

chapter

14

사이트 제작 전 준비 작업 및 프론트 페이지 제작

14장에서는 사이트 제작에 들어가기에 앞서, 필요한 사항을 먼저 점검하도록 하겠습니다.

사이트를 제작하기 위해선 먼저 준비해야 하는 사항이 몇 가지 있는데, 실제 서버와 같이 동작하게 해 주는 개인용 웹 서버가 준비되어 있어야 하고, 사이트의 디렉토리 또한 구성되어야 합니다. 또한 사이트를 제작하려면, 포토샵과 같은 그래픽 저작 도구를 이용해서 레이아웃을 구성해야 함은 물론이고, 그래픽 요소가 필요한 부분을 따로 저장해야 하는 방법도 알아야 합니다.

이번 장에서는 본격적으로 사이트를 제작하도록 하겠습니다. 우선 사이트 제작에 앞서 몇 가지 준비사항 및 주의점이 있습니다.

1. 예제를 실행하기 위해선 WAMP 또는 MAMP 서버가 필요합니다.
WAMP 서버 설치 방법은 부록을 참고하기 바랍니다.
2. 포토샵을 이용해서 만들어진 사이트 디자인 레이아웃이 있어야 합니다.
포토샵 사용법에 대해 자세히 학습하지 않습니다만, 웹에서 사용하는 레이아웃은 어떻게 잘라내고 어떤 파일을 사용하는가에 대한 설명은 간단하게 진행하겠습니다. 포토샵으로 만들어진 샘플 파일은 원본 PSD 파일로 제공됩니다.
3. 필요한 폴더들은 미리 만들어 놓습니다.
사이트 제작 과정에서 필요한 폴더 명을 알려드립니다.
4. 여기 사용된 예제들에는 웹 프로그래밍(PHP, ASP, .Net 등)이 적용되지 않은 상태입니다. 웹 프로그래밍에 대해서도 전혀 언급되지 않습니다.
5. 제작되는 사이트는 HTML5, CSS3의 거의 모든 기능이 지원되는 브라우저를 사용해야 합니다. IE10 이상, 구글 크롬 26 이상, 파이어폭스 20 이상, 사파리의 최신 버전은 맥에서만 지원되기 때문에 여기서 언급을 하진 않겠습니다. 하위 브라우저에 대한 호환성도 다루긴 합니다만, 아무래도 효과나 기타 여러 기능들이 완벽하게 동작하지 않을 수 있습니다.

위와 같은 준비사항을 명심하고, 이제부터 사이트 제작에 들어가 보도록 하겠습니다.

책에서 나오는 예제는 임베디드 방식으로 CSS를 적용합니다. 실제 사이트 작업 시에는 CSS 파일을 따로 만들어서 작업을 하는데, 책에서는 예제 파일을 하나씩 실행해야 하는 문제가 있어, 작업이 끝나기 전에는 임베딩 방식으로 CSS를 적용하고, 최종 웹사이트에서는 CSS를 별도로 분리하도록 하겠습니다.

주의) 예제로 제공된 압축 파일을 풀면, yngsite라는 폴더가 생성되고, 그 폴더 안에 파일들이 있습니다. 학습이 제대로 이루어지려면, 먼저 부록에 나와 있는 WAMP 또는 MAMP를 설치하여야 합니다.

WAMP 또는 MAMP가 설치된 상태에서 WAMP 서버 내에 있는 www 폴더, MAMP 서버의 httdocs 폴더에 yngsite라는 폴더를 이동해 주면 됩니다.

모든 예제들은 브라우저에서 http://localhost/yngsite라는 곳에서 실행되게 되며, 예제 소스 경로 또한 localhost로 표기됩니다. 예를 들어, http://localhost/yngsite/index04.php 이렇게 표시됩니다.



1

사이트 준비 작업

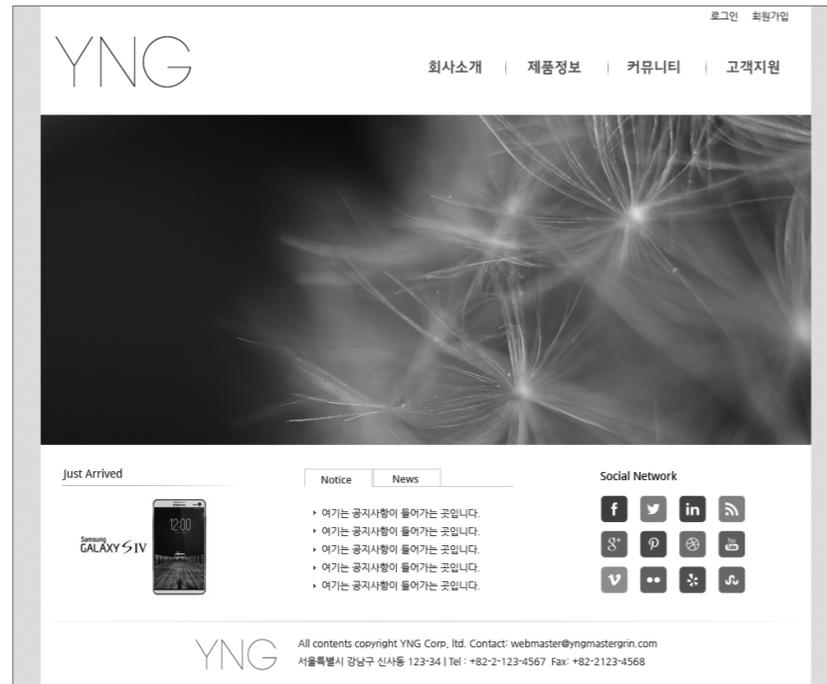
우선 WAMP 서버가 PC에 설치된 상태이면, WAMP 서버가 설치된 폴더에 있는 서버 폴더 www 내부에 사이트용 폴더를 하나 만듭니다. 필자는 yngsite라는 폴더를 하나 만들겠습니다.

여기에 다음과 같이 일단 3개의 폴더를 만들어 놓습니다.

imgs/ 이미지 파일들이 저장되는 곳입니다.
css/ CSS 파일은 여기에 넣습니다.
scripts/ 자바스크립트 파일들은 여기에 넣습니다.

이 외에 사이트가 아주 큰 경우에는 사이트의 메인 메뉴에 따른 폴더들을 하나씩 만들어 놓습니다. 하지만 전체 페이지가 20페이지 이내일 경우에는 폴더에 넣지 않고 루트 폴더에 작업하는 것이 편리합니다. 현재 만들 사이트도 규모가 크지 않기 때문에 루트에 파일을 넣고 작업하도록 하겠습니다.

최종적으로 만들려는 사이트의 프론트 페이지는 [그림 14-1]과 같습니다.



[그림 14-1] 제작할 사이트의 프론트 페이지

이런 형태의 웹사이트 레이아웃은 단순하지만, 일반적으로 많이 사용되고, 오랜 기간 사용할 수 있다는 장점이 있습니다. 만들려는 사이트는 기본적으로 HTML5와 CSS3를 기본으로 제작할 예정이며, CSS3를 이용할 수 없는 특수 효과는 jQuery를 이용해서 만들겠습니다.

가장 우선적으로 해야 할 작업은 프론트 페이지에서 이미지로 처리할 부분과 일반 코드로 처리할 것을 구분하는 것입니다. 또한 만들어진 디자인 레이아웃을 기반으로 HTML 코드를 작성해 두면 편리합니다.

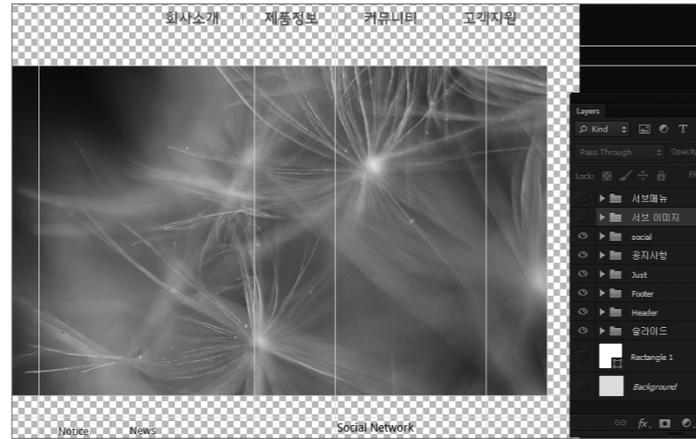
[그림 14-2]와 같이 미리 각 부분별로 태그 선택자와 아이디 선택자를 정리합니다.



