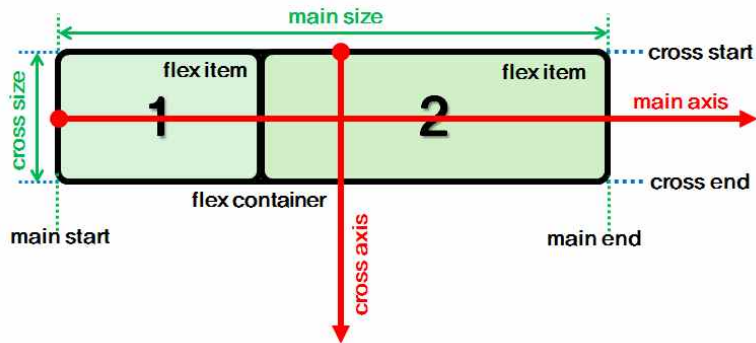


8강. CSS3 [5] : 고급 레이아웃

1. 유연한 박스 레이아웃

1.1 유연한 박스 레이아웃

- 화면과 브라우저 창의 크기가 변경되더라도 요소의 상대적 위치 및 크기를 일정하게 유지하는데 유용하기 때문에 복잡한 웹 문서의 레이아웃에 사용된다.
- CSS3 에서 새롭게 추가된 "display: flex | inline-flex"에 의해 계산된 요소에 의해 생성된 박스를 플렉스 컨테이너(flex container)라고 한다. 플렉스 컨테이너의 자식들을 플렉스 항목(flex item)이라 부르며, 유연한 레이아웃(flex layout) 모델을 사용하여 배치하게 된다.



1.2 유연한 박스의 순서 및 방향 관련 속성

- 플렉스 컨테이너의 내용은 순서에 관계없이 어떤 방향이든 배치할 수 있다.

1.2.1 flex-direction 속성 : 유연한 박스의 배치 방법 지정

- 플렉스 컨테이너의 기본 축(main axis) 방향을 설정함으로써, 플렉스 항목이 플렉스 컨테이너에 배치하는 방법을 지정하는 속성이다.

- 속성값 → row (기본) | row-reverse | column | column-reverse
 - 기본 축 → 현재 쓰기 모드의 인라인 축
 - 기본 축 → 현재 쓰기 모드의 블록 축

```

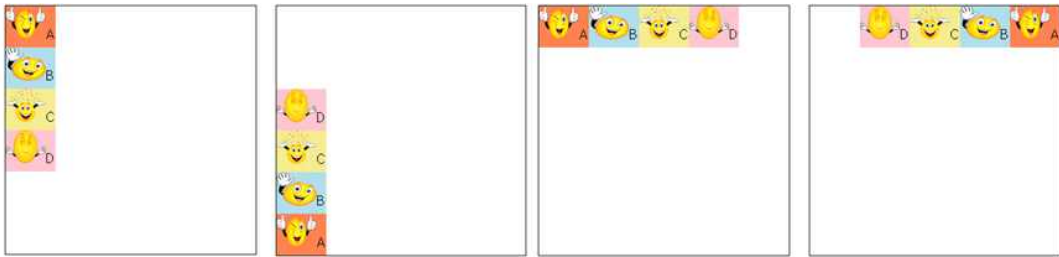
예제
<!DOCTYPE html> <html><head>
<style>
#main {
width: 300px; height: 300px; border: 1px solid #000;
display: flex;
flex-direction: column;

```

```

}
#main div { width: 60px; height: 50px; }
img { width: 80%; height: 80% }
</style>
</head>
<body>
<div id="main">
  <div >A</div>
  <div >B</div>
  <div >C</div>
  <div >D</div>
</div>
</body>
</html>

```



flex-direction : column; flex-direction : column-reverse ;

flex-direction : row; flex-direction : row-reverse ;

1.2.2 flex-wrap 속성

- 플렉스 컨테이너를 한 줄 또는 여러 줄에 배치할 것인지 제어, 새로운 줄이 교차 축에 어떤 방향으로 진행되는지를 지정하는 속성이다.

