**CSS float 속성 이해**

*CSS float 속성을 자세히 살펴보자*

[Michael Russell](http://www.ibm.com/developerworks/kr/library/wa-css/index.html#author1), 시스템 아키텍트, Vicki Fox Productions, Inc.



Michael Russell은 물리학에서 학사 학위를, 전자계산학에서 석사 학위를 소지하고 있다. 그는 물류 엔지니어이자, 기술 서비스 관리자이자, 지난 14년 동안 IBM이 인증한 IT 아키텍트다. 현재는 군수 항공 물류와 시뮬레이션 업계에서 시스템 아키텍트로 일한다. 윈도우, 유닉스, OS/400 환경에 익숙하며, 회사 웹 사이트를 통해 재미삼아 웹 기술을 실험한다.

**요약:** CSS float 속성은 웹 디자이너들이 웹 페이지 레이아웃을 설계하는 과정에서 자주 사용하는 도구입니다. 하지만 제대로 이해하지 못하고 사용하는 경우가 많고, 브라우저마다 지원하는 방식이 달라서 문제가 생기는 경우도 많습니다. 이 기사에서는 float 속성을 정확히 이해하고, 널리 사용되는 방식을 소개하고, 개발 도구 고려 사항과 브라우저마다 나타나는 차이점을 논합니다.

옛날 옛적 인간이 처음 글을 쓰기 시작한 시절부터 우리는 이미지 주변에 뭔가를 끼적여 왔다.

이렇듯 이미지 주변을 둘러싸는 문자열의 미학과 공간 절약을 이해한 초창기 웹 개척자들은 넷스케이프(Netscape) 1.0과 HTML 2.0에 "부유(float)" 기능을 추가했다. 이후로 페이지에서 요소를 부유시키는 기능은 HTML 표준과 CSS 명세에서 기본으로 제공하는 속성이 되었다.

"float" 속성의 정의 자체는 매우 간단하다. 그러나 속성을 구현하고 사용하는 방식이 다양한 탓에 많은 웹 디자이너들이 원하는 페이지를 얻느라 많은 시간을 낭비한다.

이 기사에서는 "float" 속성에 관해 다음 측면을 살펴본다.

* 정의와 개념 모델
* 일반적인 사용 방법
* 개발 도구 고려 사항
* 브라우저 고려 사항

"float" 속성이란?

CSS2 명세는 float 속성을 명확히 정의하고 동작 방식도 규정한다. 이 절에서는 CSS2 명세 9장 5절 화면 구도 모델(Visual Fomatting Mode)에 기술된 내용을 토대로 float 속성을 살펴본다(CSS2 명세 링크는 [참고자료](http://www.ibm.com/developerworks/kr/library/wa-css/index.html#resources)에 있다).

정의

CSS2 8장 박스 모델에 나오듯이 float 속성은 박스를 부유시킬 위치를 지정한다. 지정할 수 있는 위치는 왼쪽(left) 또는 오른쪽(right)이다. 박스를 생성하는 요소 중 절대적으로 위치가 지정되지 않은 요소는 언제든 float 속성을 설정할 수 있다.

지정할 수 있는 속성 값은 다음과 같다.

* **inherit** - 요소를 감싸는 바깥 요소에서 float 속성을 상속받는다. 기본 값이다.
* **left** - 요소는 왼쪽에 부유하는 블록 박스를 생성한다. 페이지 내용은 박스 오른쪽에 위치하며 위에서 아래로 흐른다. 이후 요소에 clear 속성이 있으면 페이지 흐름이 달라진다. "none"이 아니라면 "display" 속성은 무시된다.
* **right** - 요소는 오른쪽에 부유하는 블록 박스를 생성한다. 페이지 내용은 박스 왼쪽에 위치하며 위에서 아래로 흐른다. 이후 요소에 clear 속성이 있으면 페이지 흐름이 달라진다. "none"이 아니라면 "display" 속성은 무시된다.
* **none** - 요소를 부유시키지 않는다.

동작 방식

다음은 float 속성이 따르는 규칙이다. 여기서는 왼쪽에서 부유하는 경우만 설명했다. 오른쪽에서 부유하는 경우는 정반대다. 즉, '왼쪽'을 '오른쪽'으로 바꾸면 된다.

