< CSS 4 >

1학년 반 번 이름

<table>
<tr><th colspan="2">반응형 웹
(웹 요소를 화면 크기에 맞게 재배치하고 각 요소의 표시 방법만 바꾸어 사이트를 구현)</th></tr>
<tr><td>viewport</td><td>
웹 페이지가 접속한 기기에서 보이는 실제 영역의 크기
<meta name="viewport" content="속성1=값1, 속성2=값2…">
<table>
<tr><th>속성 종류</th><th>설명</th><th>사용 가능한 값</th><th>기본값</th></tr>
<tr><td>width</td><td>뷰포트 너비</td><td>device-width/크기</td><td>브라우저 기본값</td></tr>
<tr><td>height</td><td>뷰포트 높이</td><td>device-height/크기</td><td>브라우저 기본값</td></tr>
<tr><td>user-scalable</td><td>확대, 축소 가능 여부</td><td>yes/no</td><td>yes</td></tr>
<tr><td>initial-scale</td><td>초기 확대, 축소 값</td><td>1~10</td><td>1</td></tr>
<tr><td>minimun-scale</td><td>최소 축소비율 설정</td><td>크기</td><td>0.25</td></tr>
<tr><td>maximum-scale</td><td>최대 확대 비율 설정</td><td>크기</td><td>5.0</td></tr>
<tr><td colspan="4">예 <mate name="viewport" content="width=device-width, initial-scale=1"></td></tr>
</table>
<table>
<tr><th colspan="2">단위</th></tr>
<tr><td>vw
(viewport width)</td><td>뷰포트 너비의 1%</td></tr>
<tr><td>vh
(viewport height)</td><td>뷰포트 높이의 1%</td></tr>
<tr><td>vmin
(viewport minimum)</td><td>뷰포트 너비와 높이 중 작은 값의 1%</td></tr>
<tr><td>vmax
(viewport maximum)</td><td>뷰포트 너비와 높이 중 큰 값의 1%</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th colspan="4">미디어 쿼리
(사이트에 접속하는 장치에 따라 특정한 CSS 스타일을 사용하는 방법)</th></tr>
<tr><td rowspan="3">미디어 쿼리 적용하기</td><td rowspan="3">외부</td><td><link></td><td>
<head></head> 태그 사이
<link rel="stylesheet" media="미디어 쿼리 조건" href="css파일 경로">
예. <link rel="stylesheet" media="screen" href="screen.css">
</td></tr>
<tr><td>@import</td><td>
<style></style> 태그 사이
@import url("css파일 경로") 미디어 쿼리 조건
예. @import url("tablet.css") anly screen and (min-width:321px) and (max-width:768px);
</td></tr>
<tr><td colspan="3">
*참고
@import와 자바스크립트가 함께 있는 경우 자바스크립트를 먼저 내려받은 후 import에 있는 css를 다운로드. 자바스크립트에서 스타일과 관련된 정보를 처리하는 경우 오류가 날 수도 있음.
==>CSS 파일이 많고 규모가 큰 사이트를 개발한다면 <link>를 주로 사용함.
</td></tr>
</table>

<table>
<tr><td rowspan="3"></td><td rowspan="3">내부</td><td colspan="2">@media[only | not] 미디어 유형 (조건) and (조건)
only: 미디어 쿼리를 지원하는 기기만 미디어 쿼리를 해석
not: not 다음에 지정하는 미디어 유형을 제외
and: 조건을 여러 개 연결해서 추가할 수 있음</td></tr>
<tr><td>media</td><td><style media="조건">{
스타일 규칙
}
</style>
스타일 태그 안에서 하나의 조건을 지정

예. <style media="screen and (max-width:320px)">
body{
background-color: pink;
}
</style></td></tr>
<tr><td>@media</td><td><style>
@media 조건{
스타일 규칙
}
</style>
스타일 태그 안에 여러 개의 조건 나열, 선택하여 사용 가능

예. <style>
@media screen and (max-width:320px){
body{
background-color: pink;
}
}
</style></td></tr>
<tr><td colspan="3">미디어 유형의 종류</td><td><table><tr><td>all</td><td>모든 미디어 유형에서 사용</td></tr><tr><td>print</td><td>인쇄 장치에서 사용</td></tr><tr><td>screen</td><td>컴퓨터 화면, 스마트 기기에서 사용</td></tr><tr><td>tv</td><td>음성과 영상이 동시 출력되는 tv</td></tr></table></td></tr>
<tr><td colspan="3">뷰포트의 너비와 높이</td><td><table><tr><td>width, height</td><td>너비, 높이 지정</td></tr><tr><td>min-width, min-height</td><td>최소 너비, 최소 높이 지정</td></tr><tr><td>max-width, max-height</td><td>최대 너비, 최대 높이 지정</td></tr></table></td></tr>
<tr><td colspan="3">화면 회전 속성</td><td><table><tr><td>orientation: portrait</td><td>세로 모드</td></tr><tr><td>orientation: landscape</td><td>기본값. 가로 모드</td></tr></table></td></tr>
</table>

