin : 0 auto ; 🡪 요소를 중앙에 배치 할 때 사용 |
| 3 | height | 높이 |
| 4 | margin | 바깥여백 |
| 5 | padding | 안쪽여백 |
| 6 | border-style | 테두리 스타일 | none : 나타나지 않음  soilid : 실선  dashed: 파선  dottoed : 점선  double : 이중 겹선 |
| 7 | border-width | 테두리 두께 | border-width : 2px ; 🡪 사방 모두 2px  border-width : 2px 4px ; 🡪 위,아래 : 2px / 왼쪽,오른쪽 : 4px  border-width : 10px 5px 15px 20px ; 🡪 시계방향 – 위, 오른쪽, 아래, 왼쪽 순서 |
| 8 | border-color | 테두리 색상 |  |
| 9 | border |  | border : 1px solid black ; |
| 10 | border-radius | 박스 모서리 둥글게 | border-radius :  border-top-left-radius :  border-bottom-right-radius : |
| 11 | box-shoadow | 박스 그림자 | box-shadow : 7px 3px 5px red ;  가로거리 세로거리 번짐정도 색상   * 수치를 음수로 지정하면 반대방향   box-shadow : inset 7px 3px 5px red ;   * inset : 내부 그림자 만들 때 옵션 추가 |
| 12 | box-sizing | 박스 너비 기준 정하기 | border-box : width 값에 padding, border 수치 포함  content-box : width 값에 padding, border 포함x, 콘텐츠 영역만 크기 지정   * 박스사이즈 : width + padding \*2 + border \* 2 |

▶ 블록레벨 vs 인라인 레벨 요소

|  |  |
| --- | --- |
| 블록레벨 | * 혼자 한 줄 차지, 옆에 다른 요소 올 수 없음 * 너비, 마진, 패딩 등을 지정하려면 블록레벨 요소이어야 함.   ex) <div> <p> <table> <form> |
| 인라인레벨 | * 줄을 차지하지 않음. 화면에 표시 되는 영역만큼 차지 * 가로로 나열 한 줄에 여러 개 표시 * 마진, 패딩 지정x   ex) <img> <span> <a> |

Q. 인라인 레벨을 블록 레벨로 변경하려면?

- display 속성 사용 – display : block /

Q. 인라인 레벨에 너비, 높이 마진, 패딩을 적용하려면?

- 블록 레벨로 변경해야함 - display : block / inline-block;

▶ 전체 너비 계산 방법

* 내용영역 + 패딩(안쪽여백) \* 2 + 테두리 \* 2

ex) width 500px, padding 10px, border 2px 이라면? 전체 너비 : 500px + 10px \* 2 + 1px \* 2 = 522px;

Q. width를 지정할 때, padding, border 사이즈가 포함되게끔 하려면?

- box-sizing : border-box ; 🡪 속성 함께 지정

다단/표 관련 CSS

|  |  |  |  |
| --- | --- | --- | --- |
| # | 태그 | 설명 | 비고 |
| ▶ 다단 관련 | | | |
| 1 | column-width | 다단 만들기  단의 너비 고정 | column-width : 120px   * 단 **개수**가 브라우저 너비에 따라 **달라짐** |
| 2 | column-count | 다단 만들기  단의 개수 고정 | column-count : 3   * 단 **너비**가 브라우저의 너비에 따라 **달라짐** |
| 3 | column-rule | 구분선 속성 | column-rule : 1px solid red ; 🡪 border 속성과 사용법 같음   * column-rule-width : 선 굵기 * column-rule-style : 선 모양 * column-rule-color : 선 색상 |
| 4 | column-span | 단 합치기 | column-span |
| 5 | break-before  break-after | 특정 요소 앞 or 뒤에서  단 나누기 | column : 단 나눔0  avoid-column : 단 나누지 x |
| ▶ 표 관련 | | | |
| 6 | border-collapse | 테두리 통합/분리 | collapse : table, td 테두리 합치기  separate : table, td 테두리 분리 |
| 7 | caption-side | 표 제목 위치 | top : 표 위에 캡션 위치  bottom : 표 아래에 캡션 위치 |
| 8 | border-spacing | 인접한 셀 테두리  사이 거리 지정 | border-spacing : 10px ;  border-spacing : 10px 20px ;ㄴ |
| 9 | empty-cells | 빈셀 표시 여부 지정 | show : 빈셀 표시0  hide : 빈셀 표시x |
| 10 | table-layout | 내용에 맞게  셀너비 조정여부 | auto : 셀 내용에 따라 너비가 달라짐(기본값)  fixed : 셀 너비보다 내용이 긴 경우 내용이 바깥으로 넘침 |
| 11 | test-align | 셀 안에서 수평정렬 | left / right / center |
| 12 | vertical-align | 셀 안에서 수직정렬 | top / middle / bottom 🡪 세로 정렬은 표에서만 적용됨 |

