



7

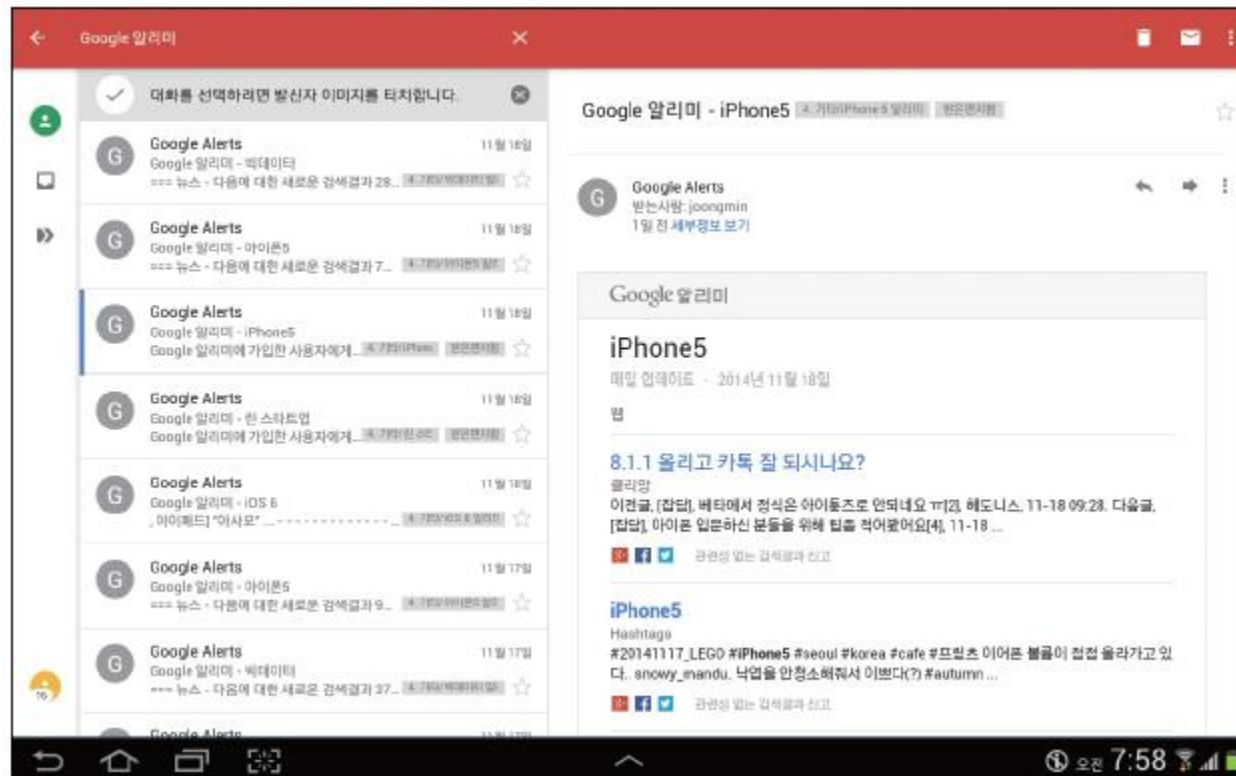
태블릿 PC 레이아웃

모던 웹 디자인을 위한 HTML5 + CSS3 입문(개정판)

❖ 동적 너비 레이아웃

- 동적 너비 레이아웃은 화면의 너비에 딱 맞게 구성되는 레이아웃이다.
- 일반적으로 태블릿 PC 애플리케이션은 동적 너비 레이아웃을 갖는다.

