

□ 개념 확인

(1) 플렉스 박스 컨테이너에 적용하는 속성에 대하여 설명하시오.

-컨테이너에 적용하는 속성:

1.flex-direction: flex 컨테이너의 주축(main axis) 방향설정

1)row :좌에서 우로 수평 배치, flex-direction 속성의 기본값

2) row-reverse : 우에서 좌로 수평 배치

3)column: 위에서 아래로 수직 배치

4)column-reverse:아래에서 위로 수직배치

2.flex-wrap:flex 컨테이너의 width 보다 flex item들의 width의 합계가 더 큰 경우 표현 방식 설정

1)nowrap: flex item을 줄 바꿈 없이 한 행에 배치, 기본값

2)wrap : flex item들의 width의 합계가 flex 컨테이너의 width보다 큰 경우 여러 행에 배치, 기본적으로 좌에서 우로,위에서 아래로 배치

3)wrap-reverse: 속성값 wrap과 동일하나 아래에서 위로 배치

3.flex-flow:flex-direction 속성과 flex-wrap 속성을 동시에 설정, 기본값은 row nowrap

1)flex-flow:<flex-direction> || <flex-wrap>;

4.justify-content:플렉스 컨테이너의 중심축을 기준으로 flex item을 수평 정렬

1)flex-start: main start(좌측)를 기준으로 정렬, 기본값.

2)flex-end: main end(우측)를 기준으로 정렬

3)center: 플렉스 컨테이너 중앙에 정렬

4)space-between: 첫번째와 마지막 flex item은 좌우 측면에 정렬되고 나머지는 균등한 간격으로 정렬

5)space-around: 모든 플렉스 아이템이 균등한 간격으로 정렬

6)space-evenly: 균등 정렬

5.align-items: flex item을 flex container의 수직 방향(cross axis)으로 정렬

1)stretch:모든 flex item은 flex container의 높이에 꼭 찬 높이를 갖는다.

2)flex-start: flex container의 cross start 기준으로 정렬

3)flex-end: flex container 의 cross end 기준으로 정렬

4)center: flex container의 cross axis의 중앙에 정렬

5)baseline: flex container 의 텍스트 baseline을 기준으로 정렬

6.align-content:

1)stretch:모든 flex item은 flex item의 행 이후에 균등하게 분배된 공간에 정렬되어 배치. 기본값

2)flex-start: flex container의 cross start 기준으로 정렬

3)flex-end: flex container 의 cross end 기준으로 정렬

4)center: flex container의 cross axis의 중앙에 정렬

5)space-between: 첫번째와 마지막 flex item은 좌우 측면에 정렬되고 나머지는 균등 분할된 공간내에 배치 정렬

6)space-around: 모든 플렉스 아이템이 균등 분할된 공간내에 배치 정렬

(2) 플렉스 박스 아이템에 적용하는 속성에 대하여 설명하시오.

-아이템에 적용하는 속성:

1.flex grow : 플렉스 박스에 여백이 있을 때 적용

2.flex-shrink: 플렉스 박스 안의 flex item의 크기가 넘칠 경우

3.flex-basis: flex item의 (공간 배분 전) 너비 기본값을 px,% 등의 단위로 지정, 기본값은 auto

4.align-self: align-items 속성보다 우선하여 개별 flex item 정렬, 기본값은 auto

5.order: flex item의 배치 순서 지정

(3) 미디어 쿼리를 적용하는 두가지 방법을 설명하시오.

1) @-규칙 - 스타일시트 내부에서 특정한 규칙을 표현하는 데 사용

@media(<미디어 쿼리>){ //css코드 }

2) media 속성 -link 태그에 입력해서 해당 미디어 쿼리 조건에 맞는 장치에서 만CSS 파일을 불러올 때 사용

```
<link rel="stylesheet" href="<파일 이름>" media="<미디어 쿼리>">
```