명세에서는 박스 모서리(box edge)를 이용하여 동작 방식을 정의한다. 다음은 CSS 박스 모델에서 정의하는 모서리 네 가지다.

* **내용 모서리(content edge)** 또는 **내부 모서리(inner edge)**는 요소 내용을 감싼다. 내용에 가장 가까운 모서리다.
* **패딩 모서리(padding edge)**는 요소에 적용된 패딩을 포함한다. 패딩 모서리는 용기 블록(containing block) 모서리를 정의한다.
* **테두리 모서리(border edge)**는 요소에 적용된 테두리를 포함한다.
* **여백 모서리(margin edge)** 또는 **외부 모서리(outer edge)**에는 요소에 적용된 여백이 포함된다.

**용기 블록 수평 조건**: 왼쪽 부유 박스의 왼쪽 마진 모서리는 용기 블록의 왼쪽 패딩 모서리보다 왼쪽에 위치하지 못한다.

**용기 블록 수직 조건**: 왼쪽 부유 박스의 위쪽 마진 모서리는 용기 블록의 위쪽 패딩 모서리보다 위쪽에 위치하지 못한다.

**이전 부유 조건**: 현재 박스가 왼쪽 부유 박스이고 이전 박스도 왼쪽 부유 박스라면 1) 현재 박스의 왼쪽 마진 모서리는 이전 박스의 오른쪽 마진 모서리보다 오른쪽에 있거나 2) 현재 박스의 위쪽 마진 모서리는 이전 박스의 아래쪽 마진 모서리보다 아래쪽에 있어야 한다.

**겹침 없음 조건**: 같은 수평 영역을 공유하는 왼쪽 부유 박스의 오른쪽 마진 모서리는 오른쪽 부유 박스의 왼쪽 마진 모서리보다 오른쪽에 위치하지 못한다.

**이전 요소 수직 조건**: 왼쪽 부유 박스의 위쪽 마진 모서리는 이전 블록이나 이전 부유 블록의 위쪽 마진 모서리보다 위쪽에 위치하지 못한다.

**라인 박스 수직 조건**: 왼쪽 부유 박스의 위쪽 마진 모서리는 이전 요소가 생성하는 박스를 포함하는 라인 박스의 위쪽 패딩 모서리보다 위쪽에 위치할 수 없다. 라인 박스(line box)란 용기 블록 내에서 일직선을 이루는 인라인 블록을 모두 포함하는 가상의 직사각형을 가리킨다. 라인 박스 위쪽 패딩 모서리 높이는 가장 높은 박스의 높이와 일치한다.

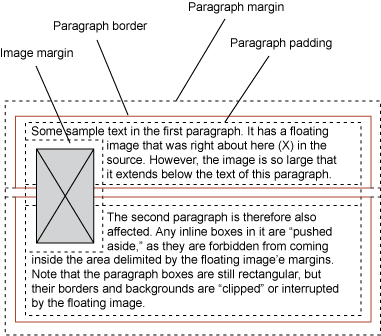
**반대편 수평 마진 조건**: 왼쪽에 왼쪽 부유 박스가 있는 왼쪽 부유 박스는 오른쪽 마진 모서리가 용기 블록의 오른쪽 패딩 모서리보다 오른쪽에 위치할 수 없다. 즉, 왼쪽 부유 박스는 이미 최대한 왼쪽으로 배치되지 않는 한 용기 블록의 오른쪽 모서리를 넘어서 “튀어나오지” 못한다.

**최적 수직 조건**: 부유 박스는 용기 블록 내에서 최대한 위쪽에 놓여야 한다.

**최적 수평 조건**: 왼쪽 부유 박스는 용기 블록 내에서 최대한 왼쪽에 놓여야 한다.

**위치 우선순위 규칙**: 부유 박스 위치를 결정할 때는 최적 수직 조건이 최적 수평 조건에 우선한다. 즉, 부유 박스를 최대한 위쪽으로 두는 위치를 최대한 옆쪽으로 두는 위치보다 우선한다.

그림 1은 위 규칙에 따라 이미지를 배치한 예다. 텍스트 블록은 이미지와 상대적인 마진을 유지한다.