[그림 14-2] 선택자를 각 부분별로 정의

여기서 아주 잠깐 포토샵을 이용해서 이미지로 처리할 부분에 대해서 잘라내고 저장하는 방법에 대해서 설명해 드리겠습니다. 지금 필자가 사용하는 포토샵은 CS6 버전입니다만, 어떤 포토샵 버전을 사용하더라도 동일한 작업을 할 수 있습니다.

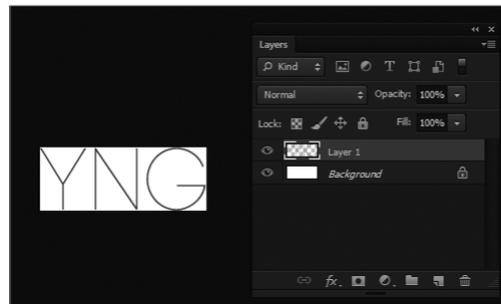
[그림 14-3]을 보면 포토샵의 레이어 팔레트에서 배경으로 사용된 레이어의 눈 아이콘을 비활성화하면 배경이 사라져 투명해지는 것을 알 수 있습니다. 이때 이미지로 저장할 부분을 선택툴로 선택한 후 CTRL + SHIFT + C 키를 누릅니다. CTRL + SHIFT + C는 copy merged 기능으로 어떤 레이어 상태에서 작업하더라도 해당 선택 부분을 복사하는 기능이 있습니다. 그 후 CTRL + N 키를 누르면 새로운 작업 윈도우가 나오며, 여기에 CTRL + V 키를 눌러 붙여넣기를 합니다(그림 14-5 참조).



[그림 14-3] 포토샵에서 제작된 파일의 배경 레이어 감추기

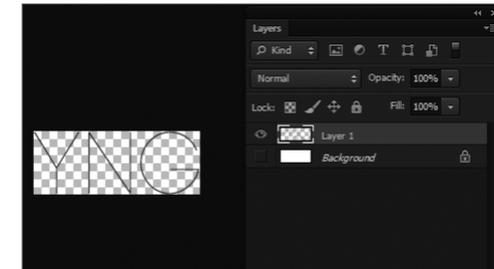


[그림 14-4] 선택 툴을 이용해서 해당 이미지 부분 선택



[그림 14-5] 선택된 이미지의 붙여넣기 후 상태

[그림 14-5]를 보면 이미지가 선택된 부분보다 작게 이미지 크기에 딱 맞는 크기로 붙여 넣어진 것을 확인할 수 있습니다. 여기서 다시 Background 레이어에 있는 눈 아이콘을 클릭해서 배경을 없애 줍니다(그림 14-6 참조).



[그림 14-6] 배경이 제거된 이미지

이렇게 배경이 제거된 이미지는 3가지 파일 포맷으로 저장할 수 있는데, 그림과 같이 배경이 사용되지 않는 경우 gif 파일 또는 png 파일로 저장하면 됩니다. <여기서 잠깐>에서도 언급되지만 로고와 같이 복잡한 곡선이 들어가 있는 경우에는 png 파일로 저장하는 것이 좋습니다.

| 여기서 잠깐 |

배경이 투명한 파일은 GIF와 PNG라는 두 가지 타입으로 저장할 수 있습니다.

GIF 파일은

인터넷 초창기부터 사용하던 파일 포맷이며, 역사가 가장 오래된 파일 타입이라고 할 수 있습니다. GIF 파일의 장점은 파일 사이즈가 작으며, Animated GIF 즉 파일 자체에서 애니메이션 파일로 만들 수 있습니다. 또한 배경을 투명하게 만들 수 있습니다. 하지만 색상 표현이 256 컬러로 제한되고, 투명한 배경을 지정할 경우 복잡한 곡선은 거칠게 표현됩니다. 직선으로만 되어 있으면 GIF 파일을 추천합니다. 따라서 웹에서 사용하는 아이콘이나, 색이 복잡하지 않은 단순한 이미지는 GIF 파일이 좋습니다.

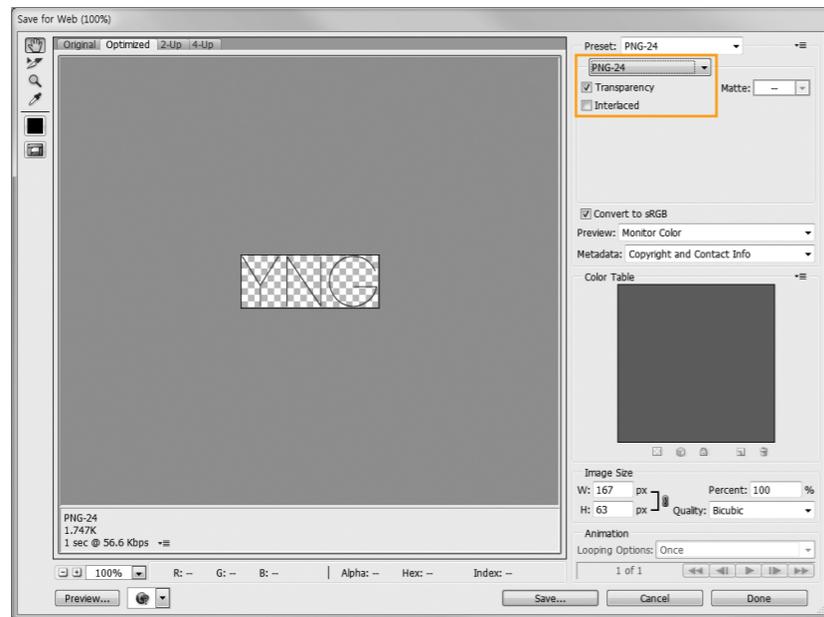
PNG 파일은

인터넷 전용으로 나온 파일로 Portable Network Graphic이란 의미입니다. 가장 최근에 나온 그래픽 전용 파일로, GIF 파일과 달리 투명한 배경을 완벽하게 투명하게 만들 수 있는 장점이 있습니다. PNG 파일은 png8, png16, png24, png32와 같이 8비트 16비트 24비트 32비트 파일 포맷이 존재합니다. png8 포맷은 투명 배경 효과가 gif보다 약간 뛰어나거나 별 차이가 없습니다. 색상 또한 256 컬러만 지원합니다. png24 이상은 투명 효과가 완벽하며, 많은 수의 색상을 구현할 수 있습니다만, 파일 크기가 제일 크다는 단점이 있습니다.

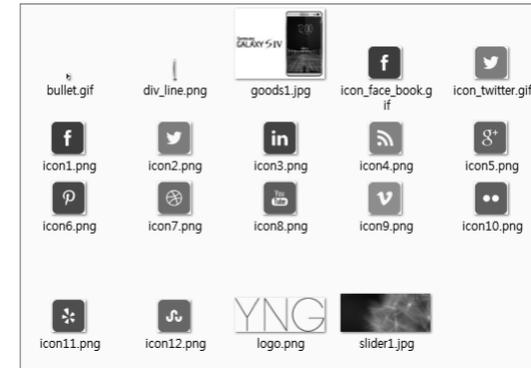
따라서 인터넷에서 이미지 파일을 사용할 때, 간단한 아이콘은 gif 파일로, 복잡한 곡선이 들어가지만 투명한 배경이 필요하다면 png 파일을, 사진 파일은 jpg 파일로 사용하는 것이 좋습니다.

포토샵에서 단축키 ALT + SHIFT + CTRL + S 즉 Save for web(웹으로 저장)을 누릅니다. 그러면 [그림 14-7]과 같은 화면이 나오며, 그림에서와 같이 파일 타입을 선택한 후 저장을 하면 됩니다. 여기서는 PNG-24로 선택하고 Transparency에 체크를 해주면 배경이 투명한 png 파일로 저장을 할 수 있습니다. 이 파일에 이름을 지정하고 wamp 서버 www 폴더 내에 있는 yngsite의 imgs 폴더에 저장합니다. 이런 방식으로 이미지로 처리할 부분은 전부 저장합니다.