그림 7-1 안드로이드 Gmail 애플리케이션

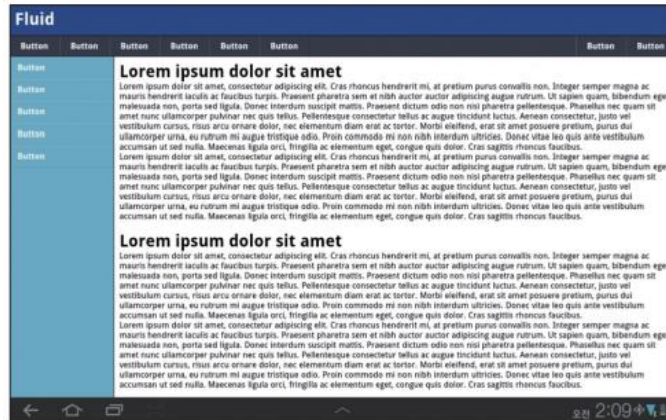


7.1 초기화

❖ 예제 결과

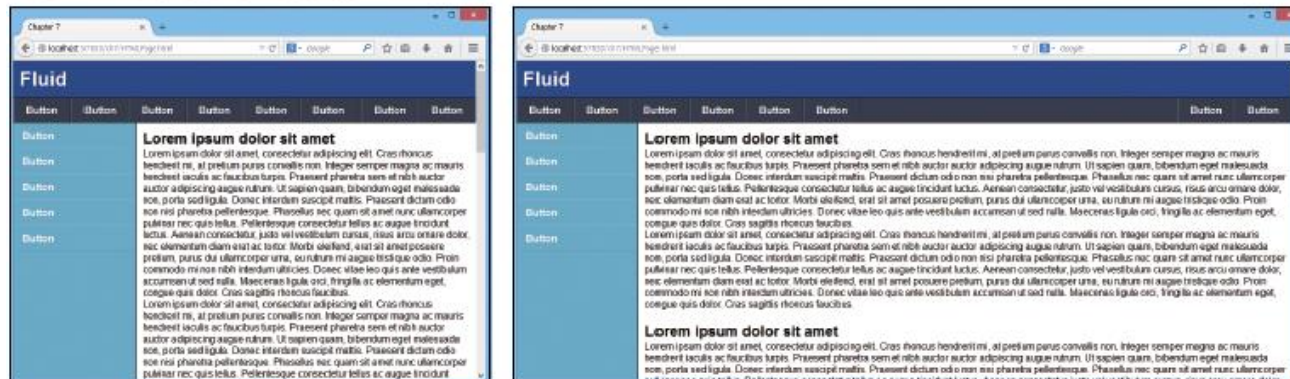
- 이번 장에서는 다음과 같은 웹 페이지를 만든다.

그림 7-2 7장 예제



- 너비가 변경되어도 내부의 구성물이 깨지지 않고 유지된다.

그림 7-3 너비가 동적인 레이아웃



7.1 초기화

❖ 다음과 같은 초기화 코드를 사용한다.

코드 7-2 초기화 코드

```
<!-- 초기화 -->
<style>
  * { margin: 0; padding: 0; }
  body { font-family: 'Helvetica', sans-serif; }
  li { list-style: none; }
  a { text-decoration: none; }
</style>
```



7.2 헤더 구성

❖ body 태그에 다음과 같은 header 태그를 생성한다.

코드 7-4 #main_header 태그의 내용 입력

```
<body>
  <header id="main_header">
    <h1>Fluid</h1>
  </header>
</body>
```

❖ #main_header 태그에는 다음 스타일을 적용한다.

코드 7-5 #main_header 태그의 스타일 사용

```
<!-- 헤더 -->
<style>
  #main_header {
    height: 60px;
    line-height: 60px;
    padding-left: 10px;
    border-bottom: 1px solid black;

    background: #1D4088;
    color: white;
  }
</style>
```



7.2 헤더 구성

❖ 코드를 실행하면 다음과 같이 출력한다.

그림 7-4 스타일을 적용한 #main_header 태그



7.3 내비게이션 구성

❖ body 태그에 다음과 같은 nav 태그를 생성한다.

