

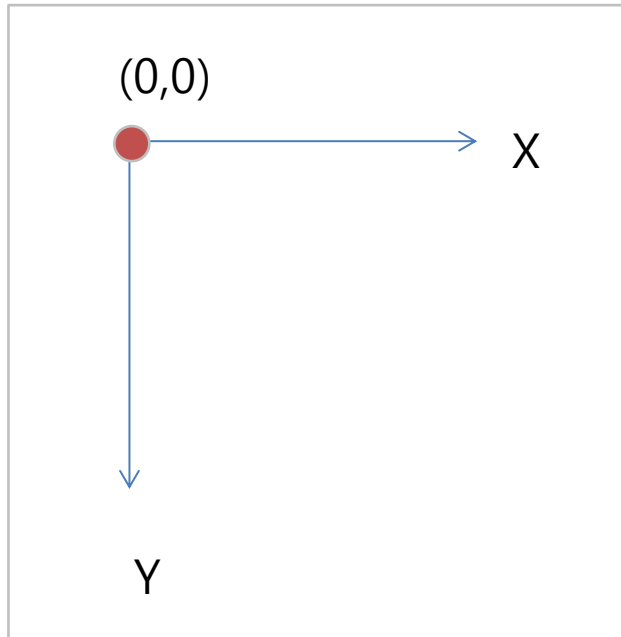
06

CSS와 애니메이션(p449)

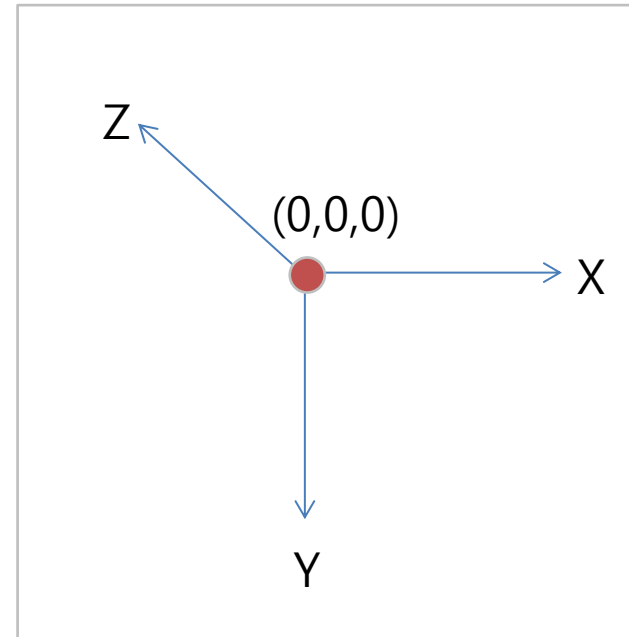
1. Transform
2. Transition
3. animation

❖ 변형(transform)

- 변형 : 특정 요소의 크기나 모양이 변하는 것
- 2차원 변형(2D) : 평면상에서 이동, 회전, 왜곡 시키는 것(x,y좌표)
- 3차원 변형(3D) : X,Y축에 원근감(Z축)을 추가해 변형 하는 것



2차원 좌표계



3차원 좌표계

❖ 변형(transform)

2차원 변형 함수 		3차원 변형 함수 	
translate(tx,ty)	이동 (x,y)	matrix3d(n[,n])	4*4행렬 이용 변환
translateX(tx)	이동(x)	translate3d(tx,ty,tz)	이동(x,y,z)
translateY(ty)	이동(y)	translateZ(tz)	이동(z)
scale(sx,sy)	확대/축소 (x,y)	scale3d(sx,sy,sz)	확대/축소(x,y,z)
scaleX(sx)	확대/축소(x)	scale(sz)	확대/축소(z)
scaleY(sy)	확대/축소(y)	rotate3d(rx,ry,rz,각도)	회전(x,y,z,각도)
rotate(각도)	회전 (각도)	rotateX(각도)	X축 회전(각도)
skew(ax,ay)	왜곡 (x,y)	rotateY(각도)	Y축 회전(각도)
skewX(ax)	왜곡(x)	rotateZ(각도)	Z축 회전(각도)
skewY(ay)	왜곡(y)	perspective(길이)	원근감:입체적 깊이

❖ 변형과 관련된 속성들

속성	속성 값	설명
transform-origin	<x축> <y축> <z축> initial inherit	변형 기준점 설정하기(특정 지점)
perspective	크기 none	원근감 표현하기
perspective-origin	<x축 값> <y축 값>	더 높은 곳에서 원근 조절(50%)
transform-style	flat preserve-3d	하위요소에 3D 변형 적용하기
backface-visibility	visible hidden	요소의 뒷면 표시하기(회전각도>90deg)
코딩 예제	<pre>.ltop{transform-origin:left top;} .pers{perspective:300px;} .tr_sty{transform-style:preserve-3d;} .back{backface-visibility:visible;}</pre>	변형기준점:왼쪽 위 원근감 : 300px 부모요소 속성-하위 요소에도 적용 요소의 뒷면 표시