**그림 1. 왼쪽 부유 박스가 놓이는 예**  


추가로 고려할 사항

초보 웹 디자이너들은 흔히 문서 내에서 부유 요소가 놓이는 위치를 정확히 이해하지 못해서 혼란을 겪는다. 이러한 혼란을 조금이나마 덜어줄 경험적 법칙을 소개한다.

**위치**: 브라우저는 문서의 “정상 흐름”에 속한 요소 위치를 기준으로 부유 요소의 수직 위치를 결정한다. 정상 흐름이란 부라우저가 위치 지정 속성을 무시하는 경우 문서가 표시되는 방식을 가리킨다. 부유 요소는 정상 흐름에서 벗어나 용기 요소 내 최대한 왼쪽이나 오른쪽에 배치된다.

**인라인 블록**: 부유 요소는 화면에서 블록 박스가 된다. 즉, display: block 속성을 지정하는 경우와 동일하다.

**폭 지정**: 부유 요소는 폭을 지정해야 한다. CSS2 명세는 1) "width" 속성에 폭을 명시적으로 지정하든지 2) 이미지와 같은 하위 요소로부터 암시적으로 폭을 계산한다. 이미지는 이미지 속성 자체가 폭을 포함한다. 부유 요소에 폭을 지정하지 않으면 결과를 예측할 수 없다.

[위로](http://www.ibm.com/developerworks/kr/library/wa-css/index.html#ibm-pcon)

"clear" 속성이란?

부유 요소 다음에 나오는 요소는 부유 요소를 둘러싸며 흐른다. 다음 요소가 텍스트라면 바람직한 결과가 나온다. 그러나 페이지 레이아웃을 잡으려고 float 속성을 사용한다면 둘러싸기 동작을 취소해야 한다. 이 때 둘러싸기 동작을 취소하려는 요소에 "clear" 속성을 지정한다.

CSS2 명세는 clear 속성을 명확히 정의하고 동작 방식도 규정한다. 이 절에서는 CSS2 섹션 9.5 화면구도 모델(Visual Fomatting Model)에 기술된 내용을 토대로 clear 속성을 살펴본다([참고자료](http://www.ibm.com/developerworks/kr/library/wa-css/index.html#resources) 참조).

정의

clear 속성은 이전 부유 박스와 인접하지 말아야 할 요소 박스 측면을 지정한다. clear 속성을 지정하려는 요소가 부유 요소를 포함하는 경우 clear 속성은 하위 요소에 영향을 미치지 않는다.

clear 속성은 블록 수준 요소에만 적용된다. 여기에는 float 속성으로 블록 수준 요소가 된 요소도 포함된다.

속성은 다음 값 중 하나가 된다.

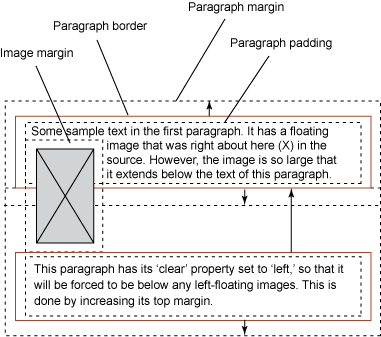
* **inherit** - 요소를 감싸는 바깥 요소에서 clear 속성을 상속 받는다. 기본 값이다.
* **left** - 생성된 박스의 위쪽 마진을 증가시켜 위쪽 테두리 모서리가 이전 왼쪽 부유 박스의 아래쪽 마진 모서리보다 아래쪽에 위치하게 만든다.
* **right** - 생성된 박스의 위쪽 마진을 증가시켜 위쪽 테두리 모서리가 이전 오른쪽 부유 박스의 아래쪽 마진 모서리보다 아래쪽에 위치하게 만든다.
* **both** - 생성된 박스 전부를 이전 부유 요소 아래로 배치한다.
* **none** - 이전 부유 박스 위치와 상관 없이 박스를 배치한다.

동작 방식

clear 속성은 본질적으로 부유 요소가 따르는 위치 규칙을 수정한다. 명세서에서는 이를 부유 요소에 적용되는 추가 제약 조건으로 취급한다.