슬라이더 이미지는 압축률 60% 정도로 설정한 jpg 형식으로 나머지 파일 또한 적당한 파일 포맷으로 저장합니다. 필자가 미리 작업해 둔 폴더(그림 14-8 참고)를 보면 각 파일들이 어떤 포맷으로 저장되어 있는지 알 수 있습니다. 참고하기 바랍니다.



[그림 14-7] save for web 화면



[그림 14-8] imgs 폴더에 저장된 파일 형식

마지막으로 준비할 사항은 CSS Reset이라 불리는 CSS 초기화 파일을 설정하는 것입니다. CSS 초기화는 각 브라우저마다 HTML 태그들 간의 마진과 패딩 및 기타 속성 값이 조금씩 차이가 나기 때문에 CSS를 이용해서 값을 초기화하는 것을 말합니다.

가장 간단한 방법은 CSS 속성에 다음과 같이 적용하는 것입니다.

```
* {margin:0; padding:0}
```

전체 선택자를 이용해서 모든 태그 선택자의 마진과 패딩을 0으로 만드는 것입니다.

가장 간단하고, 쉽습니다. 하지만 이렇게 전체 선택자를 이용해서 모든 선택자의 마진과 패딩을 0으로 만든다고 문제가 해결되지 않는 경우도 있습니다. 이럴 경우 조금은 정교하게 CSS를 이용하여 선택자의 값을 설정해줘야 합니다. 이럴 때 사용하는 것이 외부 파일로 CSS reset 파일로 만들어 @import 또는 link 명령을 이용해 만들려는 사이트에 포함하면 됩니다. 이 CSS reset 중 가장 유명한 것이 Eric Meyer의 CSS Reset입니다. 그 다음 유명한 것이 야후에서 만든 YUI Reset이 있습니다. 그리고 HTML5가 나오면서 html5doctor라는 사이트에서 만든 HTML5 reset도 있습니다. 필자가 추천하는 Reset 파일은 구글 코드에 있는 Eric Meyer's CSS Reset과 hmlt5doctor.com's HTML5 reset을 결합한 reset5라는 CSS reset을 추천합니다. 해당 파일은 <https://code.google.com/p/reset5/>에서 구할 수 있습니다.

이제 프론트 페이지를 시작으로 사이트를 제작해 보겠습니다.



프론트 페이지 제작

이제 본격적으로 프론트 페이지를 제작합니다. 3부 1장에서 HTML5 템플릿 파일 “html5css3template.html”을 만들었습니다. 이제부터 설명하는 예제들은 “html5css3template.html”을 기초로 작업을 하겠습니다.

먼저 에디터에서 html5css3template.html 파일을 열어 “다른 이름으로 저장(save as)”을 선택하여, index.php 파일이라고 저장합니다.

제일 먼저 #wrap이라는 아이디 선택자를 이용해서 문서 전체를 감쌉니다. wrap이라는 아이디 선택자는 문서 전체를 감싸고 문서를 중앙에 배치하는 역할을 합니다.

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>YNG에 오신 여러분을 환영합니다.</title>
  <style></style>
</head>
<body>
  <div id="wrap">

  </div>
</body>
</html>
```

7 14-1] #wrap 아이디 선택자를 이용하여 전체 문서를 감싸 줌 _ /yngsite/index01.php

이제 header 부분을 작업하겠습니다. 이전 [그림 14-2]에서 따로 헤더 부분만 분리해 보면 [그림 14-9]과 같습니다.



[그림 14-9] 헤더 부분에 있는 선택자들

[그림 14-9]에서 설정한 선택자를 기반으로 HTML 코드를 구성해 보겠습니다.

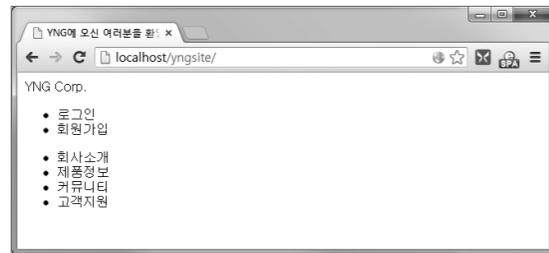
```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>YNG에 오신 여러분을 환영합니다.</title>
  <style></style>
</head>
<body>
  <div id="wrap">
    <header>
      <div id="logo">YNG Corp.</div>
      <div id="hlink">
        <ul>
          <li>로그인</li>
          <li>회원가입</li>
        </ul>
      </div>
      <nav>
        <ul>
          <li>회사소개</li>
          <li>제품정보</li>
          <li>커뮤니티</li>
          <li>고객지원</li>
        </ul>
      </nav>
    </header>
  </div>
</body>
</html>
```

• [코드 14-2] header 부분에 대한 HTML 코드 정리 _ /yngsite/index02.php

코드를 보면 알겠지만, 현재 모든 부분이 텍스트로 구성되어 있습니다. 이미지는 전혀 들어가지 않은 상태입니다. 결과는 [그림 14-10]에서 확인할 수 있습니다.

* 최종 파일은 index.php 파일이 되지만, 단계별 학습을 위해 예제 파일에 숫자를 붙여서 표기하도록 하겠습니다.

이제 CSS 속성을 이용해서 header 부분에 대한 디자인 작업을 시작해 보겠습니다.



[그림 14-10] header 부분의 HTML 코딩 후 결과 화면

이제 여기에 reset5.css 파일을 링크로 연결해 주고 기본적인 body와 #wrap 선택자 그리고 기본적인 링크 효과에 대한 속성을 정의합니다.

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>YNG에 오신 여러분을 환영합니다.</title>
  <link href="css/reset5.css" rel="stylesheet" type="text/css">
  <!-- reset5.css 파일을 링크 형식으로 적용 -->
  <style>
  body{
    font-size: 0.75em; /* 사이트 전체 기본 폰트 사이즈 설정 */
    font-family: "맑은 고딕", "Malgun Gothic", "돋움", Dotum, AppleGothic, sans-serif;
    /*사이트 전체 기본 폰트 설정 */
    line-height:1.2em; /* 글 단락 사이 기본 간격 설정 */
    color:#333; /* 기본 폰트 색상 지정 */
    background-color: #e8e8e8; /* 기본 바탕 배경색 지정 */
  }
  #wrap{
    margin: 0 auto; /* 실제 사이트를 감싼 wrap 중앙 정렬 */
    width: 944px; /* 실제 사이트의 크기 설정 */
    background-color: #fff; /* wrap 부분 배경 색상 설정 */
  }
  </style>
</head>
</html>
```

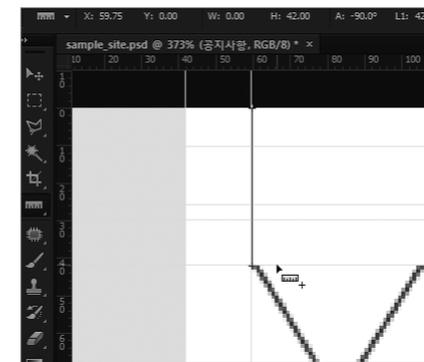
```
a {text-decoration:none;color:#333;} /* 기본 링크 색상 및 밑줄 제거 */
a:hover {color:#39F; text-decoration:none} /* 마우스 오버시 색상 및 효과 적용 */
.clear { clear:both } /* clear 선택자 설정
... 하단 생략 ...
```

● [코드 14-3] reset5.css 적용 및 body, #wrap 등 기본 선택자에 속성 적용

[코드 14-3]까지가 사이트 전체 기본이 되는 뼈대라고 할 수 있습니다.

이제부터는 실제 웹사이트의 요소들인 선택자에 대해 속성을 적용해 보겠습니다.

먼저 포토샵에서 작업된 부분에 정확한 위치를 파악해야만 해당 요소들을 정확하게 배치할 수 있습니다. 포토샵에는 ruler tool이 있어 해당 요소가 정확하게 어느 정도의 위치에 있는지 거리를 잴 수 있습니다. [그림 14-11]에서 회사 로고가 있는 부분을 상단에서부터 재보면 H:42.00으로 표시되어 있는 것을 볼 수 있습니다(그림에서 보이는 포토샵은 단위가 pixel로 설정되어 있는 상태입니다).