그리드 레이아웃

(웹 사이트를 여러 개의 칼럼으로 나누어 웹 요소를 화면에 맞게 배치하는 것)

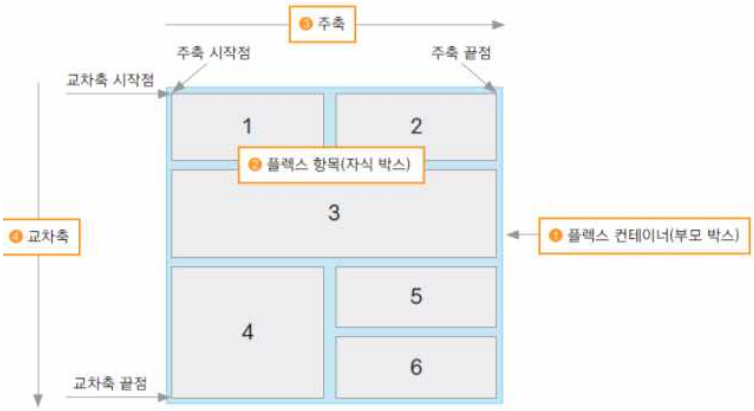
장점: 화면을 규칙적으로 배열하여 레이아웃 일관성을 유지, 시각적으로 안정된 디자인, 업데이트가 쉬움, 요소를 자유롭게 배치 가능

방법: CSS의 float속성 활용, 플렉서블 박스 레이아웃, CSS그리드 레이아웃

플렉스 박스 레이아웃

그리드 레이아웃을 기본으로 각 박스를 원하는 위치에 따라 배치하는 것(=플렉서블 박스 레이아웃)

수평 방향이나 수직 방향 중에서 한쪽을 주축으로 정하고 박스를 배치



❶ **플렉스 컨테이너(부모 박스):** 플렉스 박스 레이아웃을 적용할 대상을 묶는 요소입니다.

❷ **플렉스 항목(자식 박스):** 플렉스 박스 레이아웃을 적용할 대상으로 그림 12-18에서 1~6까지 작은 박스들이 모두 해당합니다.

❸ **주축(main axis):** 플렉스 컨테이너 안에서 플렉스 항목을 배치하는 기본 방향입니다. 기본적으로 왼쪽에서 오른쪽이며 수평 방향으로 배치합니다. 플렉스 항목의 배치가 시작되는 위치를 '주축 시작점', 끝나는 위치를 '주축 끝점'이라고 합니다.

❹ **교차축(cross axis):** 주축과 교차하는 방향을 말하며 기본적으로 위에서 아래로 배치합니다. 플렉스 항목의 배치가 시작되는 위치를 '교차축 시작점', 끝나는 위치를 '교차축 끝점'이라고 합니다.

display

플렉스 컨테이너(부모 박스)를 지정

flex	블록 레벨 요소로 배치
inline-flex	인라인 레벨 요소로 베치

flex-direction

플렉스 항목(자식 박스)을 배치하는 주축과 방향을 지정

row	기본값, 주축을 가로로 지정, 왼쪽에서 오른쪽으로 배치
row-reverse	주축을 가로로 지정, 오른쪽에서 왼쪽으로 배치
column	주축을 세로로 지정, 위에서 아래로 배치
column-reverse	주축을 세로로 지정, 아래에서 위로 배치

flex-wrap

플렉스 항목의 줄을 바꿈

(플렉스 컨테이너 너비보다 많은 플렉스 항목이 있는 경우 줄을 바꿀지 여부를 지정)

nowrap	기본값, 항목을 한 줄에 표시
wrap	여러 줄에 표시
wrap-reverse	여러 줄에 표시하되, 시작점과 끝점이 바뀜

flex-flow

flex-direction, flex-wrap 속성을 한꺼번에 지정하여 배치 방향, 줄 바꿈 결정

기본값: row nowrap

예> flex-flow: row wrap;