❖ 상품 이미지 위로 마우스 올릴 때 : 상품 가격 표시하기



Html5&CSS kfn

신상품 목록

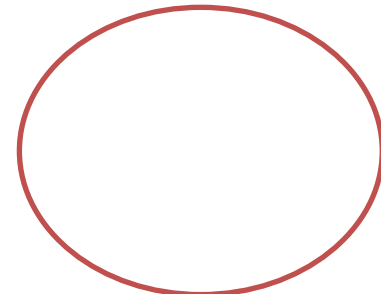
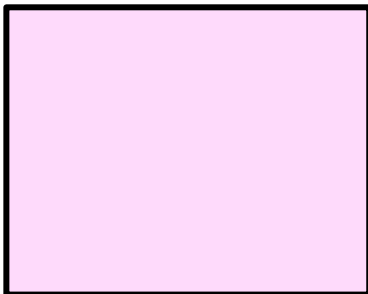
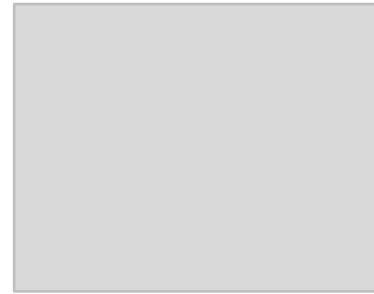


```
<style>
  #container{
    width:80%;
    margin:0 auto;
  }
  ul li{
    list-style:none;
    float:left;
    margin-right:10px;
    position:relative;
    overflow:hidden;
  }
  .caption{
    background-color:rgba(0,0,0,0.6);
```


```
    opacity:0;
    width:300px;
    height:200px;
    position:absolute;
    top:200px;
    z-index:10;
  }
  ul li:hover .caption {
    opacity: 1;
    transform: translateY(-200px);
    color:#fff;
    text-align:center;
  }
</style>
```

❖ 변환 (transition)