[그림 14-11] 포토샵에서 ruler를 이용, 요소들의 정확한 위치 재기

이런 방식을 통해 각 요소들의 위치를 파악할 수 있습니다. 이렇게 파악된 위치를 바탕으로 header 부분에 대한 작업을 하겠습니다.

우선 header 전체의 높이는 137픽셀입니다. 그리고 현재 로고에는 텍스트로 YNG Corp.이라고 되어 있는데, 보통 사이트들은 회사 로고를 클릭하면 항상 홈 화면으로 돌아가게 처리합니다. 따라서 HTML에

```
<div id="logo"><a href="#">YNG Corp.</a></div>
```

와 같이 링크를 걸어줍니다.

| 여기서 잠깐 |

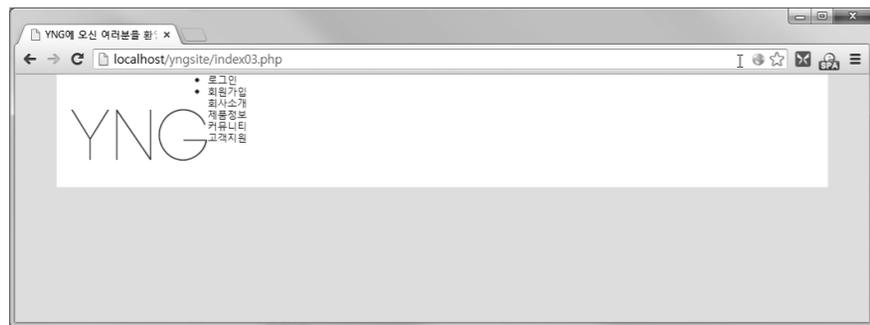
더미(가짜) 링크

#을 이용한 링크는 보통 더미(가짜) 링크라고 합니다. 웹 사이트 제작 시 링크가 되는 부분에 대해 먼저 더미 링크를 이용해서 링크 효과를 확인할 때 주로 사용되며, 사이트가 제작되는 중간에 #을 제거하고 실제 링크로 대체합니다.

그리고 logo 부분에는 다음과 같이 속성을 정의합니다.

```
#logo{
  float: left; /* 왼쪽 정렬 */
  text-indent: -9999px; /* text-indent 속성으로 텍스트 감춤 */
  width:167px; /* 로고 넓이 설정 */
  height:63px; /* 로고 높이 설정 */
  background:url(imgs/logo.png) no-repeat left top;
  /* 배경 이미지로 로고 이미지 사용 */
  margin:42px 0 0 18px; /* 포토샵에서 측정한 마진 값 설정 */
}
```

여기까지의 결과는 [그림 14-12]와 같습니다.



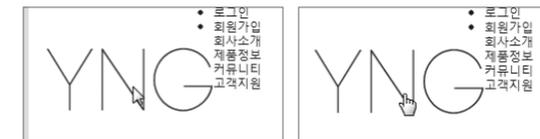
[그림 14-12] 로고 부분 텍스트를 이미지로 교체

하지만 여기서 로고 부분 즉 텍스트에 링크를 걸었는데, 로고 이미지를 클릭해 보면 링크가 되어 있지 않는 것을 확인할 수 있습니다. 이유는 텍스트에 링크는 걸려져 있는데, 텍스트를 감춰버려 링크 대상이 없다는 것입니다. 그럼 이런 경우 어떻게 처리해야 할까요?

방법은 단순합니다. #logo a라는 선택자를 만들어서 #logo에 적용된 width와 height를 #logo a에 적용해 주고 인라인 속성인 #logo a를 display:block 즉 블록 속성으로 변환해 버리면 됩니다. 다음과 같이 적용해 주면 됩니다.

```
#logo{
  float: left;
  text-indent: -9999px;
  background:url(imgs/logo.png) no-repeat left top;
  margin:42px 0 0 18px;
}
#logo a {
  display:block;
  width:167px;
  height:63px;
}
```

이렇게 적용해 주면 로고 부분 전체에 링크가 걸리게 되는 것입니다. [그림 14-13]을 보면 차이점을 알 수 있습니다.



[그림 14-13] #logo a 적용 전(왼쪽 그림)과 적용 후 (오른쪽 그림)

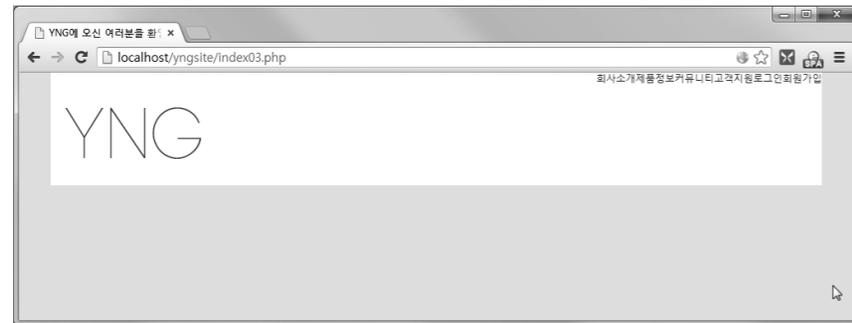
이제 로그인 부분 즉 #hlink라고 되어 있는 부분과 메인 메뉴에 대한 처리를 해 보겠습니다.

[그림 14-13]에서 보면 로그인과 회원가입 부분에는 리스트 스타일이 적용되어 있는데, 회사소개, 제품정보가 있는 메인 메뉴에는 리스트 스타일이 적용되어 있지 않습니다. 그 이유는 reset5.css 파일 내부에 nav ul {list-style:none}이라고 되어 있기 때문에 nav 부분에는 리스트 스타일이 없습니다. 먼저 #hlink와 nav는 오른쪽으로 정렬되어 있습니다. 따라서 그룹 선택자를 사용하여 묶어주고 float:right를 적용합니다. 그리고 #hlink 부분에 list-style:none을 지정합니다.

또한 현재 상하로 배치되어 있는 리스트를 좌우 방향으로 배치하기 위해서 #hlink li와 nav li의 속성을 float:left로 지정합니다. 정리를 하면 다음과 같습니다.

```
#hlink, nav {float:right;}
#hlink ul {list-style:none;}
#hlink li, nav li {float:left;}
```

그에 따른 결과는 [그림 14-14]를 참조하면 됩니다.



[그림 14-14] #hlink 부분과 nav 부분에 대한 처리

[그림 14-14]를 보면 #hlink 부분과 nav 부분이 동시에 float:right가 되어 있어 한 줄에 배치되어 있는 것을 알 수 있습니다. nav 부분은 #hlink 다음 라인에 배치하려 면 nav에만 clear:right를 지정해 주면 문제가 해결됩니다.

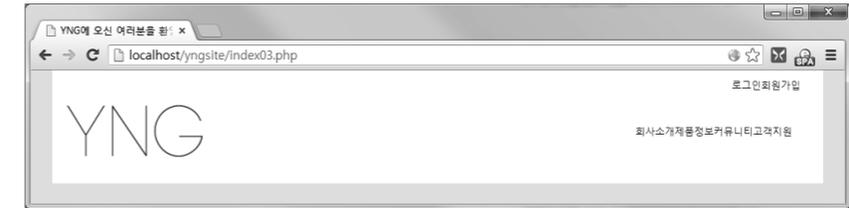


[그림 14-15] nav 선택자에 clear:right 적용하여, 다음 라인으로 배치 후 모습

이제 #hlink와 nav 부분에 대한 마진값을 설정하여 정확한 위치에 배치하도록 하겠습니다.

```
#hlink {margin:11px 28px 0 0;}
nav {margin:42px 38px 0 0; clear:right;}
```

이 속성을 적용한 후 결과는 [그림 14-16]과 같습니다.



[그림 14-16] #hlink와 nav의 마진값을 이용하여 자리 배치 후 모습

이제 각 텍스트 사이의 간격과 메인 메뉴의 크기를 조절하도록 하겠습니다. 텍스트 간격은 패딩을 이용해서 처리합니다. 그리고 메인 메뉴에는 메뉴 간의 구분을 위해 서 구분선이 되어 있는데, 이 부분은 백그라운드 이미지를 이용해서 처리합니다.



