**■ 박스의 배치 방식**  
    • 일반적인 흐름(normal flow)으로 배치되는 박스  
    • 떠있는(float) 상태로 배치되는 박스  
    • 절대적인 위치(absolute position)로 배치되는 박스  
  
**■ Float 속성**  
    float:left, right, inherit, none  
    부모요소를 기준으로 왼쪽, 오른쪽에 띄워 위치 시킨다.  
    이런 상태는 위의 박스배치 방식중 float상태로 배치되는 것이다.

<div class="block pink float"></div>

<div class="block blue float"></div>

<div class="block green"></div>

<div class="block orange"></div>

.block {

width: 200px;

height: 200px;

}

.float { float: left; }

.pink { background: #ee3e64; }

.blue { background: #44accf; }

.green { background: #b7d84b; }

.orange { background: #E2A741; }

**※ 이해**  
       - pink,blue는 왼쪽으로 float되어 표시됩니다.  
        - pink,blue는 float로 배치되고,

          green,orange는 일반적흐름(normal flow)로 배치되므로  
          green는 pink아래쪽에 표시되게 됩니다.

    아래 show green 버튼은 green을 보였다 안보였다하게합니다.  
     green이 일반적인흐름(normal flow)으로 배치되어

    부모요소 body에 기본배치됨으로 pink아래 위치하고  
     orange는 green의 아래쪽에 위치하게된것입니다.

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html lang="en">

<head>

<meta http-equiv="content-type" content="text/html; charset=utf-8">

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.1/jquery.min.js"></script>

<title>CSS Floats 101</title>

<style>

.block {

width: 200px;

height: 200px;

}

.float { float: left; }

.float\_r { float: right; }

.pink { background: #ee3e64; }

.blue { background: #44accf; }

.green { background: #b7d84b; }

.orange { background: #E2A741; }

.gray { background: #c0c0c0; }

</style>

<script>

function togglePink() {

var v = $(".pink").get(0).style.visibility;

if ( v=="hidden" ) {

v = "visible";

$("#btnToggleGreen").get(0).innerHTML = "show green";

} else {

v = "hidden";

$("#btnToggleGreen").get(0).innerHTML = "hide green";

}

$(".pink").get(0).style.visibility = v; //visible

}

</script>

</head>

<body>

<div class="block pink float"></div>

<div class="block blue float"></div>

<div class="block gray float\_r"></div>

<div class="block green"></div>

<div class="block orange"></div>

<button onclick="togglePink()" id="btnToggleGreen">show green</button>

</body>

</html>

**■ Clear 속성**  
    **속성값 : left, right, both, inherit, none**    float속성이 지정된 요소는 일반적인흐름(normal flow)을 벗어나 플로트되어 표시된요소입니다.  
    따라서, 플로트된요소 이후의 엘리먼트이면서 float속성이 지정되지 않은 요소들은

       일반적인흐름(normal flow)로 배치되게 됩니다.

    그래서 위에서 처럼 green이 pink뒤로 숨는 증상이 나타나게 된것입니다.

     - clear는 left     적용한 엘리먼트는 상위에 위치한 float:left를 가지는 엘리먼트보다 아래에 위치하게됩니다.  
     - clear는 right   적용한 엘리먼트는 상위에 위치한 float:right를 가지는 엘리먼트보다 아래에 위치하게됩니다.

     - clear는 both   적용한 엘리먼트는 상위에 위치한 float:left,float:right를 가지는 엘리먼트보다 아래에 위치하게됩니다.

     - clear는 inherit 적용한 엘리먼트는 부모요소로부터 clear속성을 물려받습니다.

     - clear는 none   적용한 엘리먼트는 아무것도 지정하지 않은것과 같습니다.

**※ clear로 지정한 속성을 가진 상위의 float 엘리먼트보다 아래 위치한다.**

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html lang="en">

<head>

<meta http-equiv="content-type" content="text/html; charset=utf-8">

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.1/jquery.min.js"></script>

<title>CSS Floats 101</title>

<style>

.block {

width: 100px;

height: 100px;

border:1px solid black;

}

.float { float: left; }

.pink { background: #ee3e64; }

.blue { background: #44accf; }

.green { background: #b7d84b; }

.orange { background: #E2A741; }

.clear { clear:both }

</style>

<script>

//var colors = ["#ee3e64","#44accf","#b7d84b","#E2A741","#FF0000","#00FF00","#0000FF","#F0F000","#00F0F0","#F0000F","#0F000F","#0F0F00","#0000FF"];

var colors = ["#ff0000","#ff8c00","#ffff00","#008000","#0000ff","#4b0082","#800080"];

var winTimer = null;

var pos = 0;

function toggleColor(obj) {

var v = obj.get(0).style.visibility;

if ( v=="hidden" ) {

v = "visible";

//$("#btnToggleGreen").get(0).innerHTML = "show green";

} else {

v = "hidden";

//$("#btnToggleGreen").get(0).innerHTML = "hide green";

}

//obj.css("visibility",v); //visible

if (pos >= colors.length -1 ) {

pos = 0;

}

pos++;

obj.html(pos);

obj.css("background-color",colors[pos]); //visible

}

window.onload = function() {

/\*

winTimer = window.setInterval(function() {

toggleColor($(".pink"));