▪ 속성값 → nowrap (기본) | wrap | wrap-reverse

한 줄에 배치 ← → 여러 줄에 배치

예제

```

<!DOCTYPE html> <html> <head>
<style>
#main { width: 200px; height: 200px; border: 1px solid #000;
display: flex;
flex-wrap: wrap-reverse;
}
#main div { width: 70px; height: 50px;}
img { width: 70%; height: 80% }
</style>
</head>

```

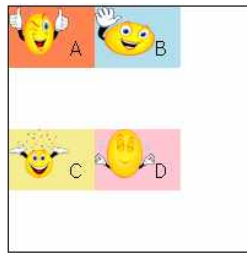
```

<body>
<div id="main">
  <div>A</div>
  <div>B</div>
  <div>C</div>
  <div>D</div>
</div>
</body>
</html>

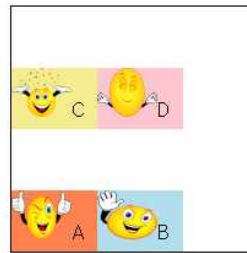
```



flex-wrap: nowrap;



flex-wrap: wrap;



flex-wrap: wrap-reverse;

1.2.3 flex-flow 속성

- flex-direction 속성과 flex-wrap 속성의 일괄 지정할 수 있는 속성으로 플렉스 컨테이너의 기본 축과 교차 축을 정의하는 속성이다.

1.2.4 order 속성

- 플렉스 항목들은 기본적으로 문서에 요소들이 표시되는 순서와 동일하게 배치하게 되는데 order 속성은 항목들이 배치되는 순서를 지정하는 속성이다.
- 속성에서 지정한 숫자 중에서 가장 낮은 번호의 요소가 가장 먼저 배치된다.

1.3 유연한 박스의 유연성 관련 속성

1.3.1 flex-grow 속성

- 플렉스 항목의 폭의 확대 비율을 지정하여 플렉스 컨테이너 폭에 맞게 자동으로 크기를 조절해주는 속성이다.
- 속성 값이 지정되어 있는 경우에는 값에 따라 플렉스 컨테이너의 여유 공간 내에서 폭에 맞도록 자동으로 크기를 조절된다. 음수 값을 사용할 수 없으며, 생략된 경우에는 1로 설정된다.

1.3.2 flex-shrink 속성

- 플렉스 항목의 폭의 축소 비율을 지정하여 자동으로 크기를 조절해주는 속성이다.

1.3.3 flex-basis 속성

- 플렉스 컨테이너 내에 배치되는 플렉스 항목의 기준 폭을 지정하는 속성이다.

- 속성값 → auto | <width>
↳ 기준 폭이 플렉스 항목의 기본 크기와 동일

1.3.4 flex 속성


- flex-grow, flex-shrink, flex-basis 속성에 대한 일괄 지정할 수 있는 속성이다.
- 속성값 → none | [<flex-grow> <flex-shrink>? || <flex-basis>]

예제

```

<!DOCTYPE html> <html><head>
<style>
#main { width: 300px; height: 50px; border: 1px solid #000;
  display: flex;
}
#main div { flex: 1 }
div#myRedDIV { background-color:coral; }
div#myBlueDIV { background-color:lightblue; }
div#myGreenDIV { background-color:lightgreen; }
div#myPinkDIV { background-color:pink; }
img { width: 70%; height: 80% }
</style>
</head>
<body>
<div id="main">
  <div id="myRedDIV">A</div>
  <div id="myBlueDIV">B</div>
  <div id="myGreenDIV">C</div>
  <div id="myPinkDIV">D</div>
</div>
</body>
</html>

```



1.4 유연한 박스의 정렬 및 교차 축 지정

1.4.1 justify-content 속성

- 플렉스 컨테이너 안에서 현재 줄의 기본 축을 따라 플렉스 항목의 정렬 방법을 지정하는 속성이다.
- 속성값 → flex-start (기본) | flex-end | center | space-between | space-around

1.4.2 align-content 속성

- 플렉스 컨테이너 안에서 교차 축의 여분의 공간이 있을 때, 플렉스 컨테이너 줄의 정렬 방식을 지정하는 속성이다.

▪ 속성값 → `flex-start` | `flex-end` | `center` | `space-between` | `space-around` | `stretch` (기본)

1.4.3 align-items 속성

- 항목을 세로 방향으로 정렬할 때, 전체 항목을 동일한 기준 축으로 지정하는 속성이다.