코드 7-6 #main_gnb 태그 생성

```
<body>
  <header id="main_header"><!-- 생략 --></header>
  <nav id="main_gnb">

  </nav>
</body>
```

❖ 이어서 #main_gnb 태그는 다음과 같이 구성한다.

코드 7-7 #main_gnb 태그의 내용 입력

```
<nav id="main_gnb">
  <ul class="left">
    <li><a href="#">Button</a></li>
    <li><a href="#">Button</a></li>
    <li><a href="#">Button</a></li>
    <li><a href="#">Button</a></li>
    <li><a href="#">Button</a></li>
    <li><a href="#">Button</a></li>
  </ul>
  <ul class="right">
    <li><a href="#">Button</a></li>
    <li><a href="#">Button</a></li>
  </ul>
</nav>
```



7.3 내비게이션 구성

❖ #main_gnb 태그에는 다음 스타일을 적용한다.

코드 7-9 #main_gnb 태그의 모양 꾸미기

```
<!-- 내비게이션 -->
<style>
    #main_gnb {
        overflow: hidden;
        border-bottom: 1px solid black;
        background: #32394A;
    }

    #main_gnb > ul.left {
        overflow: hidden;
        float: left;
    }
    #main_gnb > ul.right {
        overflow: hidden;
        float: right;
    }

    #main_gnb > ul.left > li { float: left; }
    #main_gnb > ul.right > li { float: left; }
```



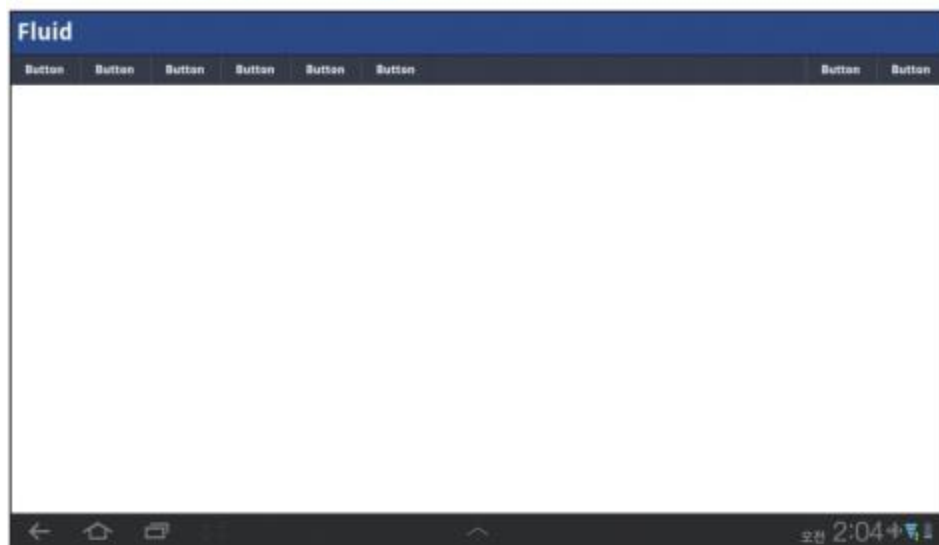
7.3 내비게이션 구성

```
/* a 태그 설정 */
#main_gnb a {
    /* 레이아웃 설정 */
    display: block;
    padding: 10px 20px;

    /* 색상 설정 */
    border-left: 1px solid #5F6673;
    border-right: 1px solid #242A37;
    color: white;
    font-weight: bold;
}
</style>
```

❖ 코드를 실행하면 다음과 같이 출력한다.

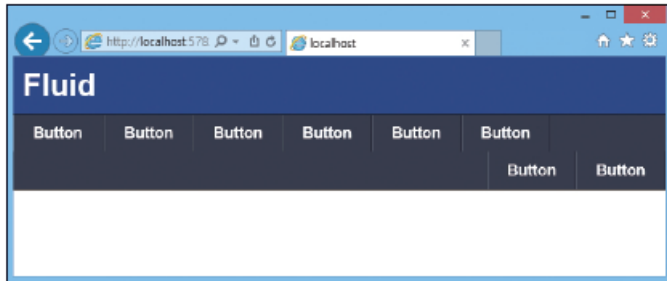
그림 7-5 스타일을 적용한 #main_gnb 태그



7.3 내비게이션 구성

- ❖ 현재 만들어진 내비게이션은 웹 브라우저의 너비가 작을 때에 다음과 같이 레이아웃이 깨진다.