포지셔닝/레이아웃 관련 CSS

|  |  |  |  |
| --- | --- | --- | --- |
| # | 속성 | 설명 | 비고 |
| 1 | position | 웹문서 안에서 요소의  위치를 지정하는 방법 | static : 태그를 나열한 순서대로 배치 - 위치지정x  relative : 태그를 나열한 순서대로 배치, 위치지정 0   * 기준 : 태그 순서상 내 위치 * left, right, top, bottom 속성으로 좌표 값으로 이동   absolte : 원하는 위치에 자유롭게 배치   * 부모 요소 안에서 원하는 위치 배치 * 단, **부모요소 반드시 position : relative로 지정**   fixed : 브라우저 기준으로 원하는 위치에 자유롭게 배치   * 브라우저 기준이므로 화면에 고정됨. |
| 2 | float | 화면의  왼쪽 or 오른쪽 배치 | left : 화면의 왼쪽에 배치  right : 화면의 오른쪽 배치   * 자신이 차지할 만큼만 차지하고 나머지 공간 비워둠. 그리고 요소를 왼쪽 or 오른쪽으로 배치. * float를 지정하지 않은 요소는 밑으로 깔림(텍스트는 깔리지 않고 밀림) * 다른 요소보다 한층 떠 있게 배치가 됨(포토샵에서 레이어처럼) |
| 3 | clear | float 속성 해제 | clear : both ; |
| 4 | z-index | 요소 쌓는 순서  (레이어 순서) | z-index 값이 작으면 아래쪽으로 / 크면 위쪽으로 올라감. |
| 5 |  |  |  |
| 6 | visibility | 요소를 보이거나  보이지 않게 함 | visible : 화면에 요소 표시  hidden : 화면에 요소 표시x, 자리는 차지함   * display : none ( 요소표시x, 자리차지x ) |

▶ float을 clear하는 방법(clearfix)

* CSS작업을 하다 보면 상단 메뉴 영역 만들 때 float속성으로 원하는 곳에 배치를 하면 그 다음에 오는 태그들의 위치가 뭉개진다 ( float 밑으로 깔리면서 뭉개지는 것)
* 원래 float 뒤에 포지션이 뭉개지는 현상은 브라우저상의 오류로 생겨난 것이고, 이를 해결하는 clearfix는 위의 오류를 해결하기 위해 똑똑한 분들이 만들어낸 일종의 버그해결 방법이다.
* 가상요소 after에 clear : both를 만들어 넣어 줌.
* nav :: after { content: ' ' ; display : block ; clear: both ㅉ} 🡪 3가지 옵션 1set임