**clear 조건**: 부유 박스의 위쪽 마진 모서리는 ("clear:left" 경우) 모든 이전 왼쪽 부유 박스의 아래쪽 마진 모서리보다 아래쪽에 위치하거나 ("clear:right" 경우) 모든 이전 오른쪽 부유 박스의 아래쪽 마진 모서리보다 아래쪽에 위치하거나 ("clear:both" 경우) 모든 이전 부유 박스의 아래쪽 마진 모서리보다 아래쪽에 위치해야 한다.

그림 2는 clear 속성을 사용하는 예다. 두 문단에는 모두 "clear:left" 속성이 지정되어 있다. 첫 번째 문단은 clear 속성이 아무런 영향을 미치지 않으므로 부유 박스가 문단 내에 놓인다. 두 번째 문단은 clear 속성 때문에 부유 이미지보다 아래쪽에 놓인다. 두 번째 문단의 위쪽 마진 모서리가 위로 연장되어 첫 번째 문단의 아래쪽 마진 모서리와 접촉한다는 사실에 주목한다.

**그림 2. clear 속성을 사용하는 예**  


추가로 고려할 사항

clear 속성을 사용할 때 한 가지 어려운 점이라면, 부유 요소 다음에 나오는 요소에 마크업을 추가해야 한다는 점이다.

Listing 1에서 보듯이, clear를 처리하려면 HTML 문서에 내용이 없는 빈 마크업을 추가한다.

**Listing 1. "clear"를 처리하기 위해 추가한 빈 마크업**

|  |
| --- |
| <div class="myFloatClass">  <p>myFloatClass has float:left specified</p>  </div>  <div class="myClearClass" />  <!--myClearClass has clear:left specified--> |

여러 사람이 HTML 문서에 별도로 마크업을 추가하지 않고 clear 속성을 처리하는 방법을 제안했다. 그 중 몇 가지를 살펴보자.

* 컨테이너 요소에 "float"를 추가한다
* 컨테이너 요소에 "overflow:hidden"을 추가한다.
* 컨테이너 요소에 CSS2 ":after" 의사-클래스를 추가한다.

**부유 컨테이너 사용**: 페이지 레이아웃과 수평 탐색 메뉴에서 흔히 사용하는 기교로, 부유 컨테이너 내에 부유 요소를 넣는다. 그러면 부유 컨테이너는 내부 부유 요소를 모두 포함하도록 확장된다. 이 방법을 사용하려면 컨테이너 폭 속성을 올바로 설정해야 한다. 흔히 사용하는 값은 100%다.

이 방법은 여러 가지 단점이 있다. 첫째, 폭을 설정하기 어렵다. 100%를 지정했다가는 패딩과 충돌할 우려가 생긴다. 둘째, IE V6는 아래쪽에 마진을 추가할지도 모른다. 마지막으로, 부유 박스를 여러 겹 중첩하는 경우 일부 브라우저가 페이지를 어떻게 표시할지 예측하기 어렵다.

Listing 2는 부유 컨테이너를 정의한 CSS 코드다.

**Listing 2. 부유 컨테이너 사용**

|  |
| --- |
| .myFloatClass {  float: left;  width: 100%;  } |

**"overflow:hidden" 컨테이너 사용**: 거의 사용하지 않는 방법이다. "overflow" 속성과 기본 "clip" 속성이 동작하는 방식을 이용한다. overflow 요소는 (부유 요소를 포함하여) 내부 요소 전체의 크기에 맞춰 확장된다.

이 기술에는 몇 가지 단점이 있다. 첫째, overflow 속성은 컨테이너 높이에 영향을 미친다. 둘째, overflow 속성을 사용하면 IE V6에서 "hasLayout" 속성도 적용된다.

Listing 3은 "overflow:hidden" 컨테이너를 사용하는 CSS 코드다.

**Listing 3. "overflow:hidden" 컨테이너 사용**

|  |
| --- |
| .myFloatClass {  overflow: hidden;  height: 1%;  /\* Or zoom:1 to trigger IE's hasLayout \*/  } |

**:after 의사-클래스 기술**: 이 방법은 거의 쓰이지 않는다. CSS 의사-클래스를 사용하여 컨테이너 다음에 내용을 생성하는 방법이다. :after 의사-클래스를 사용하여 결과 HTML 문서 다음에 clear 속성이 지정된 새 요소를 추가한다.

