

CHAPTER 13

프로젝트: 모바일 페이지 제작

01 미리 알아볼 내용

02 프로젝트 소개

03 레이아웃 구성

04 스타일시트 적용

학습목표

- 지금까지 배운 HTML과 CSS를 활용해서 웹 페이지를 만들 수 있습니다.
- CSS를 사용해 사용자와 상호 작용하는 방법을 배웁니다.

01

미리 알아볼 내용

보통 CSS와 자바스크립트를 구분할 때 'CSS는 디자인할 때 사용하고, 자바스크립트는 사용자와 상호 작용할 때 사용한다'고 말합니다. 그런데 사실 CSS는 사용자와 상호 작용할 때도 사용할 수 있습니다. 프로젝트를 본격적으로 만들기 전에 상호 작용할 때 사용할 수 있는 CSS부터 알아봅시다.

1 동위 선택자

5장에서 CSS 선택자를 배웠습니다. [표 5-1]에서 CSS 선택자를 설명할 때 다루지 않은 선택자가 하나 있는데, 바로 '동위 선택자'입니다. 동위 선택자는 동위 관계(같은 단계에 있는 것)에서 뒤에 위치한 태그를 선택할 때 사용합니다.

표 13-1 동위 선택자

선택자	설명
선택자A + 선택자B	선택자A 바로 뒤에 위치하는 선택자B를 선택
선택자A ~ 선택자B	선택자A 뒤에 위치하는 선택자B를 선택

조금 말이 어려우니 간단하게 예제를 살펴봅시다.

기본 예제 13-1

동위 선택자

1 첫 번째 형태: + h2 태그는 div.section-a 태그 내부에 있습니다. 이러한 단계에 있고 바로 뒤에 있는 태그를 선택하므로, h3 태그의 header 글자에만 색상이 적용됩니다.

코드 13-1 + 선택자(sibling_adjacent.html)

```
<!DOCTYPE html>
<html>
<head>
  <title>동위 선택자</title>
  <style>
    h2 + * {
      color: red;
    }
  </style>
</head>
<body>
  <div class="section-a">
    <h1>header</h1>
    <h2>header</h2>
    <h3>header</h3>
    <h4>header</h4>
  </div>
  <h3>content title</h3>
</body>
</html>
```

header

header

header

header

content title

2 두 번째 형태: ~ h2 태그 뒤에서 이러한 단계에 있고 뒤에 있는 모든 태그를 선택하므로, h3 태그와 h4 태그에 입력한 header 글자에만 모두 색상이 적용됩니다.

코드 13-2 ~ 선택자(sibling_general.html)

```
<!DOCTYPE html>
<html>
<head>
  <title>동위 선택자</title>
  <style>
    h2 ~ * {
      color: red;
    }
  </style>
</head>
<body>
  <div class="section-a">
    <h1>header</h1>
    <h2>header</h2>
    <h3>header</h3>
    <h4>header</h4>
  </div>
  <h3>content title</h3>
</body>
</html>
```

header
header
header
header
content title

2 동위 선택자를 활용한 풀다운 메뉴

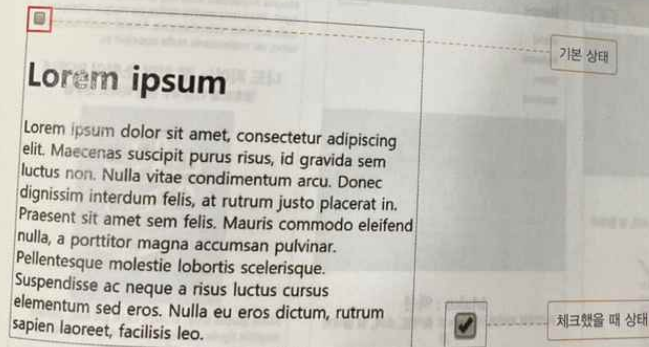
그럼 동위 선택자를 활용해 풀다운 메뉴를 만들어 보겠습니다. 일일이 설명하는 것보다 실제 코드를 살펴보는 것이 훨씬 이해하기 쉬운 것입니다.

기본 예제 13-2 input 태그를 활용한 풀다운 메뉴

[코드 13-3]을 살펴봅시다. input 태그 뒤에 있는 div 태그를 선택하는 선택자를 사용하고, display 속성을 none으로 변경하는 코드입니다. 이 코드를 실행한 후 input 태그에 체크하면 뒤에 있는 div 태그가 사라집니다. 자바스크립트를 사용하지 않고도 사용자와 상호 작용을 할 것입니다.