[그림 14-17] header 부분 전체 레이아웃 모습

[그림 14-17]을 보면 메인 메뉴 부분에 구분선이 들어가 있는데, 마지막 부분 “고객 지원” 다음에도 구분선이 들어가 있습니다. 원본 디자인에는 이곳에 구분선이 없습니다. 따라서 이 부분은 제거해야 합니다. :last-child라는 수도 선택자를 이용해서 배경 이미지를 없애버리면 됩니다.

여기까지 적용된 CSS 속성은 다음과 같습니다.

```
#hlink, nav {float:right;}
#hlink ul {list-style:none;}
#hlink li, nav li {float:left;}
#hlink {margin:11px 28px 0 0;}
#hlink li a {padding:0 10px}
nav {margin:42px 38px 0 0; font-size:1.4em;clear:right;}
nav li {background:url(imgs/div_line.png) no-repeat center right}
/* 배경 이미지로 구분선 적용 */
nav li a {padding:0 20px} /* 각 메뉴 사이 간격 제거 */
nav li:last-child {background:none} /* 마지막 구분선 배경 이미지 제거 */
```

여기서 원본 포토샵에서는 메인 메뉴의 폰트가 지금까지 작업한 nav 부분과 다릅니다. HTML로 적용된 메뉴는 기본 폰트가 “맑은 고딕”인데 원본 포토샵 파일에는 “제주 고딕”이 사용되었습니다. 그렇다면 이럴 경우 어떻게 처리해야 할까요? 두 가지 방법이 있는데, 그 방법은 다음과 같습니다.

첫 번째는 현재 만들어진 메인 메뉴 부분을 그래픽 이미지로 변경하는 방법이고 두 번째는 “제주 고딕”을 웹 폰트로 변경한 후 @font-family 속성을 이용해서 처리하는 방법입니다.

두 번째 방법이 좋겠지만, 한글 폰트는 웹 폰트로 변경하게 되면 파일 크기가 상당히 크다는 단점이 있습니다. 따라서 “제주 고딕”이라는 글씨체가 현재 웹사이트의 메인 폰트인 경우에는 웹 폰트를 사용해야지만, 현재와 같이 메인 메뉴에만 사용하면 그래픽 이미지로 처리하는 것이 좋습니다.

우선 메인 메뉴를 그래픽 파일로 처리하는데, 메인 메뉴에 마우스 오버 동작 시 일반적으로 색이 변하거나, 애니메이션 효과를 주는 것이 일반적입니다. 이 책에서는 웹 표준 방식을 이용해서 텍스트를 그래픽 파일로 변환하는 세 가지 방법에 대해서 설명해 드리겠습니다.

먼저 메인 메뉴에 사용될 이미지를 준비합니다. 일반적으로 4개의 메뉴가 있으면, 그래픽 파일은 8개를 만듭니다. 각 메뉴에 사용될 이미지 하나, 마우스 오버 시 변화할 메뉴 하나씩 준비합니다.



[그림 14-18] 메인 메뉴에 사용될 그래픽 이미지

우선 웹 표준 방식으로 제작하기 전에 이전 방식 즉 WISIWIG 에디터(대표적으로 드림위버)에서 롤오버 이미지를 삽입하여 메인 메뉴를 제작하는 방법을 보겠습니다. 분명히 이 책을 보는 분들 중에서도 이런 방식으로 지금까지 메인 메뉴 부분을 작업했던 분들이 계십니다.

[그림 14-19]와 같이 드림위버의 메뉴 중에서 **삽입 > 이미지 객체 > 롤오버 이미지**를 이용해서 이미지의 롤오버 효과를 내주는 방법이 있습니다. 하지만 이 방법을 쓰면, 쓸데없는 자바스크립트 코드가 생기고 코드가 아주 지저분해지는 단점이 있습니다.



[그림 14-19] 드림위버에서 이미지 객체 -> 롤오버 이미지 삽입

```

23 nav li a { padding:0; display:block; width:118px; height:20px;text-align:center }
24 nav li:last-child { background:none }
25
26 </style>
27 <script type="text/javascript">
28   function MM_swapImgRestore() { //v3.0
29     var i,x,a=document.MM_sr; for(i=0;a&&i<a.length&&(x=a[i])&&x.oSrc;i++) x.src=x.oSrc;
30   }
31   function MM_preloadImages() { //v3.0
32     var d=document; if(d.images){ if(!d.MM_p) d.MM_p=new Array();
33     var i,j=d.MM_p.length,a=MM_preloadImages.arguments; for(i=0; i<a.length; i++)
34     if (a[i].indexOf("#")!=0){ d.MM_p[j]=new Image; d.MM_p[j++].src=a[i];}}
35   }
36
37   function MM_findObj(n, d) { //v4.01
38     var p,i,x;  if(!d) d=document; if((p=n.indexOf("?"))>0&&parent.frames.length) {
39     d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);}
40     if(!(x=d[a])&&d.all) x=d.all[0]; for (i=0; i&&i<d.frames.length; i++) x=d.frames[i][n];
41     if(i=0; i&&i<d.layers.length; i++) x=MM_findObj(n,d.layers[i].document);
42     if(!x && d.getElementById) x=d.getElementById(n); return x;
43   }
44
45   function MM_swapImage() { //v3.0
46     var i,j=0,x,a=MM_swapImage.arguments; document.MM_sr=new Array; for(i=0; i<(a.length-2); i+=3)
47     if ((x=MM_findObj(a[i]))!=null){document.MM_sr[j++]=x; if(!x.oSrc) x.oSrc=x.src; x.src=a[i+2];}
48   }
49 </script>
50 </head>
51 <body onLoad="MM_preloadImages('imgs/menu1.png','imgs/menu2.png','imgs/menu31.png','imgs/menu41.png')">
52   <div id="wrap">
53     <header>
54       <div id="logo"><a href="#">YNG Corp.</a></div>
55       <div id="hlink">
56         <ul>
57           <li><a href="#">로그인</a></li>
58           <li><a href="#">회원가입</a></li>
59         </ul>
60       </div>
61     </header>
62     <nav>
63       <ul>
64         <li><a href="#" onmouseout="MM_swapImgRestore()" onmouseover="MM_swapImage('image1','','imgs/menu1.png',1)"></a></li>
66         <li><a href="#" onmouseout="MM_swapImgRestore()" onmouseover="MM_swapImage('image2','','imgs/menu21.png',1)"></a></li>
68         <li><a href="#" onmouseout="MM_swapImgRestore()" onmouseover="MM_swapImage('image3','','imgs/menu31.png',1)"></a></li>
70         <li><a href="#" onmouseout="MM_swapImgRestore()" onmouseover="MM_swapImage('image4','','imgs/menu41.png',1)"></a></li>
72       </ul>
73     </nav>
74   </div>

```

[그림 14-20] 드림위버의 롤오버 이미지 사용 시 쓸데없는 코드가 생성된다.

<http://localhost/yngsite/index04.php>

[그림 14-20]에서와 같이 자동으로 자바스크립트 코드가 생성되고 메인 메뉴가 들어가는 부분과 body 태그에도 코드들이 들어가 있습니다. 전체적으로 너무 많은 코드가 생성되어 지저분하게 보입니다.



[그림 14-21] 그래픽 이미지를 이용하여 메인 메뉴 처리 후 모습(드림위버 롤 오버 이미지 사용)

웹 표준 방식을 사용하면, 이런 쓸데 없는 코드는 제거하고, 아주 깔끔하게 코드를 처리할 수 있으며 심지어 메인 메뉴에 있는 텍스트도 유지할 수 있습니다.

먼저 메인 메뉴 부분에 클래스 선택자를 추가해 줍니다.

```
<li class="n1"><a href="#">회사소개</a></li>
<li class="n2"><a href="#">제품정보</a></li>
<li class="n3"><a href="#">커뮤니티</a></li>
<li class="n4"><a href="#">고객지원</a></li>
```

그리고 CSS에서 nav 부분에 text-indent:-9999px를 적용하여 텍스트들을 전부 감춥니다.