▪ 속성값 → `stretch` (기본) | `flex-start` | `flex-end` | `center` | `baseline`

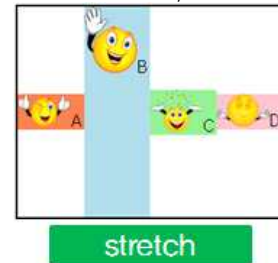
1.4.4 align-self 속성

- 플렉스 항목마다 정렬 기준 축을 별도로 지정하고자 할 때 사용하는 속성이다.

▪ 속성값 → `auto` (기본) | `flex-start` | `flex-end` | `center` | `baseline` | `stretch`

예제

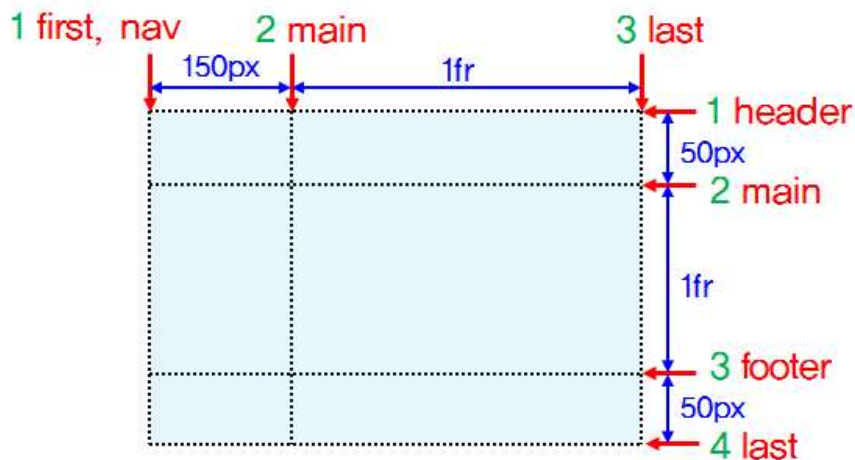
```
<!DOCTYPE html> <html> <head>
<style>
div#main { width: 250px; height: 200px; border: 2px solid #000000;
display: flex;
align-items: center
}
div#main div { flex: 1}
img { width: 80%; height: 80% }
div#myRedDIV { background-color:coral; }
div#myBlueDIV { background-color:lightblue; align-self: stretch }
div#myGreenDIV { background-color:lightgreen; }
div#myPinkDIV { background-color:pink; }
</style>
</head>
<body>
<div id="main">
<div id="myRedDIV">A</div>
<div id="myBlueDIV">B</div>
<div id="myGreenDIV">C</div>
<div id="myPinkDIV">D</div>
</div>
</body>
</html>
```



2. 그리드 레이아웃

2.1 개념 및 용어

- 그리드 라인(Line)은 그리드의 가로 및 세로의 분할 선을 나타내는 것으로, 그리드 컨테이너의 공간을 그리드 영역으로 분할하는 역할을 하고, 그리드 항목이 이러한 그리드 라인에 위치하고 정렬되어 배치하게 된다.
- 그리드 트랙(track)은 그리드 열과 행에 대한 또 다른 이름의 일반적인 용어로서, 두 개의 인접하는 그리드 라인 사이의 공간을 의미하며 지정한 요소들이 위치하게 될 가상의 그리드를 의미한다.
- 그리드 셀(Cell)은 두 개의 인접한 행이나 열의 그리드 라인 사이의 공간으로써, 그리드 항목을 위치할 때 참조할 수 있는 최소 단위를 나타낸다.
- 그리드 영역(Areas)은 하나 이상의 그리드 항목을 배치하는데 사용되는 논리적인 공간을 의미한다.



2.2 트랙(행/열) 크기 지정

- grid-template-columns 속성은 각 그리드 열의 폭을 지정하고, grid-template-rows 속성은 각 그리드 행의 높이를 지정한다. 열의 폭 및 행의 높이를 지정할 때, 각각의 열 및 행은 공백으로 구분하고 지정한다.
- 속성값 → <length>, <%>, <flex>, max-content, min-content, minmax(min, max), auto, none, subgrid

grid-template-columns: 100px 1fr max-content minmax(min-content, 1fr)

grid-template-rows: 1fr minmax(min-content, 1fr)

grid-template-rows: 10px repeat(2, 1fr auto minmax(30%, 1fr))

grid-template-rows: calc(4em - 5px)

4em에서 5px 뺀 만큼의 크기

내부 표현식에는 +, -, *, / 등의 사칙 연산이 가능
(+ 또는 -는 반드시 기호 양쪽으로 공백 삽입)

2.2.1 <flex> fr 단위

- 사용 가능한 공간을 분수 값에 따라 열 또는 행 사이에 나누어야 함을 나타내는 단위이다.