이 기술은 세 가지 방법 중 가장 바람직하지 못하며 지원하는 브라우저도 별로 없다. :after 의사-클래스는 IE V7 이전 버전에서 지원되지 않는다. 하지만 가장 중요하게는 결과 HTML에 의미 없는 내용을 추가한다는 점에서 바람직하지 못하다.

Listing 4는 :after 의사-클래스 기법을 사용하는 CSS 코드다.

**Listing 4. :after 의사-클래스 기법**

|  |
| --- |
| .myFloatClass {  height: 1%;  /\* Or zoom:1 to trigger IE's hasLayout \*/  }  .myFloatClass:after {  content: ".";  display: block;  height: 0;  clear: both;  visibility: hidden;  } |

**"float"의 용도**

요즘 웹 디자이너들은 float 속성을 즐겨 사용한다. 쓰임새는 디자이너의 상상력에 따라 무궁무진하다. 여기서는 많이 쓰이는 용도 몇 가지를 소개한다.

float를 사용하여 아래 효과를 일으키는 구체적인 방법은 [참고자료](http://www.ibm.com/developerworks/kr/library/wa-css/index.html#resources)에서 소개하는 책과 웹 사이트를 참조한다.

**이미지 고정**: 단순히 이미지를 왼쪽이나 오른쪽으로 옮기는 정도는 float 용도의 절반에 불과하다. float 속성은 원본 문서에서 요소가 포함된 시점부터 유효하므로 부유 요소는 그 지점에 고정된다. 이미지를 설명하는 텍스트를 이미지 근처에 적절히 배치하려는 경우에 특히 중요한 특성이다.

**제목 추가**: 이미지에 "그림 1. 예제 이미지"라는 제목을 달려는 경우가 생긴다. 이미지와 제목을 <div> 블록 안에 넣고 div 블록을 부유시키면 쉽게 처리할 수 있다.

**커다란 대문자**: 텍스트 블록의 왼쪽 상단에 커다란 대문자가 표시되는 전형적인 책 스타일을 만들려고 한다. 단순히 글꼴 크기를 변경하면 문자가 다른 텍스트 위로 늘어나므로 바람직하지 못하다. 대신, float 속성을 사용하면 문자가 아래로 늘어나 나머지 텍스트가 문자를 둘러싸게 만든다.

**인라인 리스트**: float와 "display:inline" 속성을 조합해서 순서없는 <ul> 리스트를 수평 메뉴나 탐색 탭으로 변환한다. 메뉴를 <ul> 리스트로 표현하면 (텍스트 전용 브라우저처럼) 고급 레이아웃이나 그래픽 레이아웃을 지원하지 않는 웹 브라우저에서도 메뉴를 표현할 수 있다.

**다단 페이지 레이아웃**: 웹 페이지에서 테이블을 사용해서 열을 생성하면 원본 문서에 내용과 무관한 마크업이 늘어나서 페이지를 관리하기가 어려워진다. 웹 표준을 준수하는 브라우저가 늘어나면서 div와 CSS를 사용하여 마크업을 줄이고 같은 효과를 얻어내기가 쉬워졌다. float 속성을 사용하면 div 블록을 테이블 행/열과 비슷하게 배치하기 쉽다.

**유연한 갤러리 페이지**: 전통적으로 이미지 갤러리는 테이블을 사용하여 이미지를 배치했다. 하지만 테이블은 유동적이지 못해서 사용자가 윈도우 크기를 변경하면 페이지 레이아웃이 손상되기 쉽다. 대신 부유 이미지를 사용하면 갤러리가 테이블 레이아웃처럼 보이지만 사용자가 윈도우 크기를 조정하면 레이아웃이 좁아지거나 넓어진다. 예를 들어, 갤러리 페이지에서 이미지 12개를 1x12, 2x6, 3x4, 4x3, 6x2, 12x1 등 다양한 형식으로 표시할 수 있다.

개발 도구 고려 사항

개발자라면 누구나 알겠지만, HTML/CSS WYSIWYG 편집기가 모두 동급은 아니다. 편집기마다 나름대로 기벽이 있고, 웹 표준을 준수하는 정도도 다르고, 사용하는 페이지 렌더링 엔진도 다르다.