justify-content

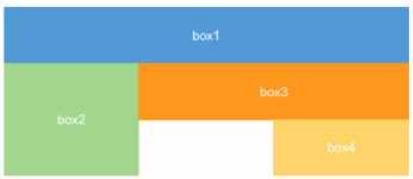
주축 정렬 방법 지정

flex-start	주축의 시작점에 맞춰 배치
flex-end	주축의 끝점에 맞춰 배치
center	주축의 중앙에 맞춰 배치

<table>
<tr><td rowspan="4"></td><td></td><td>
<table>
<tr><td>space-between</td><td>첫 번째 항목과 끝 항목을 주축의 시작점과 끝점에 배치한 후 나머지 항목은 같은 간격으로 배치</td></tr>
<tr><td>space-around</td><td>모든 항목의 둘레(around)가 균일하도록 배치</td></tr>
<tr><td>space-evenly</td><td>모든 항목 사이와 양끝의 간격이 균일하도록 배치</td></tr>
</table>
</td></tr>
<tr><td>align-items</td><td>교차축 정렬 방법 지정
<table>
<tr><td>flex-start</td><td>교차축의 시작점에 맞춰 배치</td></tr>
<tr><td>flex-end</td><td>교차축의 끝점에 맞춰 배치</td></tr>
<tr><td>center</td><td>교차축의 중앙에 배치</td></tr>
<tr><td>baseline</td><td>교차축의 문자 기준선에 맞춰 배치</td></tr>
<tr><td>stretch</td><td>플렉스 항목을 늘려 교차축에 가득 차게 배치, 기본값</td></tr>
</table>
</td></tr>
<tr><td>align-self</td><td>특정 항목만 정렬 방법을 지정
(align-items는 플렉스 컨테이너를 지정하는 선택자에 사용,
align-self는 플렉스 항목 선택자에 사용)
속성은 align-items와 동일</td></tr>
<tr><td>align-content</td><td>플렉스 항목이 여러 줄일 때 교차축 정렬 방법을 지정
<table>
<tr><td>flex-start</td><td>교차축의 시작점에 맞춰 배치</td></tr>
<tr><td>flex-end</td><td>교차축의 끝점에 맞춰 배치</td></tr>
<tr><td>center</td><td>교차축의 중앙에 맞춰 배치</td></tr>
<tr><td>space-between</td><td>첫 번째 항목과 끝 항목을 교차축의 시작점과 끝점에 배치한 후 나머지 항목은 같은 간격으로 배치</td></tr>
<tr><td>space-around</td><td>모든 항목을 교차축에 같은 간격으로 배치</td></tr>
<tr><td>stretch</td><td>플렉스 항목을 늘려 교차축에 가득 차게 배치, 기본값</td></tr>
</table>
</td></tr>
<tr><td colspan="3">플렉스 박스 레이아웃: 플렉스 항목을 배치할 때 가로나 세로 중 하나를 주축으로 정하고 배치, 1차원
CSS 그리드 레이아웃: 가로와 세로를 모두 사용, 2차원</td></tr>
<tr><td rowspan="5">CSS
그리드
레이아웃</td><td colspan="2">줄과 칼럼으로 화면을 구성, 줄과 칼럼 사이의 여백을 조절

</td></tr>
<tr><td>display</td><td>그리드 컨테이너 지정
<table>
<tr><td>grid</td><td>컨테이너 안의 항목을 블록 레벨 요소로 배치</td></tr>
<tr><td>inline-grid</td><td>컨테이너 안의 항목을 인라인 레벨 요소로 배치</td></tr>
</table>
</td></tr>
<tr><td>grid-template-columns
grid-template rows</td><td>칼럼과 줄의 크기와 개수 지정</td></tr>
<tr><td>fr
(fraction)</td><td>상대적인 크기를 지정하는 단위</td></tr>
<tr><td>repeat()</td><td>값이 반복될 때 줄여서 표현
예> grid-template-columns: 1fr 1fr 1fr;
⇨ grid-template-columns: repeat(3,1fr);</td></tr>
</table>

<table>
<tr><td rowspan="4"></td><td>minmax()</td><td>최솟값과 최댓값 지정
예> grid-template-rows: minmax(100px,auto);
/*줄 높이 최소 100px, 자동*/</td></tr>
<tr><td>auto-fill
auto-fit</td><td>화면 너비에 따라 자동으로 칼럼 개수 조절
auto-fit: 남는 공간 없이 가득 채움
auto-fill: 칼럼의 최소 너비만 표시하고 남는 공간은 비워둠</td></tr>
<tr><td>grid-gap</td><td>그리드 항목의 간격 지정
<table>
<tr><td>grid-column-gap</td><td>칼럼과 칼럼 사이의 간격 지정</td></tr>
<tr><td>grid-row-gap</td><td>줄과 줄 사이의 간격 지정</td></tr>
<tr><td>grid-gap</td><td>칼럼과 줄 사이의 간격을 한꺼번에 지정
grid-gap: 줄 간격, 칼럼 간격</td></tr>
</table></td></tr>
<tr><td>그리드 라인</td><td>

<table>
<tr><th>종류</th><th>설명</th><th>예시</th></tr>
<tr><td>grid-column-start</td><td>칼럼 시작 라인 번호 지정</td><td>1</td></tr>
<tr><td>grid-column-end</td><td>칼럼 마지막 라인 번호 지정</td><td>4</td></tr>
<tr><td>grid-column</td><td>칼럼 시작, 끝 번호 사이에 /</td><td>1/4</td></tr>
<tr><td>grid-row-start</td><td>줄 시작 라인 번호 지정</td><td>2</td></tr>
<tr><td>grid-row-end</td><td>줄 마지막 라인 번호</td><td>4</td></tr>
<tr><td>grid-row</td><td>줄 시작, 끝 번호 사이에 /</td><td>2/4</td></tr>
</table></td></tr>
<tr><td>템플릿 영역</td><td>
<table>
<tr><td>grid-area</td><td>템플릿 이름 지정</td></tr>
<tr><td>grid-template-areas</td><td>템플릿 영역 배치 방법 지정
예> grid-template-areas:
“box1 box1 box1”
/*한줄에 들어갈 영역은 큰 따옴표(“”)로 묶음*/
“box2 box3 box3”
“box2 . box4”; /*빈 영역은 마침표(.)*/
 
</td></tr>
</table></td></tr>
</table>