**1주차 과제**

1. Display 속성

* Block : 항상 새로운 줄에서 시작. 가능한 가로 전체의 너비를 차지 => 가로 전체 너비를 차지하며 세로로 쌓이는 형식

Ex) div, h1 … => <div>Introduce myself</div>, <h1>About Me</h1>

* Inline : 새로운 줄에서 시작하지 않고 문장 혹은 문단의 일부로 표시. 가능한 가로 너비만 차지하며 인라인이 포함하는 콘텐츠의 너비만큼만 공간을 차지

Ex) span, a… => <p>제 나이는<span style=”background-color: yellow;”>26살</span>입니다</p>, <p>네이버<a href=http://www.naver.com “ style=color: purple; text-decoration: underline;”>바로가기</a></p>

1. Layout

* HTML
  + <header> : 문서 or 섹션의 헤더를 정의. 문서 전체나 섹션의 상단에 배치

<header>

<h1>웹사이트 제목</h1>

<p>웹사이트 부제 또는 설명</p>

</header>

* + <nav> : 네비게이션 링크의 그룹을 정의. 주로 사이트의 메뉴나 페이지 링크 등을 포함 => 웹사이트 탐색에 사용

<nav>

<ul>

<li><a href="#">홈</a></li>

<li><a href="#">서비스</a></li>

<li><a href="#">포트폴리오</a></li>

<li><a href="#">연락처</a></li>

</ul>

</nav>

* + <section> : 문서의 섹션을 정의. 일반적으로 관련된 콘텐츠를 묶어 하나의 섹션으로 만듬

<section>

<h2>섹션 제목</h2>

<p>섹션 내용 설명</p>

</section>

* + <article> : 독립적이고 완전한 콘텐츠를 정의. 뉴스 기사, 블로그 글 등 독립적으로 의미를 가지며 재사용이 가능한 콘텐츠를 나타냄

<article>

<h3>글 제목</h3>

<p>글 내용 설명</p>

</article>

* + <aside> : 본문과 관련이 있지만 주요 내용에서 벗어난 콘텐츠를 정의. sidebar나 광고 등등

<aside>

<h4>사이드바 제목</h4>

<p>사이드바 내용 설명</p>

</aside>

* + <footer> : 문서 or 섹션의 하단에 위치. 저작권 정보, 연락처 정보, 관련 문서 링크 등등이 위치

<footer>

<p>&copy; 2023 웹사이트 이름. 모든 권리 보유됨.</p>

</footer>

* + <details> : 사용자가 추가 정보를 원할 때 여닫을 수 있는 세부사항을 정의. FAQ섹션 같은 종류
  + <summary> : <details> 요소의 요약, 제목을 정의.

<details>

<summary>자세한 정보</summary>

<p>추가 세부 정보 내용</p>

</details>

* CSS
  + Float : 주로 이미지와 텍스트를 나란히 배치하거나 다중 열 레이아웃을 만들 때 사용

.float-left {

float: left;

width: 50%;

}

.float-right {

float: right;

width: 50%;

}

* + Flexbox : 부모 요소에 ‘display: flex;’ 속성을 적용하여 자식 요소를 수평, 수직 정렬하는 레이아웃

.container {

display: flex;

justify-content: space-between;

}

* + Grid : 부모 요소에 ‘display: grid;’ 속성을 적용하여 그리드로 요소를 배치하는 레이아웃

.container {

display: grid;

grid-template-columns: 1fr 1fr 1fr;

grid-gap: 10px;

}

* + Layout grid layout : 행과 열의 간격을 설정할 수 있는 등 grid layout보다 더 유연한 레이아웃

.container {

display: grid;

grid-template-columns: 1fr 1fr 1fr;

grid-template-rows: auto;

gap: 10px;

}

1. HTML Box Model

* Content : 박스 속 내용
* Padding : 박스 내부의 여백 지정
* Border : 박스의 테두리
* Margin : 박스 바깥의 여백 지정 => 박스와 주변 요소 사이의 간격

[네가지 모두 사용한 예시]

<!DOCTYPE html>

<html>

<head>

<style>

.box {

width: 200px;

height: 200px;

padding: 20px;

border: 5px solid red;

margin: 30px;

}

</style>

</head>

<body>

<div class="box">

내용(Content) 영역

</div>

</body>

</html>

1. HTML id, class

* Id : 동일 문서 내에 같은 id를 갖는 요소는 하나만 존재해야 함. Css나 js에서 해당 요소를 선택하기 위해 사용

<div id="unique-element">이 요소는 유일한 id를 가집니다.</div>

* Class : 하나의 문서 안에 여러 요소가 동일한 class값을 공유할 수 있음. 동일한 스타일을 여러 요소에 적용하고 싶을 때 주로 사용

<p class="highlight">이 문장은 강조 스타일을 공유하는 클래스를 가집니다.</p>

<p class="highlight">또 다른 문장도 동일한 클래스를 가집니다.</p>

* 둘의 차이점 : ‘id’는 유일한 요소를 식별. ‘class’는 여러 요소에 동일한 스타일이나 동작을 적용하고자 할 때 사용.
  + 유일성 : `id`는 유일해야 하며 한 페이지에서 하나의 요소만 가질 수 있고 `class`는 여러 요소가 동일한 클래스를 공유할 수 있음
  + 적용 범위 : `id`는 한 요소에만 적용되지만, `class`는 여러 요소에 적용할 수 있음
  + 선택자 우선순위 : CSS에서 동일한 스타일 속성을 지정한 경우, `id` 선택자가 `class` 선택자보다 높은 우선순위를 갖음. => `id`로 선택한 요소의 스타일이 `class`로 선택한 요소의 스타일보다 우선적으로 적용
  + JavaScript 접근 : JavaScript에서도 `id`를 사용하여 요소에 접근할 수 있음. `class`로 선택하려면 해당 클래스를 가진 요소 중 하나를 선택해야 함