(b) media 속성

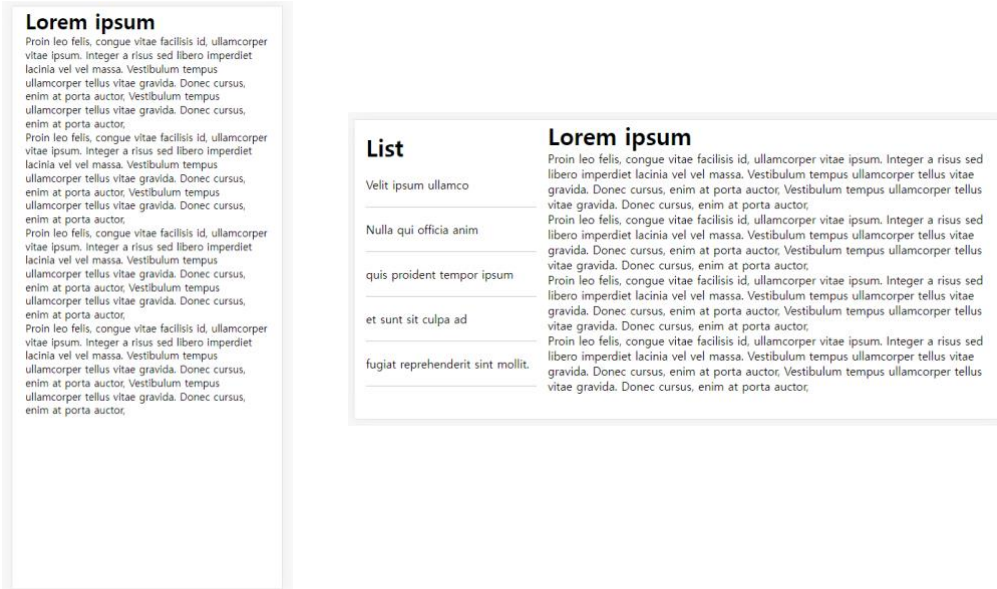
(4) 다음과 같은 One True 레이아웃을 구성하는 과정에 대하여 설명하시오.



중간 에있는 갈색 div 와 하늘색 div 부분을 감싸는 div 에 flex 를 적용시키고 갈색 div 와 하늘색 div 부분의 여백을 다르게 설정해주면 된다.

그리고 뷰포트 단위를 이용해 일정한 비율로 크기를 맞추어 준다.

- (5) 스마트폰 회전을 사용한 레이아웃을 만들어 보시오. 아이폰의 기본 메시지 앱은 다음과 같이 세로 상태에서는 메시지 목록과 내용이 별도의 페이지로 되어있다. 가로 상태에서는 메 시지 목록과 내용을 한 화면에 표시할 수 있도록 css 코드를 작성하시오. 힌트)플렉스 박스와 미디어 쿼리 사용



다음 순서로 레이아웃을 구성하시오. 제시된 레이아웃 외에는 본인이 작성

초기화	<ul style="list-style-type: none">① 모든 태그(*)의 margin 속성과 padding 속성을 0 으로 지정합니다.② li 태그의 list-style-type 을 none 으로 지정합니다.
반응형 웹	<ul style="list-style-type: none">① 세로 상태에서는 div.left 태그가 보이지 않게 만듭니다.② 가로 상태에서는 div.left 태그가 너비의 30%를 차지하고, 왼쪽에 붙게 만듭니다.③ 가로 상태에서는 div.right 태그가 너비의 70%를 차지하고 오른쪽에 붙게 만듭니다.
기본 콘텐츠 구성	<ul style="list-style-type: none">① div.content 태그의 padding 을 16px 로 지정합니다.② li 태그의 위아래 padding 속성을 24px 로 지정합니다.③ li 태그의 아래쪽 테두리를 1px solid #c7c7c7 로 지정합니다.

<body>

<div class="container">

<div class="left">

<div class="content">

<h1>List</h1>

Velit ipsum ullamco

Nulla qui officia anim

quis proident tempor ipsum

et sunt sit culpa ad

```

        <li>fugiat reprehenderit sint mollit.</li>
    </ul>
</div>
</div>
<div class="right">
    <div class="content">
        <h1>Lorem ipsum</h1>
        <p>Proin leo felis, congue vitae facilisis id, ullamcorper vitae ipsum. Integer a risus sed libero imperdiet
        lacinia vel vel massa. Vestibulum tempus ullamcorper tellus vitae gravida. Donec cursus, enim at porta auctor,
        Vestibulum tempus ullamcorper tellus vitae gravida. Donec cursus, enim at porta auctor,
        </p>
        <p>Proin leo felis, congue vitae facilisis id, ullamcorper vitae ipsum. Integer a risus sed libero imperdiet
        lacinia vel vel massa. Vestibulum tempus ullamcorper tellus vitae gravida. Donec cursus, enim at porta auctor,
        Vestibulum tempus ullamcorper tellus vitae gravida. Donec cursus, enim at porta auctor,
        </p>
        <p>Proin leo felis, congue vitae facilisis id, ullamcorper vitae ipsum. Integer a risus sed libero imperdiet
        lacinia vel vel massa. Vestibulum tempus ullamcorper tellus vitae gravida. Donec cursus, enim at porta auctor,
        Vestibulum tempus ullamcorper tellus vitae gravida. Donec cursus, enim at porta auctor,
        </p>
        <p>Proin leo felis, congue vitae facilisis id, ullamcorper vitae ipsum. Integer a risus sed libero imperdiet
        lacinia vel vel massa. Vestibulum tempus ullamcorper tellus vitae gravida. Donec cursus, enim at porta auctor,
        Vestibulum tempus ullamcorper tellus vitae gravida. Donec cursus, enim at porta auctor,
        </p>
    </div>
</div>
</div>
</body>

