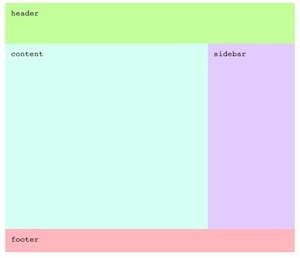
**[HTML&CSS] float 속성의 이해와 웹 페이지 레이아웃 잡기**

  이번 글에서는 CSS의 float 속성을 이용하여 웹 페이지의 레이아웃을 잡는 방법에 대해 알아보겠습니다. 이번 글에서 우리의 목표는 블로그와 같은 웹 페이지에서 자주 쓰이는 header, content, sidebar, footer로 구성된 2단 레이아웃을 만드는 것입니다. 즉 다음과 같은 형태의 레이아웃을 잡는 것이 우리의 목표입니다.



  위와 같은 레이아웃을 잡기 위한 HTML과 CSS의 기본 구조은 다음과 같습니다. <div>으로 header, content, sidebar, footer 영역을 나누어 놓았고 각 영역의 크기와 배경색을 정해주었습니다.

layout.html



layout.css



  지금까지의 결과를 브라우져에서 확인해보면 다음과 같습니다.



  HTML은 기본적으로 웹 페이지의 가장 위에서부터 엘리먼트들이 아래쪽으로 하나씩 붙어나가는 형태이기 때문에 위와 같은 결과가 나오게 됩니다. 우리가 원하는 결과는 sidebar가 content의 오른쪽에 붙어있는 모양이지요. 그럼 이제 sidebar를 content의 오른쪽으로 붙이기 위해 float 속성을 이용해보도록 하겠습니다.

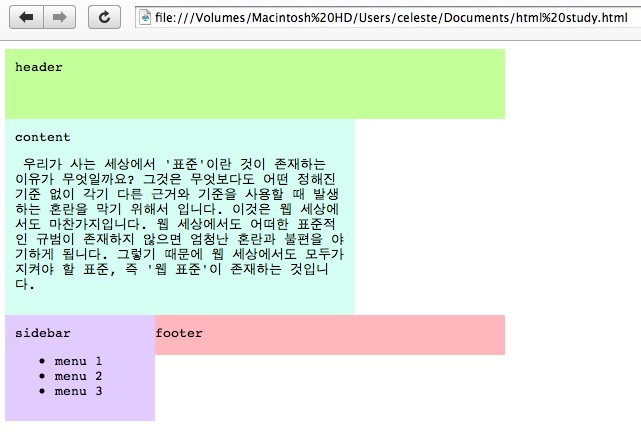
  float 속성은 그 값으로 left 또는 right를 갖습니다. float 속성이 붙은 엘리먼트는 웹 페이지 바닥에 딱 붙어서 고정되어있는 것이 아니라 바닥에서 살짝 떠서(실제로 떠있는 것처럼 보인다는 것이 아니라 의미적으로 그렇다는 것입니다^^) 자신이 속한 container의 가장 왼쪽 또는 가장 오른쪽으로 붙게 됩니다. **float 속성이 붙어있는 엘리먼트는 말 그대로 둥둥 떠있는 상태이기 때문에 그 바로 다음에 오는 엘리먼트는 그 밑으로 비집고 들어가 자리를 잡을 수 있게 됩니다.**

  우리의 예제에서 content div에 float: left; 속성을 넣어주면 content는 바닥에서 살짝 뜬 채로 화면의 왼쪽에 붙어있게 되고 sidebar는 content의 밑으로 들어가게 될 것입니다. **그러나 이때 주의할 점이 있습니다. float된 엘리먼트 밑으로 비집고 들어가긴 들어가되 그 내용은 결코 가려지지 않습니다.** 글로만 봐서는 무슨 뜻인지 잘 이해가 가지 않으실겁니다. 실제 예를 통해 다시 한번 설명드리겠습니다. 먼저 content에 float:left; 속성을 넣었을 때의 결과를 봅시다.