그 후에 다음과 같이 각 클래스 선택자 부분에 배경 이미지로 해당 그래픽 이미지를 적용합니다.

```
nav li.n1 a {background:url(imgs/menu1.png) no-repeat top center;}
```

그리고 nav li.n1 a:hover 상태에서 다음과 같이 배경 이미지를 변경합니다.

```
nav li.n1 a:hover {background:url(imgs/menu11.png) no-repeat top center;}
```

그리고 여기에 transition 효과를 적용해 보겠습니다. transition 효과를 적용하면, 메인 메뉴 마우스 오버 시 부드럽게 이미지가 변환되는 효과를 볼 수 있습니다(IE는 효과가 미미함).

메인 메뉴가 적용된 CSS와 HTML 코드는 다음과 같습니다.

[메인 메뉴 CSS 코드]

```
nav {margin:42px 38px 0 0;font-size:1.4em;clear:right;text-indent:-9999px;width:480px;height:20px;}
nav li {background:url(imgs/div_line.png) no-repeat center right}
nav li.n1 a {background:url(imgs/menu1.png) no-repeat top center;transition:all 0.2s ease-in-out;}
nav li.n2 a {background:url(imgs/menu2.png) no-repeat top center;transition:all 0.2s ease-in-out;}
nav li.n3 a {background:url(imgs/menu3.png) no-repeat top center;transition:all 0.2s ease-in-out;}
nav li.n4 a {background:url(imgs/menu4.png) no-repeat top center;transition:all 0.2s ease-in-out;}
```

```
nav li.n1 a:hover {background:url(imgs/menu11.png) no-repeat top center;}
nav li.n2 a:hover {background:url(imgs/menu21.png) no-repeat top center;}
nav li.n3 a:hover {background:url(imgs/menu31.png) no-repeat top center;}
nav li.n4 a:hover {background:url(imgs/menu41.png) no-repeat top center;}
```

```
nav li a { padding:0; display:block; width:118px; height:20px;}
nav li:last-child { background:none}
```

[메인 메뉴 HTML 코드]

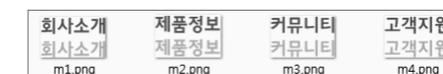
```
<nav>
<ul>
<li class="n1"><a href="#">회사소개</a></li>
<li class="n2"><a href="#">제품정보</a></li>
<li class="n3"><a href="#">커뮤니티</a></li>
<li class="n4"><a href="#">고객지원</a></li>
</ul>
</nav>
```

- [코드 14-4] 메인 메뉴 CSS를 이용하여 그래픽 이미지로 변환

이 방법이 필자가 알려드리는 가장 기본적인 방법 중 첫 번째 방법입니다.

소스코드: <http://localhost/yngsite/index05.php>

두 번째 방법은 그래픽 이미지를 4개를 만들어서 사용하는 방법입니다.



[그림 14-22] 메인 메뉴에 사용되는 4개의 그래픽 이미지

두 번째 방법도 사실 첫 번째 방법과 큰 차이는 없습니다만, 그래픽 이미지를 4개 사용한다는 점과 첫 번째 방법보다 더 큰 효과를 적용할 수 있다는 차이가 있습니다.

바로 CSS 코드를 확인해 보면 다음과 같이 처리하는 것입니다.

```
nav li.n1 a {background:url(imgs/m1.png) no-repeat center top;}
```

기본 상태인 경우 배경 이미지의 위치는 center top 상태입니다. 하지만 마우스 오버시에는 center -23px 상태입니다.

```
nav li.n1 a:hover {background:url(imgs/m1.png) no-repeat center -23px;}
```

즉 마우스 오버 시 배경 이미지의 상태에 변화를 줌으로써, 단순하게 4개의 이미지로도 동일한 효과를 낼 수 있다는 것입니다. 여기에 `transition:all 0.2s ease-in-out;` 효과를 적용해 주면, 아주 큰 효과를 낼 수 있습니다. 직접 한번 예제 파일을 실행해 보기 바랍니다.

소스코드: <http://localhost/yngsite/index06.php>

마지막 세 번째 방법은 하나의 이미지를 이용해서 메인 메뉴를 만드는 것입니다.



[그림 14-23] 메인 메뉴에 사용되는 하나의 그래픽 이미지

하나의 그래픽 파일을 이용할 경우 장점이 브라우저에서 여러 개의 파일을 불러오는 경우보다 더 빨리 이미지를 불러 올 수 있으며, 한번 브라우저에서 로딩이 된 후에도 아주 작은 차이지만, 처리 속도가 향상되는 장점이 있습니다.

그래픽 파일 하나만으로 메인 메뉴를 처리하는 방법도 사실 두 번째 방법과 유사한 방법으로 처리하는 것입니다. 하지만 차이점은 이미지 파일을 하나만 사용한다는 것입니다.

이미지 파일을 하나만 사용하기 때문에 이전 방법에서 각 메뉴에 적용되었던 이미지 파일들을 그룹 선택자를 이용해서 다음과 같이 한 번만 사용하면 됩니다.

```
nav li.n1 a, nav li.n2 a, nav li.n3 a, nav li.n4 a, nav li.n1 a:hover, nav li.n2 a:hover, nav li.n3 a:hover, nav li.n4 a:hover {
    background-image:url(imgs/main_menu.png);
    background-repeat:no-repeat;
}
```

이제 각 메인 메뉴에 적용된 클래스 선택자에 해당 배경 이미지의 위치를 다음과 같이 적용합니다.

```
nav li.n1 a {background-position:27px top;}
nav li.n2 a {background-position:-96px top;}
nav li.n3 a {background-position:-217px top;}
nav li.n4 a {background-position:-338px top;}
nav li.n1 a:hover {background-position: 27px -23px;}
nav li.n2 a:hover {background-position: -96px -23px;}
nav li.n3 a:hover {background-position: -217px -23px;}
nav li.n4 a:hover {background-position: -338px -23px;}
```

마지막으로 `transition` 효과를 적용합니다.

```
nav li.n1 a, nav li.n2 a, nav li.n3 a, nav li.n4 a{transition:all 0.2s ease-in-out;}
```

이렇게 처리한 결과는 소스코드: <http://localhost/yngsite/index07.php>에서 확인할 수 있습니다.

어떻게 보면 세 번째 방법이 가장 편리할 수도 있습니다만, 배경 이미지의 위치를 잡는 번거로움이 만만치 않는 단점이 있습니다.

지금까지 설명한 세 가지 방법을 적절히 이용하면, 책의 예제보다 더 멋진 효과를 구현할 수 있습니다. 직접 여러분들이 한 번씩 예제를 기반으로 더 많은 연습을 해 보기 바랍니다.

이제 jQuery를 이용해서 슬라이더 이미지 부분을 처리해 보도록 하겠습니다.

슬라이더 이미지 부분은 3부 2장 jQuery에서 소개했던 Nivo Slider를 이용하도록 하겠습니다.

Nivo Slider는 사용법도 편리하고, 세계적으로 가장 많이 사용되는 jQuery 슬라이더 중 하나입니다. Nivo Slider는 <http://dev7studios.com/nivo-slider/>에서 다운 받을 수 있습니다.

다운받은 Nivo Slider 패키지의 압축을 풀어 scripts 폴더에는 jquery.nivo.slider.js 파일을 css 폴더에는 nivo-slider.css 파일 및 themes라는 폴더를 넣어 줍니다. 그리고 index.php 파일에 해당 효과를 적용해 주면 됩니다.

먼저 Nivo Slider를 구동시키는 두개의 CSS 파일을 link 속성을 이용해 넣어 줍니다.

```
<link href="css/themes/default/default.css" rel="stylesheet" type="text/css">
<link href="css/nivo-slider.css" rel="stylesheet" type="text/css">
```

그리고 Nivo Slider가 들어가는 부분에는 다음과 같은 선택자를 추가하고 4개의 이미지 파일을 넣어줍니다.

```
<div class="slider-wrapper theme-default">
  <div id="slider" class="nivoSlider">
    
    
    
    
  </div>
</div>
```

마지막으로 Nivo Slider를 구동하게 해 주는 jQuery와 Nivo Slider 스크립트 파일을 넣어줍니다.