그림 7-6 #main_gnb 태그의 문제점



- 일반적으로 “반응형 웹”을 사용해 이러한 문제를 해결한다. 이는 11장에서 다루는 내용이므로 우선 다음과 같이 최소 너비를 지정해서 해결한다.

코드 7-10 min-width 속성 지정

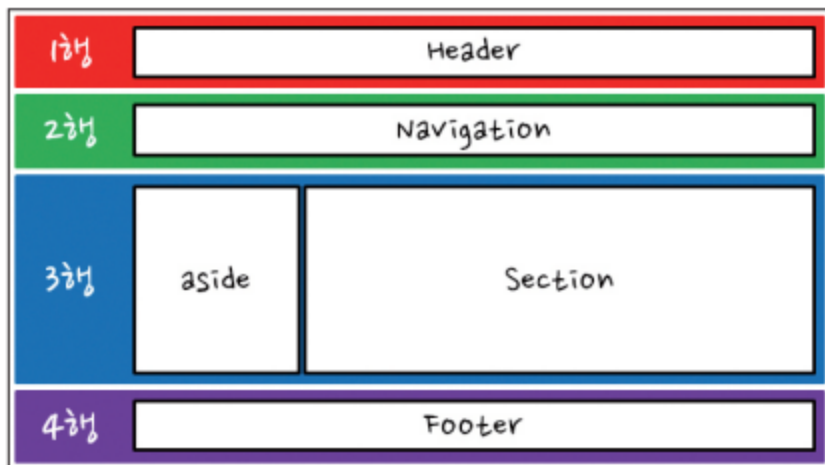
```
body {  
    min-width: 760px;  
}
```



7.4 콘텐츠 구성

- ❖ 콘텐츠는 다음 그림에서 3행을 나타낸다. 동적 너비 레이아웃에서 가장 중요한 부분이다.

그림 7-7 현재 만들고 있는 웹 페이지의 레이아웃 구성



- ❖ body 태그에 다음과 같은 div 태그를 생성한다.

코드 7-11 #wrap 태그 생성

```
<body>
  <header id="main_header"><!-- 생략 --></header>
  <nav id="main_gnb"><!-- 생략 --></nav>
  <div id="wrap">

    </div>
</body>
```



7.4 콘텐츠 구성

❖ #wrap 태그는 다음과 같이 구성한다.

코드 7-14 #wrap 태그의 내용 입력

```
<div id="wrap">
  <nav id="main_lnb">
    <ul>
      <li><a href="#">Button</a></li>
      <li><a href="#">Button</a></li>
      <li><a href="#">Button</a></li>
      <li><a href="#">Button</a></li>
      <li><a href="#">Button</a></li>
    </ul>
  </nav>
  <div id="content_wrap">
    <div id="content">
      <article>
        <h1>Lorem ipsum dolor sit amet</h1>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
      </article>
      <article>
        <h1>Lorem ipsum dolor sit amet</h1>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
      </article>
    </div>
  </div>
</div>
```



7.4 콘텐츠 구성

❖ #wrap 태그에는 다음과 같은 스타일을 적용한다.

코드 7-16 margin-left 속성과 padding-right 속성을 사용한 레이아웃 구성

```
<!-- 콘텐츠 -->
<style>
  #wrap { overflow: hidden; }
  #wrap > #main_lnb {
    float: left;
    width: 200px;
  }
  #wrap > #content_wrap {
    float: left;
    width: 100%;
    margin-right: -200px;
  }
  #wrap > #content_wrap > #content { padding-right: 200px; }
</style>
```

오른쪽으로 당기고

왼쪽으로 밀어줍니다.