- 변환 : 특정 요소의 **스타일 속성이 바뀌는 것**.
- 배경색이나 도형의 테두리 등

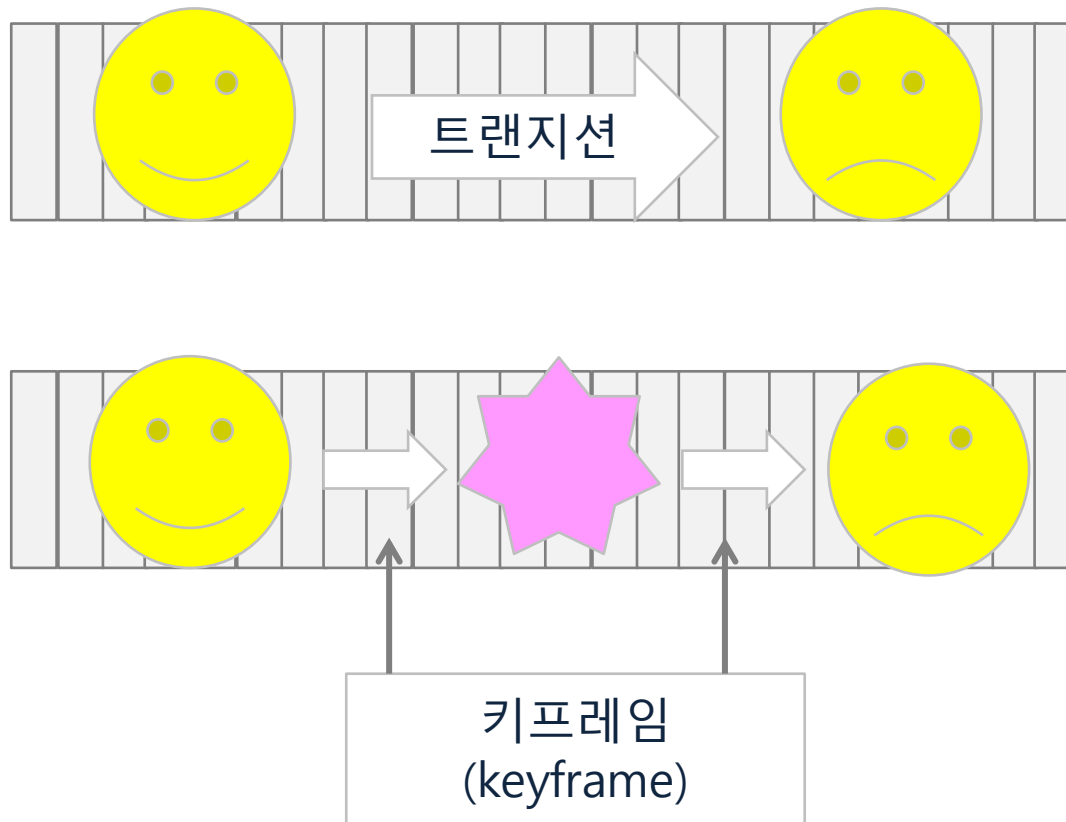


❖ 변환과 관련된 속성들

속성	속성 값	설명
transition-property	all none <속성 이름>	적용할 속성 지정하기
transition-duration	<시간>	진행시간 지정하기
transition-timing-function	linear : 같은 속도 ease : 천천히 시작-빨라지다가-천천히 ease-in : 시작을 느리게 ease-out : 느리게 끝냄 ease-in-out : 느리게 시작-느리게 끝냄 cubic-bezier(n,n,n,n) : 함수 직접 정의 해서 사용	속도 곡선 지정하기 
transition-delay	<시간>	지연 시간 설정하기
transition		트랜지션 속성 한꺼번에 지정
코딩예제	transition-property:background-color; transition-duration:2s; transition-timing-function:linear; transition-delay:1s;	해당요소의 배경색에 변화 적용 2초 동안 변화 진행 처음부터 끝까지 같은 속도로 1초 후에 시작(지연시간)

❖ 애니메이션 (animation)

- 애니메이션 : 시작해서 끝나는 동안 부드럽게 변하는 것
 - 원하는 곳에 스타일 바뀌는 지점 추가 가능(키프레임:keyframe)



❖ 애니메이션 관련 속성

속성	속성 값	설명
@keyframes	<이름>{<선택자>{<스타일>}}	애니메이션 바뀌는 지점 설정
animation-name	<키프레임 이름> none	이름 지정하기
animation-duration	<시간>	실행 시간 설정
animation-direction	normal alternate(왕복)	애니메이션 방향 지정하기
animation-iteration-count	<숫자> infinite	반복 횟수 지정(1), 무한반복
animation-timing-function	linear ease ease-in ease-out ease-in-out cubic-bezier(n,n,n,n)	속도 곡선 지정하기
코딩 예제	<pre>@keyframes cha_bg{ from{...} to{...} animation-name:cha_bg; animation-duration:3s; animation-direction:normal; animation-iteration-count:infinite; animation-timing-function:ease;</pre>	<p>애니메이션 지정 cha_bg{ -에서 ~로 변형</p> <p>이름:cha_bg</p> <p>3초 동안 실행</p> <p>실행 후 원래 위치로</p> <p>무한반복</p> <p>천천히 시작-빨라지다가-다시 천천히 끝냄</p>

```
<style>
#myball {
    position:relative;
    width:100px;
    height:100px;
    border-radius:0px;
    border:2px solid red;
    background:url(img/f1.png);
    animation-name:myani;
    animation-duration:5s;
    animation-iteration-count:2;
    animation-direction:alternate;
    animation-timing-function:linear;
}
```

```
<body>
    <div id="myball"> </div>
</body>
```

```
@keyframes myani {
0%{
    left: 10px;}
40% {
    top:50px;
    border-radius:50px;
    border:2px solid blue;
    background:url(img/f2.png);}
100% {
    left:500px;
}
```

07

반응형 웹 사이트(p488)

1. 반응형 웹
2. 미디어쿼리
3. 플렉스 박스 레이아웃

❖ 반응형 웹 디자인(responsive web design)이란?



- 화면의 크기에 반응해 화면 요소들을 자동으로 바꾸어 사이트를 구현하는 것
- 뷰포트(viewport) : 스마트폰 화면에서 실제 내용이 표시되는 영역
 - 모바일 기기가 아닌 '웹 브라우저 창 너비'에 반응하는 것이다.
- 장점
 - 사이트 하나를 만들어 모든 기기에 사용 가능하므로 효율적이다.
 - 사이트 유지 · 관리가 쉽다.