나는 IBM® Rational® Software Architect V7, 어도비 드림위버 CS3, 지뉴이텍 마이이클립스 V6 등 인기 있는 WYSIWYG 편집기를 두루 써보았는데, 모두 처음 시작하기에는 유용하지만 어느 정도 수준에 이르면 결국 조율을 위해 원시 코드를 손봐야 했다. 페이지를 제작한 후에는 대상 브라우저에서 반드시 테스트해야 하는데, 많은 편집기가 편집기 메뉴에서 브라우저를 바로 띄워서 페이지를 테스트하는 추가 기능을 제공했다.

WYSIWYG 편집기를 평가하려면 Acid 2 브라우저 테스트([참고자료](http://www.ibm.com/developerworks/kr/library/wa-css/index.html#resources) 참조)에 편집기를 적용해보길 권한다. Acid 2 브라우저 테스트는 브라우저 개발업체가 브라우저 표준 준수 수준을 테스트하기 쉽도록 웹 표준 프로젝트에서 개발한 테스트다. 그러나 현재 선택을 고려 중인 HTML/CSS WYSIWYG 편집기를 Acid 2를 사용해서 테스트해도 좋다.

브라우저 고려 사항

널리 사용하는 브라우저 제품이라도 표준에 어긋나는 동작이 많다.

우선, 브라우저마다 중첩된 부유 요소를 처리하는 방식이 다르다. 그러므로 가능하면 부유 요소 내에 부유 요소를 넣는 방식은 피하는 편이 좋다. 중첩이 필요하다면 상대적인 위치를 지정하거나 절대적인 위치를 지정하는 방법을 권한다.

**인터넷 익스플로러**: [Position Is Everything](http://www.positioniseverything.net/) 웹 사이트는 IE 버그와 해결법을 소개한다. 자세한 내용은 [Explorer Exposed 페이지](http://www.positioniseverything.net/explorer.html)를 살펴본다.

**파이어폭스**: 파이어폭스에서 float 속성과 관련하여 내가 찾은 버그는 Top Gap 버그다. 열이 여러 개고 각 열은 부유 요소이지만 헤더가 부유 요소가 아닌 경우에만 나타나는 버그다. 위쪽 마진이 있는 열 요소가 헤더 위로 올라가는 바람에 윈도우 윗단과 헤더 섹션 윗단에 틈이 생긴다. 버그를 수정하려면 부유 열 위에 무언가를 넣으면 된다. 헤더 바닥에 "clear:both" 속성을 가진 요소를 추가하는 방법도 있고, header까지 모두 부유 요소로 만드는 방법도 있다.

**오페라**: [Position Is Everything](http://www.positioniseverything.net/op-omnibus.html) 웹 사이트는 오페라 버그와 해결책도 소개한다. 자세한 내용은 [Opera Omnibus 페이지](http://www.positioniseverything.net/op-omnibus.html)를 참조한다.

요약

페이지에서 요소를 부유시키는 방법은 초창기 웹 브라우저에서부터 존재했던 기술이다. 요즘 웹 디자인은 과거 테이블을 사용하던 방식을 점차 float 속성으로 대체하는 추세다. float 속성은 정의와 동작 방식이 간단하지만, 브라우저마다 구현 방식이 달라서 속성을 사용하기 쉽지 않다. 이 기사에서 살펴본 내용은 겉핥기에 불과하다. 좀더 자세한 내용은 [참고자료](http://www.ibm.com/developerworks/kr/library/wa-css/index.html#resources)를 살펴본다. 물론, 속성을 직접 사용하면서 실험해 보는 방법도 아주 좋다.