다양한 선택자 / 가상클래스 / 가상요소

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | 분류 | 속성 | 비고 | |
| 다양한 선택자 | | | | |
| 1 | 연결  선택자 | div p | * 하위 선택자 | |
| 2 | div > p | * 자식 선택자 | |
| 3 | div + p | 인접 형제 선택자 – 가장 가까운 첫번째 동생 요소 선택 | |
| 4 | div ~ p | 형제 선택자 – 모든 형제 선택 | |
| 5 | 속성  선택자 | [속성] | 지정한 속성일 때 | |
| 6 | [속성 = “값”] | * 값이 일치 할 때 | |
| 7 | [속성 ~= “값”] | * 값을 포함할 때 – 하이픈 연결된 단어, 값의 일부만 일치할 경우 적용X * - class=”btss” 🡪 X, class=”bts-txt” 🡪 X, class=”btx txt” 🡪 0 | |
| 8 | [속성 |= “값”] | * 값을 포함할 때 – 하이픈 연결된 단어 적용0 / class=”bts-txt” 🡪 0 | |
| 9 | [속성 \*= “값”] | * 값이 포함할 때 - 일부라도 포함 되어 있으면 적용0 | |
| 10 | [속성 ^= “값”] | * 값으로 시작할 때 | |
| 11 | [속성 $= “값”] | * 값으로 끝날 때 | |
| * 가상 클래스 – 여러 태그 중 원하는 태그 선택하기 위함 | | | | |
| 12 | 사용자  동작에 반응 | 1. : link | * 방문하지 않은 링크 | * ☞ 4가지 순서 반드시 순서대로 사용. |
| 13 | 1. : visited | * 방문한 링크 |
| 14 | 1. : hover | * 마우스 올렸을 때 |
| 15 | 1. : active | * 마우스 클릭 했을 때 |
| 16 | : focus | * 포커스가 위치 했을 때 | |
| 17 | UI요소  상태에 따른 | : enabled  : disabled | * 요소를 사용 할 수 있는 상태일 때 * 요소를 사용 할 수 없는 상태일 때 | |
| 18 | : checked | 라디오 박스 or 체크 박스 - 해당 항목이 체크되어 있을 때 | |
| 19 | 구조 가상클래스  (웹문서의 구조를 기준으로 특정 위치에 있는 요소를 찾음. | : root | 루트요소 – html 문서에서는 최상위 태그 <html> 뜻함. | |
| 20 | : first-child  : last-child | 모든 자식 요소 중 – 해당 태그가 첫 번째 자식 요소이면 적용  모든 자식 요소 중 – 해당 태그가 마지막 자식 요소이면 적용  ex) div > p : first-child   * div 중 첫번째 자식이 p이면 선택 (첫번째가 p가 아니면 선택 X) | |
| 21 | : nth-child(n)  : nth-last-child(n) | 앞에서 n번째 요소 적용  뒤에서 n번째 요소 적용  🡪 태그 상관 없이 해당 순서이면 선택 | |
| 22 | : first-of-type  : last-of-type | 특정 태그 중 - 첫 번째 자식 요소 - 모두 적용  특정 태그 중 - 마지막 자식 요소 - 모두 적용  ex) div > p : first-of-type  div 자식 중에 첫 번째로 오는 p 태그를 적용 ( p가 두번째 자식 이어도 처음 나오는 p라면 적용) | |
| 23 | : nth-of-type(n)  : nth-last-of-type(n) | 특정 태그들 중 - 앞에서 n번째 태그 요소 적용  특정 태그들 중 - 뒤에서 n번째 태그 요소 적용 | |
| 24 | : only-child | 자식 요소가 1개만 있을 때 적용 | |
| 25 | : only-of-type | 모든 자식 요소 중 유일한 요소일 때 적용 | |
| 26 | 그 외  가상 클래스 | : target | 앵커로 지정한 부분에 적용  ex) <a href=” **#intro** “> 앵커-링크 </a> / **#intro** : target { color : blue } | |
| 27 | : not | 괄호 안의 요소를 제외하고 적용  ex) p:not( **#ex** ) {color : blue} – p태그 중 id가 ex인 것 제외하고 모두 파란색 적용 | |
| 가상요소 – 내용의 일부만 선택하여 스타일 적용할 때 / 가상클래스와 구분하기 위해 콜론 2개 ( :: ) 붙임 | | | | |
| 28 | 가상요소 | :: before  :: after | 특정 요소 앞/뒤에 지정한 내용 추가 – 인라인 요소로 인식함.  ex) p :: before { contents : ‘내용’ } ; | |
| 29 | :: first-line  :: first-letter | 첫 번째 줄 / 첫 번째 글자에 스타일 적용 🡪 단, block 요소일 때만 적용 가능 | |
| 30 | :: selection | 사용자가 마우스로 드래그한 부분( 선택 영역) 에 대해 서식 적용 | |