여기서 jQuery는 CDN을 이용, 항상 최신 버전의 jQuery를 호출하게 처리하겠습니다.

```
<script type="text/javascript" src="http://code.jquery.com/jquery-latest.js"></script>
<script type="text/javascript" src="scripts/jquery.nivo.slider.js"></script>
<script type="text/javascript">
$(window).load(function() {
  $('#slider').nivoSlider();
});
</script>
```

이렇게 처리하면 다음과 같은 결과를 얻을 수 있습니다.



[그림 14-24] Nivo Slider를 이용한 슬라이더 부분 _ <http://localhost/yngsite/index08.php>

HTML 구조에서 슬라이더 이미지가 끝나 부분에 `<div class="clear"></div>`를 추가하여 확실하게 구분을 합니다.

이제 슬라이더 하단에 배치된 Just Arrived 부분에 대해서 처리해 보겠습니다. 이 부분은 단순한 구조를 지녔습니다. 먼저 HTML 구성부터 살펴보겠습니다.

```
<div id="justar">
  <h3>just arrived</h3>
  <figure></figure>
</div>
```

#justar라는 아이디 선택자로 박스를 만들고 내부에는 h3 태그를 이용하여 제목을 figure 태그로 이미지를 감싼 형태입니다.

CSS를 이용해서 디자인을 잡습니다. 현재 "Just Arrived"라는 텍스트는 Droid Sans 라는 공개 폰트를 사용하고 있는데, 직접 웹 폰트로 변환해도 되고, 아니면 google fonts 서비스를 이용해도 됩니다(<http://www.google.com/fonts/>).

이 책에서는 google fonts 서비스를 이용하도록 하겠습니다.

google fonts 서버를 이용하는 방법은 상당히 간단합니다. 해당 폰트가 들어있는 css 파일을 링크 형식으로 HTML 파일 내부에 넣어줍니다.

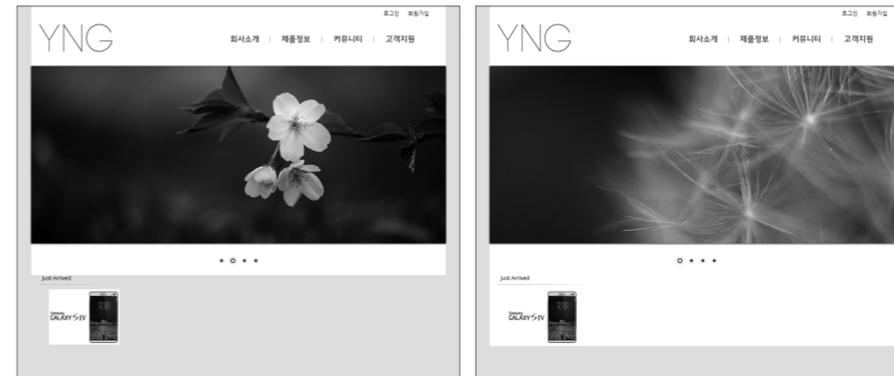
```
<link href='http://fonts.googleapis.com/css?family=Droid+Sans' rel='stylesheet' type='text/css'>
```

그리고 해당 폰트가 사용되는 선택자에 "font-family: 'Droid Sans', sans-serif;" 를 적용해 주면 됩니다.

이제 justar 부분에 대한 CSS 파일을 확인해 보겠습니다.

```
#justar {
margin:0 0 0 20px; /* 왼쪽 부분 여백 20픽셀 */
float:left; /* 왼쪽으로 배치 */
width:200px; /* 박스 크기 200픽셀 */
}
#justar h3 {
text-transform:capitalize; /* 영문 첫 글자만 대문자로 */
background:url(imgs/h3_under.jpg) no-repeat left bottom;
/* 백그라운드 이미지를 이용하여 제목 부분 장식 추가 */
padding:0 0 8px 5px; /* 패딩으로 왼쪽과 하단 부분 공간 적용 */
font-family: 'Droid Sans', sans-serif; /* 구글 폰트 사용 */
font-weight:normal;
/* h로 시작하는 태그는 전부 bold 속성을 지니기 때문에 폰트 두께를 normal로 설정 */
font-size:1.1em;
}
#justar figure {
text-align:center; /* text-align은 글자뿐만 아니라 그림도 정렬됨, 중앙 정렬 */
padding-top:10px; /* 위쪽 부분 패딩 이용 10픽셀 공간 적용 */
}
```

여기까지 HTML과 CSS를 적용한 결과는 [그림 14-23]에서 보실 수 있습니다. 하지만 여기서 조금 이상한 것이 Just Arrived 부분에는 배경 색상이 제대로 적용되지 않고 슬라이드 이미지까지만 적용된 것을 알 수 있습니다. 이 부분은 단순히 <div class="clear"></div>으로 다시 닫아 주면 문제가 해결됩니다(그림 14-25 우측 그림 참조).



[그림 14-25] Just Arrived 부분 작업, <div class="clear"></div> 적용 전 : 왼쪽, 적용 후 : 오른쪽
<http://localhost/yngsite/index09.php>

| 여기서 잠깐 |

사이트 제작 시 코드 및 영문 텍스트는 소문자로

영문 텍스트인 경우 소문자로 만드는 것이 좋습니다. HTML 코딩 및 CSS 코딩에서도 소문자가 편리하고 HTML 코드 및 텍스트인 경우에도 소문자를 사용하는 것이 에러를 줄이는 방법이기도 합니다. 또한 HTML 문서 내부에 있는 영문이 소문자이면 CSS에서 text-transform 속성을 이용하여 대문자 또는 단어의 첫 글자만 대문자로 변환 가능하기 때문입니다.

이제 화면 중앙에 있는 Notice와 News 부분을 처리하도록 하겠습니다. 이 부분은 tab 메뉴 처리가 되어 있는데, 가장 많이 사용하는 효과 중 하나입니다. 이 부분 또한 jQuery를 이용해서 처리하도록 하겠습니다. jQuery 소스는 <http://www.html5drive.net/items/show/542/Simple-Tabs-w-CSS-jQuery>에서 참조합니다.



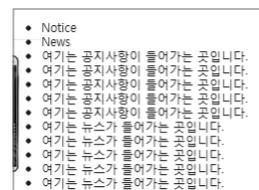
[그림 14-26] Simple Tabs w/ CSS & jQuery

해당 소스 파일을 다운로드 하진 마시고, 책에 있는 설명만 보시면 됩니다^^). 왜냐하면 소스 파일에는 특별한 소스가 있는 것이 아니고 jQuery 파일만 있습니다.

먼저 탭 메뉴가 들어 가는 부분은 #notinews라고 정의한다고 했습니다. 따라서 HTML 문서에는 다음과 같이 설정합니다.

```
<div id="notinews"><!-- 탭 부분 전체를 감싸는 아이디 선택자 -->
<div class="container">
  <ul class="tabs"><!-- 탭 부분 제목 -->
    <li><a href="#tab1">Notice</a></li>
    <li><a href="#tab2">News</a></li>
  </ul>
  <div class="tab_container">
    <div id="tab1" class="tab_content"><!-- 각 탭의 내용 -->
    <ul>
      <li><a href="#">여기는 공지사항이 들어가는 곳입니다.</a></li>
      ... 중략 ...
    </ul>
  </div>
    <div id="tab2" class="tab_content"><!-- 각 탭의 내용 -->
    <ul>
      <li><a href="#">여기는 뉴스가 들어가는 곳입니다.</a></li>
      ... 중략 ...
    </ul>
  </div>
  </div>
</div>
</div>
```

HTML 코드만으로는 [그림 14-27]과 같은 결과를 얻습니다.



[그림 14-27] 단순한 HTML 구조로 본 tab 메뉴

이제 Simple Tabs w/ CSS & jQuery에서 제공되는 CSS 속성을 적용해 보겠습니다.