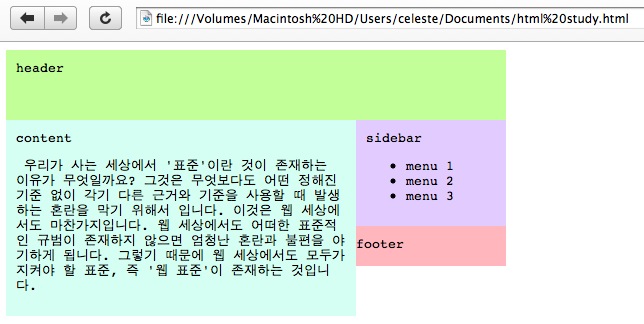
  언뜻 보기에 위의 결과와 차이가 전혀 없는 것으로 보일 수도 있으나 sidebar 영역을 잘 보시면 아까와 다르게 "sidebar"라는 글자가 content 영역 바로 아래 딱 붙어 있음을 알 수 있습니다. 이런 결과가 나온 이유는 content 영역이 float되어 있으므로 sidebar 영역이 content 영역 밑으로 들어가게 되는데 sidebar 영역의 내용("sidebar", "menu 1", "menu 2", "menu 3")이 content 영역에 가려지지 않을 정도까지만 들어간 것입니다.

  조금 이해가 되시나요? 이해를 돕기 위해 또 다른 예를 보겠습니다. 이번에는 sidebar 영역에 float: left; 속성을 주고 footer 영역이 어떻게 되는지 확인해보겠습니다.



  아까의 경우처럼 footer 영역이 sidebar 영역 밑으로 들어가되 "footer"라는 내용이 가리지 않을 정도까지만 들어갈 것이라고 예상하신 분들이 많으실 겁니다. 그러나 이번엔 아까와는 많이 다른 결과가 나왔습니다. "footer"라는 글자가 이상한 위치에 가있지요? 이런 결과가 나온 이유는 아까의 경우와는 달리 "footer"라는 내용이 옆으로 움직여 자리를 잡을 수 있는 여유 공간이 있기 때문입니다. 아까의 경우 sidebar 영역은 width가 좁아서 그 내용이 옆으로 움직일 공간이 없었지만, 이번 경우에는 footer 영역의 width가 넓어서 "footer"라는 내용이 sidebar 영역에 의해 가려지지 않고 옆으로 피할 공간이 있었기 때문에 이와 같은 결과가 나오게 된 것이죠.

  자 이제 float 속성의 특성에 대해 간략하게나마 알아보았으니 다시 본론으로 돌아와 sidebar를 content의 오른쪽에 붙여보도록 하겠습니다. 여러가지 방법이 있지만 우리는 content 영역에 float: left; 속성을 걸고 sidebar에 왼쪽 margin을 두어 목적을 달성해보도록 하겠습니다. content 영역에 float: left; 속성이 붙였을 때의 결과는 아까 위에서 보았습니다. 그 상황에서 sidebar에 왼쪽 margin을 content 영역의 너비만큼, 즉 350px 만큼 넣어주면 sidebar가 content의 오른쪽에 붙게 됩니다.

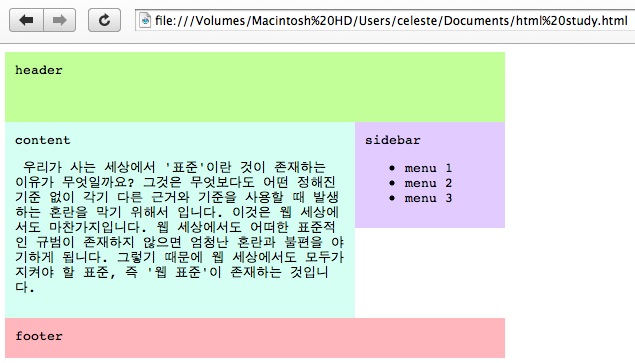


  그러나 아직 문제가 있군요. footer가 content이나 sidebar 보다 아래로 내려와야하는데 지금 상태는 footer가 content 밑에 깔려있습니다. 이 문제를 해결하기 위해서 footer 영역에 clear라는 속성을 붙여주어야 합니다.