변형(transform) CSS

|  |  |  |  |
| --- | --- | --- | --- |
| # | 속성 | 설명 | 비고 |
| 이동 / 확대, 축소 / 기울기 / 회전 | | | |
| 1 | translate (X, Y) | 이동 - X, Y | ex) transform : translate(10px , 10px) rotate(90deg) perspective(180deg) ; |
| 2 | translateX (X) | 이동 - X축 |
| 3 | translateY (Y) | 이동 - Y축 |
| 4 | scale ( X, Y ) | 확대, 축소 : 가로, 세로 |
| 5 | scaleX (X) | 확대, 축소 : 가로 |
| 6 | scaleY (Y) | 확대, 축소 : 세로 |
| 7 | skew(각도) | 기울기 : 가로, 세로 |
| 8 | skewX(각도) | 기울기 : 가로 |
| 9 | skewY(각도) | 기울기 : 세로 |
| 10 | rotate(각도) | 회전 |
| 변형과 관련된 속성들 | | | |
| 11 | transform-origin | 변형 기준점 설정 | x축 : left center right  y축 : top center bottom  ex) transform-orgin : left bottom ; |
| 12 | perspective | 원근감 적용 | perspective : 300px - 숫자가 클수록 멀리 밀어냄 |
| 13 | perspective-origin | 원금법 적용할 기준점 |  |

트랜지션/애니매이션 CSS

|  |  |  |  |
| --- | --- | --- | --- |
| # | 속성 | 설명 | 비고 |
| 트랜지션 | | | |
| 1 | transition -property | 트랜지션 적용할 속성 선택 | transition –property : all  transition –property : width, color ; |
| 2 | transition -duration | 트랜지션 진행시간 | transition –duration : 0.5s ; |
| 3 | transition -timing-function | 트랜지션 속도 곡선 | linear : 처음부터 끝까지 똑 같은 속도  ease : 천천히 시작, 점점 빨라지다, 천천히 끝냄  ease-in : 느리게 시작  ease-out : 느리게 끝냄  ease-in-out : 느리게 시작하고 느리게 끝냄  cubic-bezier : 사용자 정의 속도 🡪 https://cubic-bezier.com/ |
| 4 | transition -delay | 트랜지션 지연 시간 |  |
| 5 | transition | 한꺼번에 설정 | 지정순서: ① 속성② 진행시간 ③ 속도곡선 ④ 지연시간  ex) transition : width 2s ease 1s ; |
| 애니매이션 | | | |
| 6 | @keyframes | 애니매이션 바뀌는 지점 설정 |  |
| 7 | animation-name | 애니매이션 이름 | 적용할 keyframe 애니매이션 이름 지정 |
| 8 | animation –duration | 실행 시간 | 실행시간을 적용 안하면 애니매이션을 지정해도 실행이 안됨  반드시 실행시간 적어주어야 함! |
| 9 | animation -timing-function | 속도 곡선 | 트랜지션 옵션과 동일 |
| 10 | animation -delay | 지연 시간 |  |
| 11 | animation-direction | 종료 후 처음부터 시작할지  역방향으로 진행할지 | normal : 처음부터 실행, 반복  alternate : 처음부터 실행 후, 역방향으로 반복 |
| 12 | animation-iteration-count | 반복 횟수 | animation-iteration-count : 3 ; 🡪 3회 반복  animation-iteration-count : infinite ; 🡪 무한반복 |
| 13 | animation-play-state | 멈추거나 다시 시작 |  |
| 14 | animation | 한꺼번에 설정 | animation : move 3s alternate infinite ease ;  2개 keyframe 적용시 콤마( , )로 구분  animation : run 2s infinity **,** bg 3s infinity alternate ; |

|  |  |
| --- | --- |
| @keyframes 애니매이션이름 {  from { background-color : red; }  to { background-color : blue; }  } | @keyframes 애니매이션이름 {  0% { background-color : red ; }  50% { background-color : yellow ; }  100% { background-color : blue ; }  } |

div { animation : 애니매이션이름 3s ; }

미디어 쿼리

사이트에 접속하는 장치에 따라 특정한 CSS 스타일을 사용하도록 해주는 것.