코드 13-3 label 태그와 input 태그를 활용한 폴다운 메뉴(sibling_pulldown.html)

```
<!DOCTYPE html>
<html>
<head>
  <title>Checkbox</title>
  <style>
    input: checked + div {
      display: none;
    }
  </style>
</head>
<body>
  <input type="checkbox">
  <div>
    <h1>Lorem ipsum</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
  </div>
</body>
</html>
```



4장 99쪽의 NOTE에서 설명한 것처럼 label 태그를 입력하면 for 속성에 적용된 input 태그가 활성화됩니다. 따라서 직접 체크하지 않고 label 태그를 선택해서 체크할 수도 있습니다. 그럼 이러한 내용을 활용해서 간단한 모바일 웹 페이지를 작성해 봅시다.

02

프로젝트 소개

일반적으로 데스크톱은 가로로 넓은 화면을 사용하지만, 스마트폰 등 모바일 장치는 세로로 긴 화면을 사용합니다. 따라서 종합 예제에서 만들었던 웹 페이지처럼 본문과 메뉴를 왼쪽과 오른쪽에 배치하는 형태로 레이아웃을 잡으면 보기가 힘듭니다.

기본적으로 모바일 장치를 대상으로 하는 웹 페이지는 레이아웃을 위쪽에서 아래쪽으로 직선적으로 배치하고, 메뉴 등은 사용자가 버튼을 눌렀을 때만 보여 주는 것이 좋습니다. 이 장에서는 [그림 13-1]과 같이 간단한 모바일 페이지를 만듭니다. 버튼을 눌러 메뉴를 보여 주고 감추는 기능을 갖고 있는데요. label 태그와 input 태그를 조합해서 이를 구현해 보겠습니다.

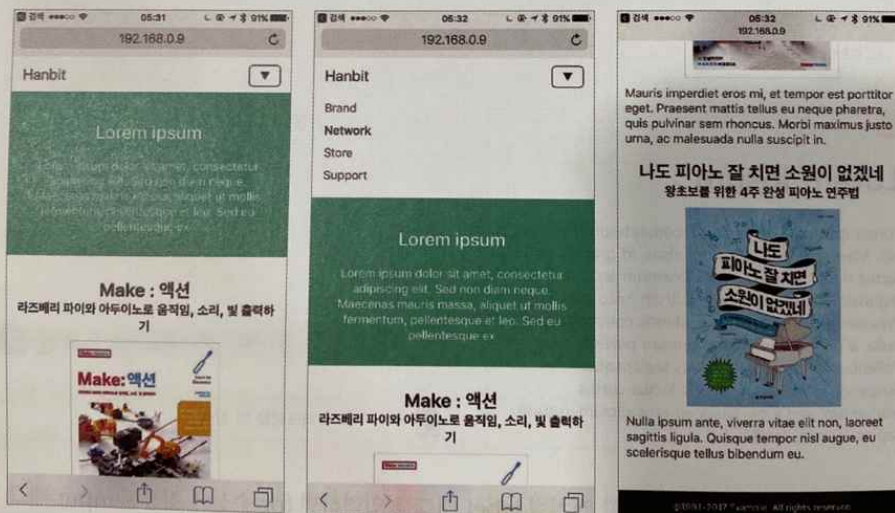


그림 13-1 이 장에서 만들 모바일 페이지

이 프로젝트는 chapter_13-시작 폴더에 있는 파일을 사용해 시작합니다.

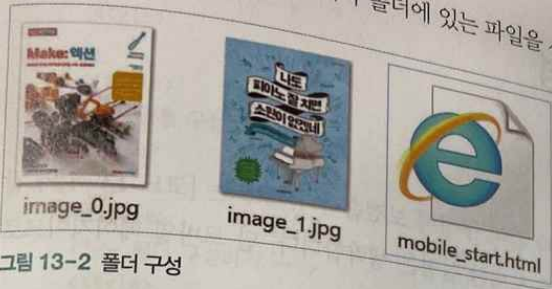


그림 13-2 폴더 구성

03

레이아웃 구성

시작 페이지 레이아웃을 [그림 13-3]과 같이 구성해 보겠습니다. 코드는 [코드 13-4]와 같습니다. 그다지 어려운 구성은 아니므로 자세한 설명은 생략합니다. 단 모바일 페이지이므로 viewport meta 태그를 사용했습니다.