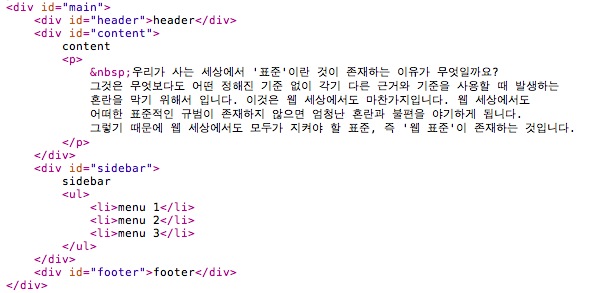
  clear는 float 속성이 걸려있는 엘리먼트들로부터 영향을 받고 싶지 않을 때 사용하는 속성으로서 그 값으로 left, right, both를 가집니다. **즉 clear 속성을 가진 엘리먼트는 다른 엘리먼트들의 float 속성을 무시하고 원래 흐름대로(처음에 잠시 언급했듯이 엘리먼트들이 위에서부터 아래로 차곡차곡 붙어나가는 것이 HTML의 기본 흐름입니다) 자신의 자리를 찾게 됩니다.** 위의 경우 footer 영역이 float: left 속성을 갖고 있는 content 영역의 영향을 받아 content의 밑으로 들어가게 된 것이므로 footer 영역에 clear: left(clear: both도 상관없겠죠^^) 속성을 넣어주면 footer 영역이 content 영역에 밑에 깔리지 않고 content 영역 아래쪽에 자리를 잡게 됩니다.(만약 sidebar 영역이 content 영역보다 더 길었다면 footer는 sidebar 아래에 붙게 되겠죠^^)

  그럼 최종 결과를 보시겠습니다.





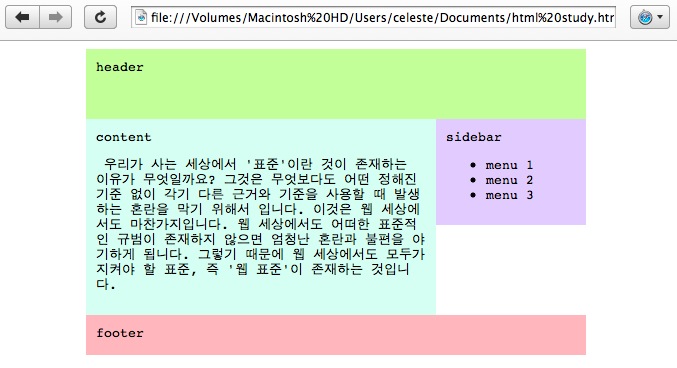
  자 이렇게 해서 header, content, sidebar, footer를 가진 기본적인 2단 레이아웃이 완성 되었습니다. 그런데 아직 한가지 문제가 남았습니다. 지금 상태로는 브라우져의 너비가 넓건 좁건 header, content, sidebar, footer를 포함하는 우리 웹 페이지의 메인 부분이 항상 브라우져의 왼쪽에 붙어있다는 것입니다. 브라우져의 너비에 관계없이 웹 페이지의 메인 부분이 브라우져의 중앙에 위치하면 좋을텐데 말이죠. 이를 위해서는 header, content, sidebar, footer 전체를 감싸는 최상위 div과 margin: auto라는 속성을 이용하면 됩니다. **엘리먼트에 margin: auto라는 속성이 붙어있으면 브라우져가 알아서 해당 엘리먼트를 container의 중앙에 배치시켜 줍니다.** 먼저 html 파일을 보겠습니다.



  위와 같이 body의 최상위에 header, content, sidebar, footer를 포함하는 main이라는 div를 만들어 주었습니다. 다음으로 css 파일에는 다음의 코드를 추가합니다.

http://pds17.egloos.com/pds/201002/19/67/d0068567_4b7e42fd37d35.jpg

  main div의 width를 고정한 뒤 margin: auto 속성을 넣어주었습니다. 이제 그 결과를 보겠습니다.



  이제는 우리가 원했던 것처럼 우리 웹 페이지의 메인 부분이 브라우져의 중앙에 위치하게 되었습니다.

  이번 글에서 제가 소개한 방법말고도 다른 수많은 방법으로 같은 형태의 레이아웃을 얻을 수 있습니다. 또는 더 멋지고 좀더 복잡한 레이아웃을 만들 수도 있겠지요. **중요한 것은 오늘 공부한 float와 clear 속성을 잘 이해하고 있어야한다는 점입니다.** float와 clear의 작동 원리만 머릿속에 명확히 숙지하고 있다면 어떤 형태의 레이아웃이라도 여러분 스스로 쉽게 만들어낼 수 있을 것입니다^^