7.4 콘텐츠 구성

❖ 코드를 실행하면 다음과 같이 출력한다.

그림 7-9 스타일을 적용한 #wrap 태그



7.5 수직 목록 구성

❖ #wrap 태그 내부의 #main_lnb 태그는 다음과 같은 스타일을 적용한다.

코드 7-18 #main_lnb 태그의 모양 꾸미기

```
<!-- 수직 목록 -->
<style>
  #wrap { background: #71B1D1; }
  #main_lnb > ul > li > a {
    display: block;
    height: 40px; line-height: 40px;
    padding-left: 15px;

    border-top: 1px solid #96D6F6;
    border-bottom: 1px solid #6298B2;
    color: white;
    font-weight: bold;
  }
</style>
```



7.6 본문 구성

❖ #wrap 태그 내부의 #content 태그는 다음과 같은 스타일을 적용한다.

코드 7-19 #content 태그의 스타일 사용

```
<!-- 본문 -->
<style>
  #content {
    background: white;
    border-left: 1px solid black;
  }
  article {
    padding: 10px;
  }
</style>
```



❖ body 태그에 다음과 같은 footer 태그를 생성한다.

코드 7-20 #main_footer 태그 생성

```
<body>
  <header id="main_header"><!-- 생략 --></header>
  <nav id="main_gnb"><!-- 생략 --></nav>
  <div id="wrap"><!-- 생략 --></div>
  <footer id="main_footer">
    <h3>HTML5 + CSS3 Basic</h3>
    <address>Website Layout Basic</address>
  </footer>
</body>
```

❖ #main_footer 태그는 다음과 같은 스타일을 적용한다.

코드 7-21 #main_footer 태그의 스타일 사용

```
<!-- 푸터 -->
<style>
  #main_footer {
    padding: 10px;
    border-top: 3px solid black;
    text-align: center;
  }
</style>
```



7.8 정리

❖ 최종적으로 다음과 같은 예제가 완성된다.