```

[소스 - css만 제시 하세요]

```

<style>
    * {
        margin: 0px;
        padding: 0px;
    }
    li {
        list-style-type: none;
        padding: 24px 0px 24px 0px;
        border-bottom: 1px solid #c7c7c7;
    }
    .content {
        padding: 16px;
    }
</style>

<style>

```

```
@media screen and (orientation: portrait){
    .left {
        display: none;
    }
}

@media screen and (orientation: landscape) {
    .container{
        display: flex;
    }

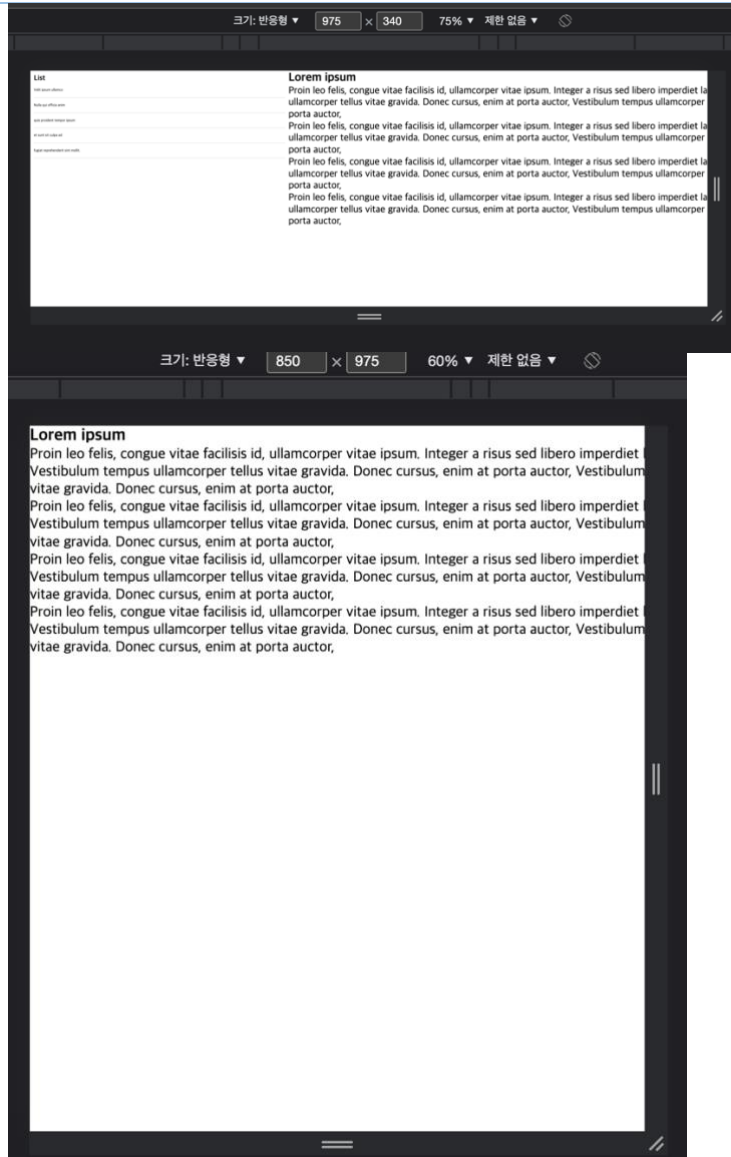
    .left {
        width: 30%;
    }

    .content{
        /* height: 100%; */
        /* font-size: 200%; */
    }

    .right {
        width: 70%;
    }
}

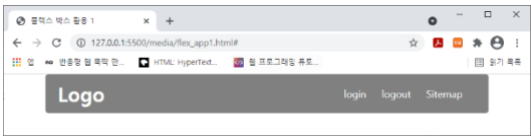
</style>
```