(열 또는 행의 <flex> 자유 공간)
(모든 flex 계수들의 합)

2.2.2 repeat() 표기

- 동일하거나 반복 패턴을 보이는 많은 열이나 행이 있을 경우, 간결한 형식으로 열이나 행을 정의할 수 있다.

repeat(반복횟수, [<라인이름>? <트랙크기>]+ <라인이름>?)

grid-template-columns: 10px 250px 10px 250px 10px 250px 10px 250px 10px 5px;

↓
repeat(4, 250px 10px)

grid-template-rows: repeat(2, (a) 1fr (b));

↓
(a) 1fr (b) (a) 1fr (b)

↓
(a) 1fr (b a) 1fr (b)

2.2.3 subgrid 키워드

- 부모 그리드 컨테이너를 기준으로 그리드의 행 또는 열의 크기를 조정할 수 있도록 하며 subgrid로 지정된 그리드 항목들의 콘텐츠를 정렬할 때 사용한다.

2.2.4 grid-template-areas 속성

- 특정 그리드 항목과 관련되지 않은 그리드 영역에 이름을 지정하는 속성이다.

2.2.5 grid-template 속성

- grid-template-columns, grid-template-rows, grid-template-areas 속성을 일괄 지정하는 속성이다.

- `<grid-template-columns> / <grid-template-rows>`
`grid-template: auto 1fr auto / auto 1fr` → `grid-template-columns: auto 1fr auto;`
`grid-template-rows: auto 1fr;`
`grid-template-areas: none`
- `[<트랙크기> /]? [<라인이름>? <string> <트랙크기>? <라인이름>?]+`
`grid-template: auto 1fr auto /`
`(h-top) "a a a" auto (h-bottom)`
`(m-top) "b b b" 1fr (m-bottom)` → `grid-template-columns: auto 1fr auto;`
`grid-template-rows: (h-top) auto`
`(h-bottom m-top) 1fr (m-bottom);`
`grid-template-areas: "a a a"`
`"b b b";`

2.2.6 grid-auto-columns, grid-auto-rows 속성

- grid-template-columns/-rows 속성에서 크기가 명시적으로 표기되지 않은 경우 그리드 크기를 지정할 수 있다.

2.2.7 grid-auto-flow 속성

- 그리드 항목을 자동으로 배치할 수 있도록 지정 하는 속성이다.
- 속성값 → `row | column` (항목 배치를 위해 필요한 경우 자동으로 행/열을 추가)

2.2.8 그리드 레이아웃 속성 일괄 지정

- grid 속성은 명시적인 속성들뿐만 아니라, 암시적인 속성들까지 동시에 모두 설정하는 속성이다.
- 속성값 → `<grid-template> [<grid-auto-flow> [<grid-auto-columns> [/ <grid-auto-rows>]?]?]`

2.3 그리드 항목 배치

2.3.1 그리드 라인을 기준으로 한 배치

- grid-row-start, grid-column-start, grid-row-end, grid-column-end 속성은 그리드 라인과 확장 또는 자동 배치를 통한 그리드 항목의 크기와 위치를 결정한다.

2.3.2 단축 속성에 의한 배치

- grid-row속성, grid-column속성은 grid-row-start/grid-row-end 속성과 grid-column-start / grid-column-end 속성의 일괄 지정할 수 있는 속성이다.
- 속성값 → `<grid-row-start> [/ <grid-row-end>]?, <grid-column-start> [/ <grid-column-end>]?`
- grid-area 속성은 4개의 속성(grid-row-start, grid-row-end, grid-column-start, grid-column-end)의 일괄 지정할 수 있는 속성이다.
- 속성값 → `<grid-row-start> / <grid-column-start> / <grid-row-end> / <grid-column-end>`

2.3.3 절대 위치에 의한 배치

- 절대 위치에 대한 오프셋 속성(top/right/bottom/left)은 일반적으로 포함 블록의 경계로부터 안쪽 방향으로의 오프셋을 의미한다.