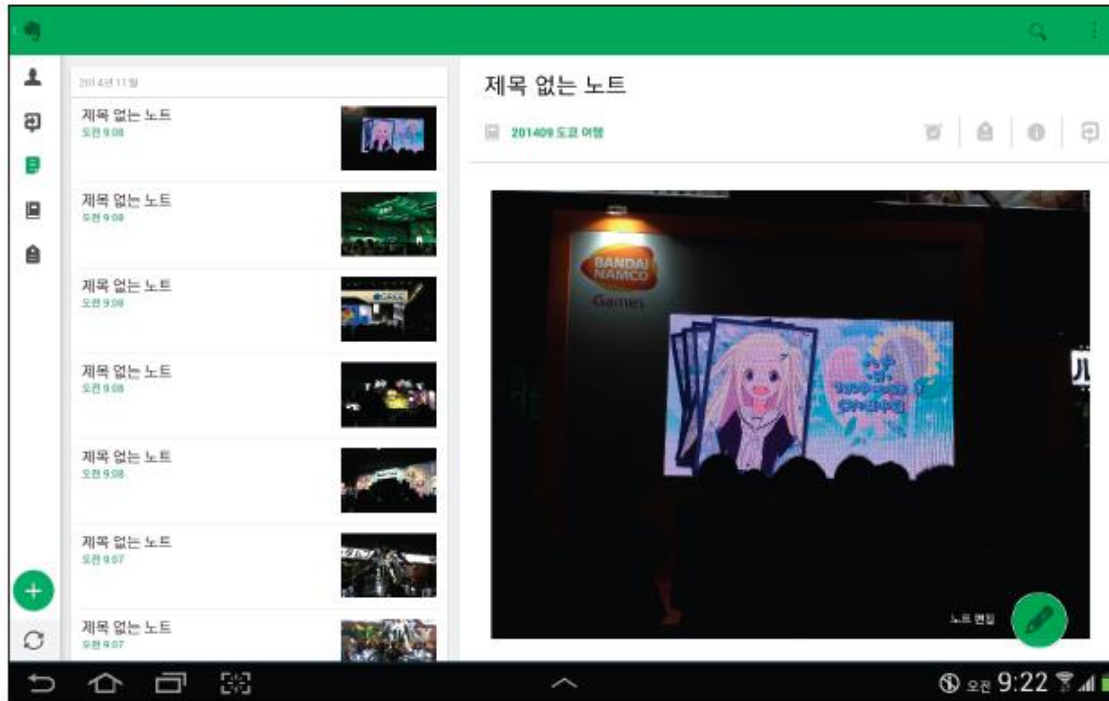
그림 7-15 완성된 7장 예제



7.9 고정 바

- ❖ 태블릿 PC의 애플리케이션을 보면 다음 그림처럼 왼쪽이나 오른쪽에 고정되어 있는 바를 볼 수 있다. 이를 고정 바라고 부른다.

그림 7-16 안드로이드 태블릿의 에버노트



❖ 간단하게 고정 바를 만드는 예제를 살펴보겠다. 다음과 같이 HTML 페이지를 구성한다.

코드 7-23 HTML 페이지 구성

```
<!DOCTYPE html>
<html>
<head>
  <title>Tablet Layout</title>
  <style>

  </style>
</head>
<body>
  <div id="top_bar">
    <h1>Top_Bar</h1>
  </div>
  <div id="bottom_bar">
    <h1>Bottom_Bar</h1>
  </div>
  <div id="left_bar">
    <h1>A</h1>
  </div>
  <div id="right_bar">
    <h1>B</h1>
  </div>
  <article>
    <h1>Lorem ipsum dolor sit amet</h1>
```



```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
<p>Nullam libero erat, ultrices non ultricies a.</p>
<p>Donec tortor nisi, ultricies a cursus ut.</p>
<p>Aliquam sapien massa, consequat in semper quis.</p>
</article>
</body>
</html>
```

- ❖ 고정 바를 생성하려면 웹 브라우저의 높이를 완전히 채워야 한다. 다음 코드를 사용해 전체 화면을 적용한다.

코드 7-24 전체화면 적용

```
<style>
  * { margin: 0; padding: 0; }
  html, body { height: 100%; }
</style>
```

- ❖ 이어서 다음과 같은 스타일을 적용한다.

코드 7-25 고정 바 생성

```
#top_bar {
  position: fixed;
  top: 0; left: 0; right: 0;
  height: 50px;

  background: red;
```



7.9 고정 바

```
}  
#bottom_bar {  
    position: fixed;  
    bottom: 0; left: 0; right: 0;  
    height: 50px;  
  
    background: red;  
}  
#left_bar {  
    position: fixed;  
    top: 50px; bottom: 50px;  
    left: 0; width: 50px;  
  
    background: blue;  
}  
#right_bar {  
    position: fixed;  
    top: 50px; bottom: 50px;  
    right: 0; width: 50px;  
  
    background: blue;  
}
```



7.9 고정 바

❖ 코드를 실행하면 다음과 같은 고정 바가 생성된다.

그림 7-17 생성된 고정 바



❖ 하지만 position 속성을 absolute 또는 fixed로 지정하면 다른 태그가 가려져서 제대로 출력되지 않는다.



❖ 따라서 본문에 다음과 같이 margin 속성을 부여한다.

코드 7-26 article 태그의 출력 영역 조정

```
article {  
    margin-top: 50px;  
    margin-left: 50px;  
    margin-right: 50px;  
    margin-bottom: 50px;  
}
```

padding 속성을 사용해도 됩니다.

❖ 최종적으로 다음과 같은 예제가 완성된다.

그림 7-18 완성된 고정 바





Thank You !

모던 웹 디자인을 위한 HTML5 + CSS3 입문(개정판)