[실행 결과]



(6) 플렉스 박스를 사용하여 다음과 같은 레이아웃을 구성할 수 있도록 css 를 작성하시오.

```
<body>
  <header>
    <h1> <a href="#">Logo</a> </h1>
    <nav>
      <ul>
        <li> <a href="#">login</a> </li>
        <li> <a href="#">logout</a> </li>
        <li> <a href="#">Sitemap</a> </li>
      </ul>
    </nav>
  </header>
</body>
```



[소스 – css만 제시하시오]

```
<style>

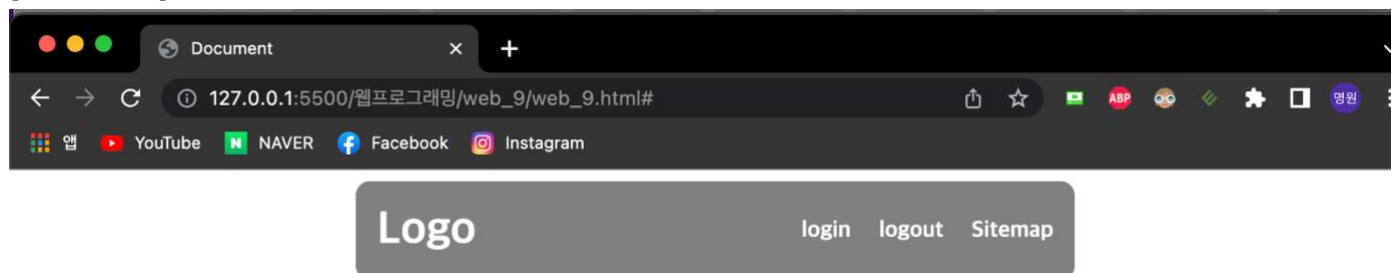
  header {

    width: 500px;
    height: 60px;
    margin: auto;
    padding: 5px;
    display: flex;
    justify-content: space-between;
    align-items: center;
    background-color: gray;
    border-radius: 10px;

  }
  ul {
    display: flex;
    list-style-type: none;
  }
  h1,li{
    padding: 10px;
  }
  a {
    text-decoration: none;
    font-weight: bold;
    color: white;
  }

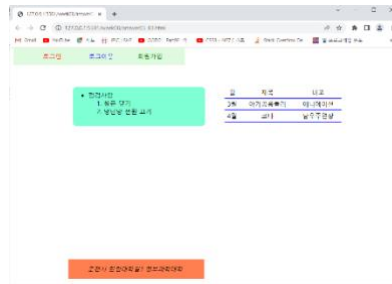
</style>
```

[실행 결과]



□ 개념 활용 응용 프로그래밍

(1) 제시된 결과처럼 출력될 수 있도록 주어진 html에 대하여 css코드를 작성하시오. 단, 요소 선택 시 클래스 애트리뷰트가 있으면 클래스 선택자를, 아이디 애트리뷰트가 있으면 아이디 선택자를 사용한다.



```
<body>
```

```
<!-- [문제 3-1번] -->
```

```
<nav>
```

```
  <ul class="menu">
```

```
    <li>로그인</li>
```

```
    <li>로그아웃</li>
```

```
    <li>회원가입</li>
```

```
  </ul>
```

```
</nav>
```

```
<!-- [문제 3-2번] -->
```

```
<section>
```

```
<!-- [문제 3-3번] -->
```

```
<div>
```

```
  <table>
```

```
    <tr><td>월</td>
```

```
    <td>제목</td>
```

```
    <td>비고</td></tr>
```

```
    <tr><td>3월</td>
```

```
    <td>아기공룡둘리</td>
```

```
    <td>애니메이션</td></tr>
```

```
    <tr><td>4월</td>
```

```
    <td>코다</td>
```

```
    <td>남우주연상</td></tr>
```

```
</table>
```

```
<ul id="check">
```

```
  <li>점검사항
```



```

        <ol>
            <li>창문 닫기 </li>
            <li>냉난방 전원 끄기 </li>
        </ol>
    </li>
</ul>
</div>

```

```
<!-- [문제 3-4번] -->
```

```

<div class="footer">
    <em>춘천시 한림대학길1 정보과학대학</em>
</div>
</section>
</body>
</html>

```

▪ 3-1번 :

1. 메뉴는 제시된 결과처럼 화면 위쪽에 고정되어 있으며, 고정위치는 왼쪽 10px
2. 마우스 커서가 메뉴 영역위에 있으면 전체 메뉴가 표시 된다
3. 메뉴 항목에 마우스가 위치하면 텍스트에 밑줄을 표시한다.
4. 메뉴 항목의 글자색은 모두 다르게 한다.
5. 메뉴 항목의 너비는 100px, 안쪽 여백은 위/아래 2px, 왼쪽/오른쪽 10px
6. 플로팅 속성 사용 불가
7. 전체 margin은 auto로 한다.