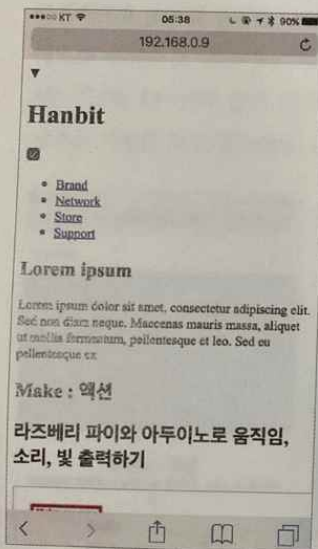


그림 13-3 시작 페이지 레이아웃

코드 13-4 시작 페이지(mobile_start.html)

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML5 with Hanbit</title>
  <meta name="viewport" content="width=device-width,initial-scale=1.0,
    minimum-scale=1.0,maximum-scale=1.0,user-scalable=no">
```



```
<!-- 초기화하기 -->
<style>

</style>
<!-- 공통 구성하기 -->
<style>

</style>
<!-- 헤더 구성하기 -->
<style>

</style>
<!-- 본문과 푸터 구성하기 -->
<style>

</style>
</head>
<body>
  <div id="main-header">
    <header class="container">
      <div>
        <label for="checkbox">▼</label>
        <h1>Hanbit</h1>
      </div>
      <input id="checkbox" checked="checked" type="checkbox">
      <nav>
        <ul>
          <li><a href="#">Brand</a></li>
          <li class="active"><a href="#">Network</a></li>
          <li><a href="#">Store</a></li>
          <li><a href="#">Support</a></li>
        </ul>
      </nav>
    </header>
  </div>
```

```
<div id="main-content">
  <article class="container">
    <h1>Lorem ipsum</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
    Sed non diam neque. Maecenas mauris massa, aliquet ut mollis
    fermentum, pellentesque et leo. Sed eu pellentesque ex</p>
  </article>
</div>
<div id="sub-content">
  <article class="container">
    <h1>Make : 액션</h1>
    <h2>라즈베리 파이와 아두이노로 움직임, 소리, 빛 출력하기</h2>
    
    <p>Mauris imperdiet eros mi, et tempor est porttitor eget.
    Praesent mattis tellus eu neque pharetra, quis pulvinar sem rhoncus.
    Morbi maximus justo urna, ac malesuada nulla suscipit in.</p>
    <br>
    <h1>나도 피아노 잘 치면 소원이 없겠네</h1>
    <h2>왕초보를 위한 4주 완성 피아노 연주법</h2>
    
    <p>Nulla ipsum ante, viverra vitae elit non, laoreet sagittis ligula.
    Quisque tempor nisl augue, eu scelerisque tellus bibendum eu.</p>
  </article>
</div>
<div id="main-footer">
  <footer class="container">
    <div>©1991-2017 Example. All rights reserved.</div>
  </footer>
</div>
</body>
</html>
```

04

스타일시트 적용

기본 레이아웃이 구성된 웹 페이지에 스타일시트를 하나씩 적용해 보겠습니다. 완성된 코드는 완료 폴더의 mobile_end.html에 있습니다.

1 초기화

초기화 부분에서는 [코드 13-5]와 같은 요소를 초기화합니다. 종합 예제에서도 언급했듯이, 태그에 적용되는 기본 스타일이 조금씩 다른 문제를 해결하려면 스타일을 강제로 지정해야 합니다.

코드 13-5 초기화

```
<!-- 초기화 -->
<style>
  * {
    margin: 0;
    padding: 0;
    font-family: "Segoe UI", Roboto, "Helvetica Neue", Arial, sans-serif;
  }

  a { text-decoration: none; }
  img { border: 0; }
  li { list-style: none; }
</style>
```

2 공통 구성

공통 구성은 [코드 13-6]과 같이 작성합니다. 이렇게 하면 클래스에 container라고 입력한 태그가 모두 중앙 정렬되며, 위쪽과 아래쪽에 여백이 생깁니다.

코드 13-6 공통 구성

```

<!-- 공통 -->
<style>
    .container {
        margin-left: auto;
        margin-right: auto;
        padding-left: 10px;
        padding-right: 10px;
    }
</style>

```

3 헤더 구성

헤더는 [코드 13-7]과 같이 작성합니다. 이 장의 앞부분에서 살펴본 내용을 바탕으로 간단하게 input 태그로 폴다운 메뉴를 만듭니다. 그리고 label 태그로 input 태그의 체크 상태를 변경할 수 있게 for 속성을 지정한 후 버튼처럼 구성합니다.

코드 13-7 헤더 스타일

```

<!-- 헤더 -->
<style>
    #main-header {
        padding: 10px 5px;
    }

    #main-header h1 {
        font-size: 20px;
        font-weight: normal;
        line-height: 27px;
        color: #222;
    }

    #main-header ul {
        margin-top: 20px;
        margin-bottom: 10px;
    }

```