2.4 그리드 정렬 및 겹침 우선 순서

- 기본적으로 그리드 항목들은 그리드 영역을 채우기 위해 늘어난다(stretch). 만일 행을 중심으로 하는 정렬을 하고자 할 때는 그리드 컨테이너의 justify-items 속성이나 그리드 항목의 justify-self속성을 사용하여 그리드 항목을 인라인 크기 안에서 정렬시킬 수 있다. 그리고 열을 중심으로 하는 정렬을 하고자 할 때는 그리드 항목의 align-self 속성이나 그리드 컨테이너의 align-items 속성을 사용하여 그리드 항목을 블록 크기 안에서 정렬시킬 수 있다. 또한 그리드의 외부 경계가 그리드 컨테이너의 콘텐츠 경계와 일치하지 않는다면 그리드는 justify-content 속성과 align-content 속성에 따라서 콘텐츠 박스 내에서 정렬된다.
- z-index 속성은 그리드 셀 내의 그리드 항목의 계층을 지정한다.

3. 사용자 인터페이스

3.1 content 속성

- ::before 및 ::after를 이용하여 특정 요소의 앞이나 뒤에 콘텐츠를 추가할 수 있도록 지정하는 속성이다.

예제	
<pre> <!DOCTYPE html> <html> <head> <style> p::before{content:open-quote;} p::after{content:close-quote;} p.address::before{content:no-open-quote;} p.address::after{content:no-close-quote;} </style> </head> <body> <p>한국방송통신대학교</p> <p class="address">서울특별시 종로구 대학로86</p> </body> </html> </pre>	

3.2 box-sizing 속성

- 해당 요소의 박스 크기를 결정하는 방식을 지정하는 방식이다.

3.3 박스의 윤곽선 지정

속성	설명
outline-color	윤곽선의 색상 지정 <color> invert → 스크린 상에서의 색을 역으로 만들.(테두리가 어떤 색상 배경이던 관계없이 확실히 보이게 함)
outline-style	윤곽선의 모양(스타일) 지정 auto "border-style 속성의 값" → none, dotted, dashed, solid, double, ridge, groove, inset, outset
outline-width	윤곽선의 굵기 지정 "border-width 속성의 값" → <length>, thin, medium, thick
outline	윤곽선 속성(outline-*)의 일괄 지정 <outline-color> <outline-style> <outline-width>
outline-offset	요소 박스의 외부 영역과 윤곽선 사이의 간격을 지정 <length>

- outline-offset 속성은 요소 박스의 외부 영역과 윤곽선 사이의 간격을 지정한다. 속성 값이 0보다 크면 테두리 가장 자리로부터 시작하여 윤곽선을 표시한다. 만일, 음수 값을 지정하면 윤곽선이 요소의 안쪽으로 그려진다.

3.4 resize 속성

- 사용자 하여금 요소 박스의 크기 조정 여부를 지정하는 속성이다.
 - 속성값 → none | both | horizontal | vertical
 - "overflow: auto | hidden | scroll"과 함께 사용



3.5 text-overflow 속성

- 요소 박스 내의 텍스트가 박스 영역에서 벗어났을 때의 표시 방식 지정한다.
 - 속성값 → dip (기본) | ellipsis

3.6 cursor 속성

- 포인팅 장치가 요소의 테두리 및 패딩, 콘텐츠 위에 있을 때 표시될 커서의 종류를 지

정하는 속성이다.

3.7 요소의 탐색 순서 지정

- nav-index 속성은 순차 탐색 순서(탭 이동 순서)를 결정하는 입력 방법을 지정하는 속성이다.
- nav-up, nav-right, nav-down, nav-left 속성은 키보드의 각 방향 키에 대해 포커스를 이동하여 반응하는 순서를 지정하는 속성으로, nav-up은 위쪽 방향, nav-right은 오른쪽 방향, nav-down은 아래쪽 방향, nav-left은 왼쪽 방향을 지정하는 속성이다.

3.8 ime-mode 속성

- 텍스트 필드에 대한 입력기(input method editor)의 상태를 제어하는 속성이다.
 - 속성값 → auto (기본) | normal | active | inactive | disabled