원본 파일에서 디자인적으로 불 필요한 부분은 실제 코드에서는 주석 처리하겠으며, 책에서는 주석 처리 및 취소선으로 표시하겠습니다.

```
/* .container {width: 500px; margin: 10px auto;} */ /* 원본 박스 제거 */
notinews { /* Notice 와 News 박스 처리 .container 대체함. */
width: 350px; /* 크기 350픽셀로 고정 */
float:left; /* 왼쪽 정렬 */
margin-left:80px; /* 왼쪽으로 마진 80픽셀 적용 */
}
ul.tabs { /* Notice 와 New 탭 부분 ul 처리 */
margin: 0;
padding: 0;
float: left;
list-style: none;
height: 24px; /* 탭 부분 전체 높이 설정 - 다음 페이지 <여기서 잠깐> 참조 */
border-bottom: 1px solid #999; /* 제목 부분 밑 줄 적용 */
border-left: 1px solid #999; /* ul 부분 왼쪽에만 선 적용 */
width: 100%;
font-family: 'Droid Sans', sans-serif;
}
ul.tabs li {
float: left;
margin: 0;
padding: 0;
height: 31px; */
height: 23px; /* 리스트에서는 1픽셀 작은 값 적용 - 다음 페이지 <여기서 잠깐> 참조 */
line-height: 31px; */
line-height: 21px;
border: 1px solid #999; /* 전체 선 적용 */
border-left: none; /* ul에서 선이 적용되어 여기서 선을 제거함 */
margin-bottom: -1px; /* 여기서 li 부분을 1픽셀 끌어 내림 - 다음 페이지 <여기서 잠깐> 참조 */
background: #F6F6F6;
overflow: hidden;
position: relative;
}
ul.tabs li a {
text-decoration: none;
color: #000;
display: block;
padding: 0 20px;
border: 1px solid #fff;
outline: none;
}
```

```

ul.tabs li a:hover {
  background: #ccc;
}
html ul.tabs li.active, html ul.tabs li.active a:hover {
  background: #fff;
  border-bottom: 1px solid #fff; /* active 상태일 때 밑줄 흰색 처리 */
}
.tab_container {
  /*border: 1px solid #999;*/ /* 원래는 전체 박스에 선 처리되어 있음 */
  border-top: none; /* 디자인을 위해 내용 부분 선 제거 */
  clear: both;
  float: left;
  width: 300px;
  background: #fff;
  /* moz-border-radius-bottomright: 5px;
  khtml-border-radius-bottomright: 5px;
  webkit-border-bottom-right-radius: 5px;
  moz-border-radius-bottomleft: 5px;
  khtml-border-radius-bottomleft: 5px;
  webkit-border-bottom-left-radius: 5px;*/
}
.tab_content {
  /* padding: 20px;
  font-size: 1.2em;*/
  padding: 20px 10px;
  font-size: 1.1em;
}
/* .tab_content h2 {
  font-weight: normal;
  padding-bottom: 10px;
  border-bottom: 1px dashed #ddd;
  font-size: 1.8em;
}
.tab_content h3 a {
  color: #254588;
}
.tab_content img {
  float: left;
  margin: 0 20px 20px 0;
  border: 1px solid #ddd;
  padding: 5px;
}*/
.tab_container ul {
  list-style:none;
}

```

```

.tab_container ul li {
  line-height:1.7em;
  background:url(imgs/bullet.gif) no-repeat left 10px;
  padding-left:10px;
}

```

[여기서 잠깐](#)

탭 메뉴 모양 설정

탭 메뉴의 높이는 ul.tabs에서 24픽셀로 설정되어 있고 ul.tabs li에서는 23픽셀로 설정되어 있으며, margin-bottom: -1px;를 이용 최종적으로 24픽셀로 세팅이 되어 있습니다. 왜 이렇게 처리하는가 하면 ul.tabs에 설정된 border-bottom: 1px solid #999; 속성과 선의 높이를 맞추기 위함입니다. 만약 ul.tabs와 ul.tabs li의 높이를 동일하게 24px로 설정하고 margin-bottom: -1px;을 적용하지 않으면 다음과 같은 현상이 벌어집니다. 이런 현상을 방지하기 위해 ul.tabs와 ul.tabs li의 높이를 1픽셀 정도의 차이를 준 후 margin-bottom: -1px 이용 높이를 맞춰주는 것입니다.



ul.tabs에는 height: 24px; ul.tabs li에는 height:24px; 적용 시 모습



ul.tabs에는 height: 24px; ul.tabs li에는 height:23px;와 margin-bottom: -1px; 적용 후

[그림 14-28] 탭 메뉴 모양 설정을 위한 CSS 속성

현재 CSS만 설정한 상태에서는 다음과 같은 결과를 얻을 수 있습니다.



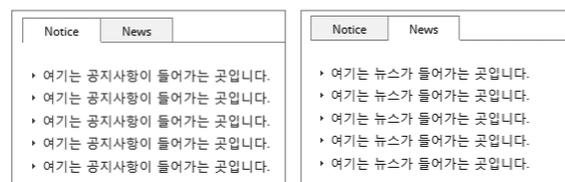
[그림 14-29] jQuery 적용 전 HTML과 CSS만 설정해 준 상태

여기서 jQuery를 이용하여 첫 부분 즉 Notice는 보여주고 News 부분은 감춘 상태로 그리고 tab을 클릭하면 해당 탭이 활성화 되는 코드를 입력하겠습니다.

현재 HTML 문서 맨 마지막 부분에 다음의 코드를 입력합니다.

```
$(document).ready(function() { // 문서 준비 단계
//Default Action
$(".tab_content").hide(); // 콘텐츠 전부 감춤
$("ul.tabs li:first").addClass("active").show(); // 첫 번째 탭 활성화
$(".tab_content:first").show(); // 첫 번째 탭 보여줌
$("ul.tabs li").click(function() { //해당 탭을 클릭했을 때 이벤트
    $(".ul.tabs li").removeClass("active"); // 클래스 선택자 active 제거
    $(this).addClass("active"); // 선택한 탭에 "active" 클래스 선택자 추가
    $(".tab_content").hide(); // 모든 탭 콘텐츠 감춤
    var activeTab = $(this).find("a").attr("href");
    // 변수 activeTab에는 active 탭과 콘텐츠 부분을 식별하는 값 적용
    $(activeTab).fadeIn(); // 탭 클릭 시 activeTab(활성 탭과 내용 부분) 페이드인 효과
    return false;
});
});
```

이렇게 jQuery를 적용해 주면 [그림 14-30]의 결과를 얻을 수 있습니다.



[그림 14-30] 완성된 tab 메뉴 <http://localhost/ynsite/index10.php>

이제 social network 부분에 대한 작업을 진행해 보겠습니다. 이 부분은 제목을 제외하고는 이미지 파일로 처리되어 있습니다. 여기는 사이트의 로고 및 메인 메뉴와는 달리 실제 아이콘을 삽입하는 것이 좋습니다.

이 부분은 #social이라는 아이디 선택자를 이용해서 박스 모델을 구성하겠습니다. 또한 모든 아이콘들을 리스트 태그를 이용해서 정렬하도록 하겠습니다.

HTML 태그 구성은 다음과 같습니다.

```
<div id="social">
<h3>social network</h3>
<ul>
<li><a href="#"></a></li>
<li><a href="#"></a></li>
... 종략 ...
</ul>
</div>
```

HTML 문서로만 구성된 상태에서는 다음과 같은 결과를 얻을 수 있습니다.



[그림 14-31] 리스트 태그만 적용했을 경우 Social Network 부분

이제 여기에 CSS를 적용하도록 하겠습니다.

```
#social {
float:right; /* 박스를 오른쪽 정렬합니다. */
margin:0 30px 0 0; /* 박스 오른쪽 여백을 설정합니다. */
width: 180px; /* social 박스의 크기를 정해 줍니다. 박스의 크기를 정해주면 왼쪽으로 float된 아이콘들이 박스 크기에 맞게 정렬됩니다. */
}
#social h3 { /* social 제목 설정 부분 */
padding:0 0 15px 0;
font-family: 'Droid Sans', sans-serif;
font-weight:normal;
font-size:1.1em;
text-transform:capitalize;
}
#social ul {list-style:none;} /* 리스트 스타일 제거 */
#social ul li {float:left;margin:5px;} /* li float 적용하여 좌우로 배치, 각 li 별 마진은 5픽셀 적용 */
```