```
#main-header li {  
    margin-top: 10px;  
    margin-bottom: 10px;  
}  
  
#main-header li a {  
    color: #555;  
}  
  
#main-header li.active a {  
    color: black;  
    font-weight: 500;  
}  
  
#main-header label {  
    text-align: center;  
    float: right;  
    width: 40px;  
    height: 27px;  
    line-height: 27px;  
    color: #222;  
    border: 1px solid #222;  
    border-radius: 5px  
}  
  
#checkbox {  
    display: none;  
}  
  
#checkbox + nav {  
    display: block;  
}  
  
#checkbox:checked + nav {  
    display: none;  
}  
</style>
```

label 태그로 체크 상태를 변경할 수 있으므로 input 태그는 보이지 않게 만들어 화면을 구성합니다. 이렇게 label 태그와 input 태그를 사용하면 [그림 13-4]와 같습니다.

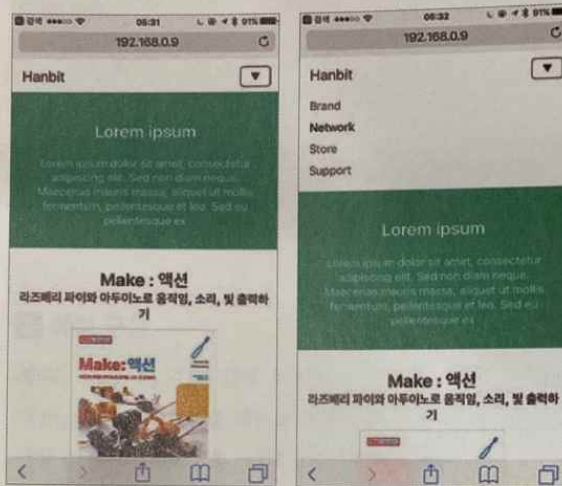


그림 13-4 label 태그와 input 태그를 사용한 풀다운 메뉴

4 본문과 푸터 구성

본문과 푸터는 [코드 13-8]과 같이 작성합니다. 글꼴, 글자색, 정렬 정도만 지정합니다.

코드 13-8 본문과 푸터 스타일

```
<!-- 본문과 푸터 -->
<style>
  #main-content {
    background: #00afa0;
    padding: 40px 15px 30px;
  }

  #main-content h1 {
    text-align: center;
    margin-bottom: 20px;
    color: white;
    font-weight: 500;
  }
</style>
```

```
#main-content p {  
    color: #222;  
    text-align: center;  
    color: white;  
}  
  
#sub-content {  
    padding-top: 30px;  
}  
  
#sub-content h1 {  
    text-align: center;  
    color: #333;  
}  
  
#sub-content h2 {  
    text-align: center;  
    color: #333;  
    font-size: 17px;  
}  
  
#sub-content p {  
    color: #555;  
}  
  
#sub-content img {  
    width: 200px;  
    display: block;  
    margin: 10px auto;  
}  
  
#main-footer {  
    padding-top: 40px;  
}
```

```

    footer {
      background: #323232;
      color: white;
      font-size: 12px;
      text-align: center;
      line-height: 40px;
    }
  </style>

```

최종 완성한 페이지는 [그림 13-5]와 같습니다.

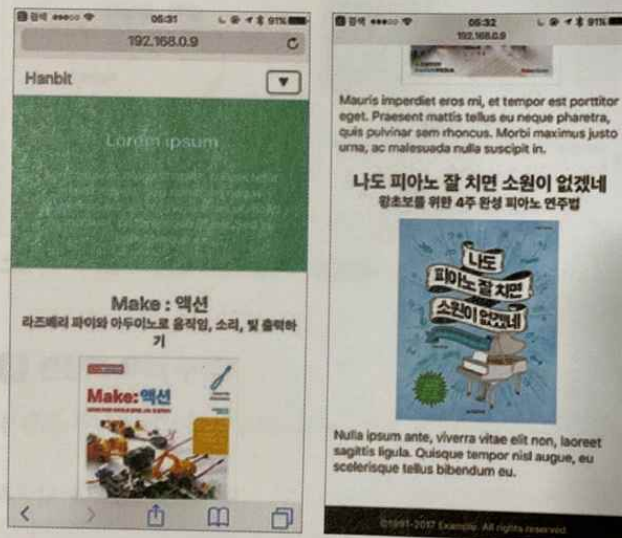


그림 13-5 완성 페이지

간단하지만 약간 특별한 요소가 조합되어 있는 모바일 페이지를 만들어 보았습니다. 자바스크립트도 배웠으므로 이 장에서 만든 풀다운 메뉴를 자바스크립트로 어떻게 구현할 수 있는지 생각해 보세요. 또 단순히 배치된 본문에 이미지 슬라이더를 적용해서 이미지도 움직여 봅니다.