

position의 사용법

- 기준이 되는 부모요소에 position: 속성을 선언
- 부모의 위치를 기준으로 움직이고자 하는 자식요소에 position: absolute 를 선언하고 원하는 위치로 움직이기 위해 top, left, right, bottom, z-index 값 할당

position 의 종류

- **position:static**

포지션 속성의 기본값

- **position:relative (normal flow를 해치지 않아요)**

상대좌표 : 자신의 현재 위치를 기준으로 하여 자신의 위치를 결정합니다.

- **position:absolute**

절대좌표 : position 속성이 적용된 가장 가까운 부모를 기준으로 하여 자신의 위치를 결정합니다.

- **position:fixed**

브라우저의 뷰포트를 기준으로 하여 자신의 위치를 결정합니다.

position 속성의 단점

1. 자동화가 불가능하여 코드의 낭비가 심하다.
2. HTML을 랜더링할 때 비용이 발생한다.

- 화면이 랜더링 되는 순서

DOM -> CSSOM -> Render tree -> 레이아웃 계산

- 그리고 이 레이아웃이 계산 될때는

BOX-MODEL —> NORMAL-FLOW —> POSITIONING

transition & transform

Transition

- 원하는 속성을 시간에 따라 변화시켜줍니다.

transition:속성 시간 움직임 딜레이

div{transition:background 0.5s linear 0.3s}

동시에 여러가지 속성을 바꾸는것도 가능합니다!

<http://cubic-bezier.com/>

Transform

transform:속성(속성 값)

각도) div{transform:rotate(360deg)}

크기) div{transform:scale(2,4)}

위치) div{transform:translate(10px, 50px)}

Page Reflow & Repaint

1. Reflow

생성된 DOM 노드의 레이아웃(너비, 높이 등) 변경 시 영향받는 모든 노드(자식, 부모)의 수치를 다시 계산하여 렌더 트리를 재생성하는 작업입니다.

width	height
padding	margin
display	border-width
border	top
position	font-size
float	text-align
overflow-y	font-weight
overflow	left
font-family	line-height
vertical-align	right
clear	white-space
bottom	min-height

Page Reflow & Repaint

2. Repaint

reflow 과정이 끝난 후 재생성된 렌더 트리를 다시 그리는 작업으로 수치와 상관없는 스타일 변경시에는 reflow 과정을 생략한 repaint 작업만 수행합니다.

color	border-style
visibility	background
text-decoration	background-image
background-position	background-repeat
outline-color	outline
outline-style	border-radius
outline-width	box-shadow
background-size	

Page Reflow & Repaint

이러한 Reflow 현상을 발생시키지 않는 유일한 속성
==> **transform**

GPU를 이용한 그래픽 가속능력 사용!

- <https://www.html5rocks.com/ko/tutorials/speed/high-performance-animations/>