▪ [문제 3-2번]

▪ [문제 3-3번]

1. 제시된 결과를 참고하여 출력과 관련된 속성만 작성
2. 표 너비와 높이는 300px, 90px
3. 점검사항의 경우 너비와 높이는 표와 동일하며 바깥쪽 여백 위/아래는 0px, 왼쪽/오른쪽 10px
4. 점검사항의 경우 안쪽여백은 위쪽만 10px, 배경색과 모서리 곡률은 임의로 정 할 것

▪ [문제 3-4번]

1. footer 요소는 부모를 기준으로 배치위치는 위쪽 10px, 왼쪽은 뷰포트를 기준으로 15
2. 수직 가운데 정렬이 되도록 한다
3. 배경색과 너비와 높이는 임의로 설정하고 안쪽 여백은 왼쪽만 50px

[소스]

```

<style>
    /* 메뉴 부분 */
    *{
        margin: auto;
    }

```

```

}

.menu{
    border-radius: 10px;
    background-color: aquamarine;
    left:10px;
    top:-10px;
    overflow: auto;
    position : fixed;
}

.menu:hover{
    top:10px;
}

.menu li{
    float: left;
    list-style-type: none;
    width: 100px;
    margin: 2px 10px;
}

.menu li:hover{
    text-decoration: underline;
}

.menu li:first-child{
    color: red;
}

.menu li:nth-child(2){
    color:blue;
}

.menu li:last-child{
    color:green;
}
</style>
<style>
/* 테이블과 표 부분 */
div:first-of-type{
    /* 색상 및 크기 적용 */
    width: 700px;
    height: 200px; /* 위치 설정 */
    position: absolute;
    left: 40%; top: 20%;
    margin-left: -250px;
}

#check{
    background-color: aquamarine;

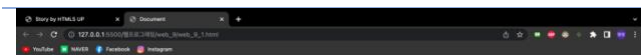
```

```

        border-radius: 10px;
        width: 300px;
        height: 90px;
        float:left;
    }
    table{
        width: 300px;
        height: 90px;
        border-collapse: collapse;
        float:right;
    }
    tr,td{
        border-bottom: 1px solid blue;
    }
</style>
<style>
    /* section 부분 */
    section{
        height: 100vh;
    }
    .footer{
        position: fixed;
        bottom:10px;
        left:15vw;
        width: 250px;
        height: 50px;
        background-color: brown;
        padding-left: 50px;
        line-height: 50px;
    }
    em{
        line-height: 50px;
    }
</style>

```

[실행 결과]



- 점검사항
- 1. 방문 달기
- 2. 냉난방 전원 끄기

월	제목	비고
3월	아기공룡돌리	애니메이션
4월	코다	남우주연상

춘천시 한림대학교1 정보과학대학



로그인 로그아웃 회원가입

- 점검사항
- 1. 방문 달기
- 2. 냉난방 전원 끄기

월	제목	비고
3월	아기공룡돌리	애니메이션
4월	코다	남우주연상

춘천시 한림대학교1 정보과학대학

(2) 플렉스 박스를 사용하여 다음과 같은 레이아웃을 구성하고 반응형 웹이 될 수 있도록 css 를 작성하시오

<body>

<nav class="menu_bar">

<div class="menu_left">

Hallym

</div>

<ul class="menu_item">

대학소개

전공

대학생활

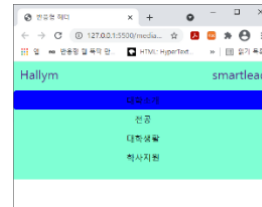


```

    <li> 학사지원 </li>
</ul>

<div class="menu_right">
    <a href="">smartlead</a>
</div>
</nav>
</body>

```



[소스]

```

<style>

    body{
        margin: 0;
    }

    .menu_bar{
        display: flex;
        background-color: aquamarine;
        justify-content: space-between;
        align-items: center;
        padding: 8px 12px;
    }

    .menu_item{
        display: flex;
        list-style: none;
        padding-left: 0;
    }

    a{
        text-decoration: none;
        color: rgb(3, 59, 59);
        font-weight: bold;
    }

    .menu_left,
    .menu_right{
        font-size: 24px;
    }

    .menu_item li{
        padding: 8px 12px;
    }

    .menu_item li:hover{
        background-color: blue;
        border-radius: 5px;
    }