},500);

$("#btnToggleGreen").get(0).innerHTML = "Stop toggle color";

\*/

$("#btnToggleGreen").get(0).innerHTML = "Change Next Color~!";

/\*

window.setInterval(function() {

var pos =0;

toggleColor($(".blue"),pos);

},200);

window.setInterval(function() {

toggleColor($(".green"));

},200);

window.setInterval(function() {

toggleColor($(".orange"));

},200);

\*/

}

function setCssClear(obj) {

obj.get(0).style.backgroundColor ="#000";

obj.get(0).style.border ="10px solid #fff";

obj.get(0).style.color = "#fff";

obj.get(0).style.clear = "none";

}

var setTimer = function (start) {

start = typeof start === "boolean"?start:!!!start;

if ( start ) {

window.clearInterval(winTimer);

winTimer = null;

winTimer = window.setInterval(function() {

toggleColor($(".pink"));

},500);

console.info("Start");

} else {

window.clearInterval(winTimer);

winTimer = null;

console.info("Stop");

}

}

function toggleTimer() {

if ( winTimer ) {

$("#btnToggleTimger").get(0).innerHTML = "Start";

setTimer(false);

} else {

$("#btnToggleTimger").get(0).innerHTML = "Stop";

setTimer(true);

}

}

</script>

</head>

<body>

<div class="block pink float">row 1</div>

<div class="block blue float">row 1</div>

<div class="block green float">row 1</div>

<div class="block orange float">row 1</div>

<div class="block pink float clear" id="div\_clear">row 2</div>

<div class="block blue float">&lt;-- clear:left row 2</div>

<div class="block green float">row 2</div>

<div class="block orange float">row 2</div>

<button onclick=&apos;toggleTimer();&apos; id="btnToggleTimger">Start</button>

<button onclick=&apos;toggleColor($(".pink"));&apos; id="btnToggleGreen">Stop toggle color</button>

<button onclick=&apos;setCssClear($("#div\_clear"));&apos;>Clear:none</button>

</body>

</html>

**■ img { float: right; margin: 10px; }**  
  
**■ visibility 속성**    속성값 : hidden / visible  
    display     : 존재여부에대한 조작  
    visibility  : 가시성에대한 조작  
  
**■ display 속성**  
    block : 블록박스로 만든다.  
    inline : 인라인 박스로 만든다.  
    inline-block : 블록박스로 만들어지지만, 인라인 박스처럼 배치된다. (CSS2.1 권고후보에 추가)  
    none : 박스를 만들지 않으며 시각적으로 완전히 보이지 않는다.  
    inherit : 상위 요소의 display속성을 상속받는다.  
  
    table : 블록레벨의 표로 만든다. (table)  
    inline-table : 인라인레벨의 표로 만든다. (table)  
    table-row : 행으로 만든다 (tr 기본 속성)  
    table-row-group : 본체 행 그룹으로 만든다. (tbody 기본 속성)  
    table-header-group : 헤더 행 그룹으로 만든다. (thead 기본 속성)  
    table-footer-group : 푸터 행 그룹으로 만든다. (tfoot 기본 속성)  
    table-column : 열로 만든다. (col 기본 속성 - 보이지 않음)  
    table-column-group : 열 그룹으로 만든다. (colgroup 기본 속성 - 보이지 않음)  
    table-cell : 하나의 셀로 만든다. (td 기본 속성)  
    table-caption : 캡션으로 만든다. (caption 기본 속성)  
  
    list-item : li요소와 같은 블록박스와 목록항목의 인라인박스로 만든다.  
    run-in : 런인박스로 만든다.  
    compact : 콤팩트 박스로 만든다. (CSS2.1권고후보에서 삭제)  
    marker : 그 요소의 전후에 생성된 내용을 마커로 정의한다. (CSS2.1권고후보에서 삭제)

**※ 참고**

<http://gskool.tistory.com/63>

<http://tranbot.net/ALA/css-floats-101/>

<http://blog.wystan.net/2009/01/12/relationships-between-position-float-display>

<http://www.google.co.kr/#q=css+float+css2&newwindow=1&lr=lang_ko&sa=X&ei=D8e3Ud7_JeqviQePn4CABw&ved=0CC4QuAE&bav=on.2,or.r_qf.&bvm=bv.47810305,d.aGc&fp=9383809cfbd919f5&biw=1680&bih=855>

<http://www.google.co.kr/#lr=lang_ko&newwindow=1&tbs=lr:lang_1ko&sclient=psy-ab&q=css+float&oq=css+float&gs_l=hp.3..0l4.180899.182607.1.182996.2.2.0.0.0.0.139.224.1j1.2.0...0.0...1c.1.17.psy-ab.ygKt6O_SNLI&pbx=1&bav=on.2,or.r_qf.&bvm=bv.47810305,d.aGc&fp=9383809cfbd919f5&biw=1680&bih=855>

<http://techbug.tistory.com/181>

<http://iyakiggun.blog.me/100162787755>

<http://blog.naver.com/tionnii?Redirect=Log&logNo=120181294436>

[C:\Users\ADMINI~1\AppData\Local\Temp\enhtmlclip\EN00ec632a.png](file:///C:\Users\Administrator\AppData\Local\Temp\enhtmlclip\css_float이해.docx)