1. 구문

* @media [only | not] 미디어유형 [and 조건] \* [and 조건]
* @media screen and (min-width:200px) and (max-width : 360px)

1. 연산자

|  |  |
| --- | --- |
| 연산자 | 설명 |
| and | 앞의 소스처럼 조건을 계속 추가할 수 있음 |
| , (쉼표) | 동일한 스타일 유형을 사용할 미디어의 유형과 조건이 있다면 쉼표를 이용해 추가 |
| only | 미디어 쿼리를 지원하는 웹 브라우저에서만 조건 인식하게 함.  Q. 만약 미디어쿼리 지원하지 않는 웹브라우저라면? 미디어쿼리 무시하고 실행x  (IE에서 미디어 쿼리를 제대로 인식X, 미디어쿼리 써도 큰 의미 없음) |
| not | not 다음에 지정하는 미디어유형 제외  EX) not tv 라고 지정한다면? tv를 제외한 미디어유형에만 적용 |

1. 미디어 유형 종류

|  |  |  |  |
| --- | --- | --- | --- |
| 연산자 | 설명 | 연산자 | 설명 |
| all | 모든 미디어 유형 | **braille** | 점자표시장치 |
| print | 인쇄 장치 | **handheld** | 패드(pad)처럼 손에 들고 다니는 장치 |
| screen | 컴퓨터 스크린(스마트폰 스크린 포함) | **projection** | 프로젝터 |
| tv | 음성과 영상이 동시 출력되는 tv | **tty** | 디스플레이 기능이 제한된 장치  (픽셀단위를 사용할수 없음) |
| aural | 음성 합성 장치  (주로 화면을 읽어 소리로 읽어주는 장치) | **embossed** | 점자 프린터 |

1. 미디어쿼리의 조건

* 미디어 쿼리는 특정 조건에 따라 적용할 css를 다르게 정의하므로 조건을 어떻게 체크할 것인지가 중요함.
* 미디어 쿼리에서 사용하는 조건에는 주로 화면 크기와 관련된 것들이 많음.
* **화면 크기를 체크해 크기에 따라 다른 css 가 적용되도록 하는 것.**

1. 가로 너비 / 세로 높이 속성

|  |  |  |  |
| --- | --- | --- | --- |
| 웹문서 | | 단말기 | |
| 속성 | 설명 | 속성 | 설명 |
| width, height | 웹페이지 가로 너비, 세로 높이 | device-width, device-height | 단말기 가로 너비, 세로 높이 |
| max-width, max-height | 최소 너비, 최소 높이 | max-device-width, max-device-height | 단말기 최대 너비, 최대 높이 |
| min-width, min-height | 최대 너비, 최대 높이 | min-device-width, min-device-height | 단말기 최소 너비, 최소 높이 |

@media all (min-width:600px) and (max-width : 959px) { }

뷰포트의 너비가 600px 이상이고 959이하일 때 다음과 같은 css 적용하겠다~

@media all (min-device-width:320px) and (min- device-height : 480px) { }

단말기의 너비가 320px 이상이고 높이가 480px 이상일 때 실행하겠다~

Q. 단말기와 뷰포트 크기 통일 할 때? meta태그의 viewport 속성 중 width = “device-width” 설정함.

css 파일 아래와 같은 구조로 주로 작성한다.

|  |
| --- |
| /\* 기본 - PC CSS \*/  @media screen and (max-width : 1024px) {  /\* 태블릿용 css \*/  }  @media screen and (max-width : 540px) {  /\* 모바일용 css \*/  } |

1. 외부 css 파일 연결하기

<link rel=”stylesheet” media=”미디어 쿼리 조건” href=”css 파일 경로”>

ex) <link rel=”stylesheet” media=”print” href=”css/print.css”>

1. @ import 구문

@import url(css파일경로) 미디어쿼리 조건

ex) @import url(“css/tablet.css”) only screen and (min-width:768px) ;

1. 웹문서에 직접 정의

|  |  |
| --- | --- |
| <style media=”screen and (min-width:768px)”>  body { }  </style> | <style>  @media screen and (min-width:768px) {  body{ }  }  </style> |