```

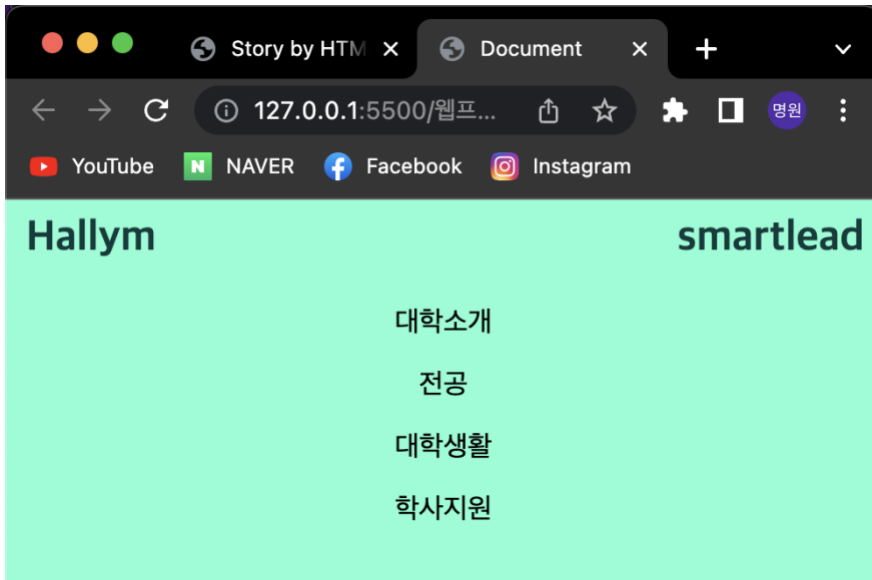
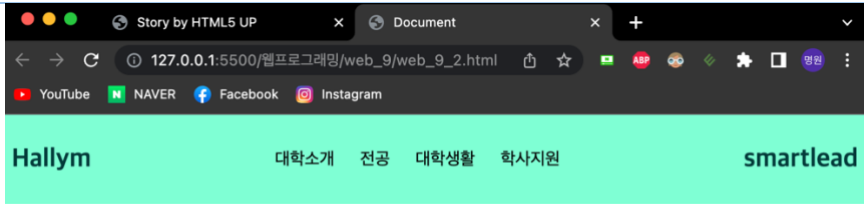
```
</style>
<style>
  @media screen and (min-width:500px) and (max-width:799px) {
    .menu_bar{
      flex-direction: column;
      align-items: flex-start;
    }

    .menu_item{
      flex-direction: column;
      align-items: center;
      width: 100%;
    }

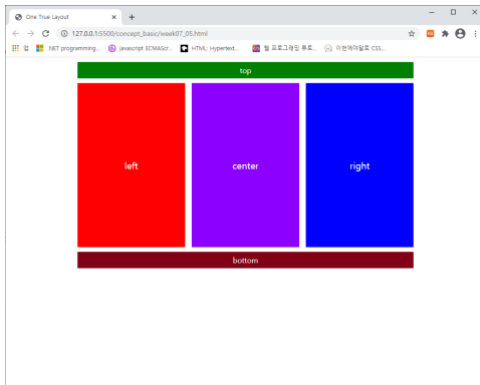
    .menu_item li{
      width: 100%;
      text-align: center;
    }

    .menu_right{
      position: absolute;
      right: 10px;
    }
  }
</style>
```

[실행 결과]



(3) 플렉스 박스를 사용하여 다음과 같은 One True 레이아웃을 작성하시오.



```
<script>
  body {
    width: 70vw;
    margin: 10px auto;
    color: white;
    text-align: center;
  }
</script>
<body>
  <div id="top">top</div>
  <div id="middle">
    <div class="item" id="left">left</div>
    <div class="item" id="center">center</div>
    <div class="item" id="right">right</div>
  </div>
  <div id="bottom">bottom</div>
</body>
```

[소스]

끝부분이 일치함 left right

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    body {
      width: 70vw;
      margin: 10px auto;
      color: white;
      text-align: center;
```



```
<div class="item" id="left">left</div>
<div class="item" id="center">center</div>
<div class="item" id="right">right</div>
</div>
<div id="bottom">bottom</div>
</body>
</html>
```

[실행 결과]