참고자료

**교육**

* 이 기사에서 소개한 그림 1과 그림 2는[Cascading Style Sheets, Level 2, CSS2 Specification](http://www.w3.org/TR/CSS2/) 문서에서 발췌했다. 그림을 사용해도 좋다는 허가는 [W3C Copyright statement](http://www.w3.org/TR/CSS2/about.html#copyright)를 참조한다.
* Eric Meyer가 쓴 온라인 기사인 "[Containing Floats](http://www.complexspiral.com/publications/containing-floats/)"는 기본 페이지 레이아웃에서 float가 동작하는 방식을 설명한다.
* [CSS: The Definitive Guide(3rd ed)](http://www.amazon.com/CSS-Definitive-Guide-Eric-Meyer/dp/0596527330)는 Eric Meyer가 집필하고 오라일리 출판사가 펴낸 CSS 참조 서적이다.
* [CSS Cookbook(2nd ed)](http://www.amazon.com/CSS-Cookbook-2nd-Christopher-Schmitt/dp/0596527411/ref=pd_bbs_sr_1?ie=UTF8&s=books&qid=1204138427&sr=1-1)은 Christopher Schmitt가 집필하고 오라일리 출판사가 펴낸 책이다. CSS 예제가 풍부하다.
* [Floatutorial](http://css.maxdesign.com.au/floatutorial/)에서는 여러 편에 걸친 튜토리얼을 통해 이미지와 페이지 레이아웃에서 float 속성을 사용하는 방법을 소개한다.
* Robert Nyman이 자신의 블로그에 올린 글인 [How to Clear CSS Floats Without Extra Markup - Different Techniques Explained](http://www.robertnyman.com/2007/04/12/how-to-clear-css-floats-without-extra-markup-different-techniques-explained/)는 웹 페이지에 태그를 별도로 추가하지 않고 float 속성을 해제(clear)하는 방법 세 가지를 소개한다.
* [Cascading Style Sheets, Level 2, CSS2 Specification](http://www.w3.org/TR/REC-CSS2/)은 W3C에서 내놓은 명세서다. 현재 CSS 속성은 대개 이 명세서가 기술하는 동작을 따른다. [9.5절](http://www.w3.org/TR/CSS2/visuren.html#floats)에서 float 속성을 기술한다.
* [Web Standards Project(WaSP)](http://webstandards.org/)는 웹 표준을 널리 퍼트리려고 노력하는 프로젝트다.
* [Acid2 테스트](http://www.webstandards.org/action/acid2/)는 웹 브라우저가 웹 표준을 준수하는지 검사한다.
* [Position Is Everything](http://www.positioniseverything.net/)은 브라우저 호환성, 버그, 해킹 소식을 제공하는 사이트다.
* Michael Russell이 쓴 기사인 "[Compare Web site appearance and functionality](http://www.ibm.com/developerworks/kr/library/wa-qualbust14.html)"는 브라우저 확장 대신 웹 표준을 사용하는 장점을 논한다.
* "[Cross-browser Web application testing made easy](http://www.ibm.com/developerworks/kr/library/wa-crossbrowser/)"는 다양한 브라우저에서 웹 응용 프로그램을 쉽게 테스트하는 방법을 소개한다.
* [Hoverbox Image Gallery](http://sonspring.com/journal/hoverbox-image-gallery)는 float 속성을 멋지게 활용하는 사이트다. [developerWorks 기사](http://www.ibm.com/developerworks/search/searchResults.jsp?searchType=1&searchSite=dW&searchScope=dW&query=hoverbox)에서도 여러 차례 언급했다.
* [developerWorks 웹 개발 영역](http://www.ibm.com/developerworks/web): 여러분의 실력을 향상시켜 줄 다양한 자원을 제공한다.

**제품 및 기술 얻기**

* [IBM 제품 평가판](http://www.ibm.com/developerworks/kr/downloads/): DB2®, Lotus®, Rational®, Tivoli®, WebSphere®와 같은 제품을 developerWorks에서 직접 내려 받아 다음 프로젝트에 활용해 보자.

**토론**

* [developerWorks 블로그](http://www.ibm.com/developerworks/blogs/)를 읽어보고 [developerWorks 공동체](http://www.ibm.com/developerworks/community)에 참여한다.

필자소개



Michael Russell은 물리학에서 학사 학위를, 전자계산학에서 석사 학위를 소지하고 있다. 그는 물류 엔지니어이자, 기술 서비스 관리자이자, 지난 14년 동안 IBM이 인증한 IT 아키텍트다. 현재는 군수 항공 물류와 시뮬레이션 업계에서 시스템 아키텍트로 일한다. 윈도우, 유닉스, OS/400 환경에 익숙하며, 회사 웹 사이트를 통해 재미삼아 웹 기술을 실험한다.