플렉스 박스

|  |  |  |  |
| --- | --- | --- | --- |
| # | 속성 | 설명 | 비고 |
| 플렉스 컨테이너 (부모) | | | |
| 1 | display : flex | flex 컨테이너로 만들기 |  |
| 2 | flex-direction | 방향 | row : 주축-가로축 / 교차축 – 세로축으로 설정 / 왼🡪오 배치  row-inverse : 오🡪왼 배치  column : 주축-세로축 / 교차측 – 가로축으로 설정/ 위🡪아래 배치  column-inverse : 아래🡪위 배치 |
| 3 | flex-wrap | 여러 줄 표시 여부 | no-wrap : 요소를 한 줄로만 배열  wrap : 요소를 여러 줄로 배열  wrap-reverse : 요소를 기존 방향의 반대방햐으로 배치 |
| 4 | flex-flow | flex-direction과  flex-wrap을 한번에 지정 | flex-flow : row no-wrap ; 🡪 기본값  flex-flow : column wrap |
| 5 | justify-contents | 수평 정렬 |  |
| 6 | align-items | 수직 정렬 – 한 줄 |  |
| 7 | align-contents | 수직 정렬 – 여러 줄 일 때 |  |
| 플렉스 요소 (자식) | | | |
| 8 | order | 정렬 : flex 박스 순서 변경 |  |
| 9 | flex | flex 항목 크기 조절 | flex : 비율  .item:nth-child(1) { flex : 3; } 전체의 3/5 차지  .item:nth-child(2) { flex : 1; } 전체의 1/5 차지  .item:nth-child(3) { flex : 1; } 전체의 1/5 차지 |
| 10 | align-self | 요소 개별 정렬 할 때 | .box { align-self:flex-start; } |

|  |  |  |
| --- | --- | --- |
| justify-content | align-itmes | align-content |
|  |  |  |

코딩할 때 주의사항

1) 해당 **영역을 의미하는 부모 아이디 OR 클래스**를 함께 붙여준다.

|  |  |
| --- | --- |
| before 🡪 | after |
| .logo{ } | **#sec  .logo{ }**  **#sec > .logo{ }** |

🡪 부모영역 id 또는 클래스를 적어서 어느 영역에 해당하는지 알아 볼 수 있게끔

2) 큰 영역 > 세부 요소 순으로 css 작성

|  |  |
| --- | --- |
| after | |
| #header .logo{}  #header .logo li{}  #header .logo li a{} | **#sec01 { }**  **#sec01 span{ }**  **#sec01 span.prev{ }**  **#sec01 span.next{ }** |

1. 겹치는 부분은 최대한 공통 클래스를 찾아내어 중복을 없앤다.

|  |  |
| --- | --- |
| before 🡪 | after |
| #sec01 span.prev{ width:20px; backgroud:yellow}  #sec01 span.next{ width:20px; backgroud:red} | **#sec01 span{ width:20px; }**  **#sec01 span.prev{ backgroud:yellow}**  **#sec01 span.next{backgroud:red}** |

4) 반복되는 CSS는 별도로 클래스를 만들어서 – 태그 클래스 속성 값에 **중첩으로 적용**해준다.

|  |  |
| --- | --- |
| after | |
| .line { border: 1px solid black }    .sec-1 { background : yellow}  .prev { background : blue}  .next { background : red} | <div class="sec-1**line**></div>  <span class="prev  **line**></span>  <p class="next  **line**></p> |

1. 선택자 사이를 한 칸 띄는 것과 / 공백이 없는 것 - 의미가 많이 다름

- div.box - div 중에 클랙스가 box

* div  .box - div 하위요소 중에 클래스 box

1. 미디어 쿼리에서 - 선택자 명칭을 항상 동일하게 맞춰 주어야 함.

같은 대상을 가리킨다고 할지라도 인식x

|  |  |
| --- | --- |
| 적용 X | 적용 0 |
| .header .logo{      width:100px; height: 100px;      background: yellow;  }    @media all and (min-width:768px){      .logo{background: red;}    } | **.header .logo**{      width:100px; height: 100px;      background: yellow;  }    @media all and (min-width:768px){  **.header .logo**{background: red } |