❖ 뷰 포트 지정하기

기본형 : `<meta name="viewport" content="속성1, 속성2 ...">`

- head 영역에 지정
- content 속성에서 뷰 포트 속성을 지정

속성	설명	값	기본값
width	뷰포트 너비	device-width, 크기	브라우저 기본값
height	뷰포트 높이	device-width, 크기	브라우저 기본값
user-scalable	확대/축소 가능여부	yes/no	yes
initial-scale	초기 확대/축소 값	1-10	1
minimum-scale	최소 확대/축소 값	0-10	0.25
maximum-scale	최대 확대/축소 값	0-10	1.6

```
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
</head>
```

❖ 가변 그리드 레이아웃(fluid grid layout)이란?(p499)



- 화면을 몇 개의 칼럼으로 나누어 요소 배치하는 것
- 화면을 단순하고, 규칙적으로 배열하여 일관성 유지
- 고정 그리드 레이아웃은 화면을 960px, 1200px로 고정하거나, 12 칼럼, 16 칼럼, 24칼럼 그리드 사용
- 가변 그리드 레이아웃은 고정 그리드 레이아웃을 백분율 값으로 계산하여 화면크기에 따라 반응할 수 있도록 조절하는 것
 - 마진 값과 패딩 값도 백분율로 지정(p503)
 - 백분율 값=(요소의 너비/전체너비)*100
 - 예) $600\text{px} / 960\text{px} * 100 = 62.5\%$

❖ 가변 레이아웃과 가변요소

요소	설명		
가변 글꼴	em	부모요소 폰트의대문자 M의 너비=1em(16px) 글자크기(em)=글자크기(px)/16px	
	rem	root의 기본 크기를 기준으로 지정	
가변 이미지	max-width=100%; height: auto; - 부모요소 만큼만 확대/축소		
고해상도 이미지			
	<picture> <source srcset="pen-large.jpg" media="(min-width:1024px)"> <source srcset="pen-medium.jpg" media="(min-width:768px)"> <source srcset="pen-small.jpg" media="(min-width:320px)"> </picture>		
가변 비디오	max-width=100%;		

❖ 미디어 쿼리(p512)

- 어떤 미디어를 사용하느냐에 따라 화면 크기에 맞는 CSS가 적용되도록 하는 것(방법)

기본형 : @media[only|not] 미디어 유형 [and 조건]*[and 조건]

- <style>...</style> 영역에 지정
- 미디어 유형 : all, print, screen, tv, aural(음성 합성 장치),
braille(점자표시장치), handheld, projection, tty,
embossed(점자 프린터)

```
<style>  
  @media all (min-width:600px) and (max-width:959px){  
    적용할 css 속성  
  }  
</style>
```


❖ 미디어 쿼리

구분	속성
웹 문서	width, height, min[max]-width, min[max]-height (p515)
단말기	device-width, device-height, min[max]-device-width, min[max]-device-height (p516)
화면 회전	orientation:portrait[landscape] 세로 방향[가로방향] (p517)
화면 비율	aspect-ratio(width/height), min[max]aspect-ratio (p518)
단말기 화면 비율	device-aspect-ratio(width/height), min[max]-device-aspect-ratio
색상당 비트 수	color(컬러 지원), color:0(컬러지원 안됨), color:3(8비트 컬러지원), min-color:3, max-color:3 (p519)
중단점	서로 다른 css를 적용할 화면 크기(break point) 표준은 없지만 가장 일반적으로 사용할 만한 기기 파악 후 사용

css프레임워크	아주 작은 화면	작은 화면	중간 화면	큰 화면	아주 큰 화면
부트스트랩	576px 미만	576px 이상	768px 이상	992px 이상	1200px 이상
마젠토	640px 미만	640px 이상	768px 이상	1024px 이상	1440px 이상

❖ 미디어 쿼리 적용하기 1

외부 파일 연결하기

- `<link rel="stylesheet" media="미디어파일 조건" href="css 파일경로">`
`<link rel="stylesheet" media="print" href="css/print.css">`
`<link rel="stylesheet" media="screen and (max-width:768)"`
`href="css/tablet.css">`
- **@import url(css파일 경로) 미디어쿼리 조건**
`@import url("css/tablet.css" only screen and (min-width:321px)`
`and (max-width:768px);`
- @import구문보다 `<link>` 태그가 안정적이고 빠르므로 link 태그 사용 권장

❖ 미디어 쿼리 적용하기 2

웹 문서에서 직접 정의

- `<style media="미디어 쿼리 조건">`

스타일 규칙들

`</style>`

`<style media="screen and (max-width:768px)">`

`body{background: orange;}`

`</style>`

- `<style>`

`@media="미디어 쿼리 조건">`

스타일 규칙들

`</style>`

```
<style>
```

```
@media screen and (max-width:320px){
```

```
body{
```

```
background: orange;
```

```
}
```

❖ 미디어 쿼리 사용 문서 만들기(p526)

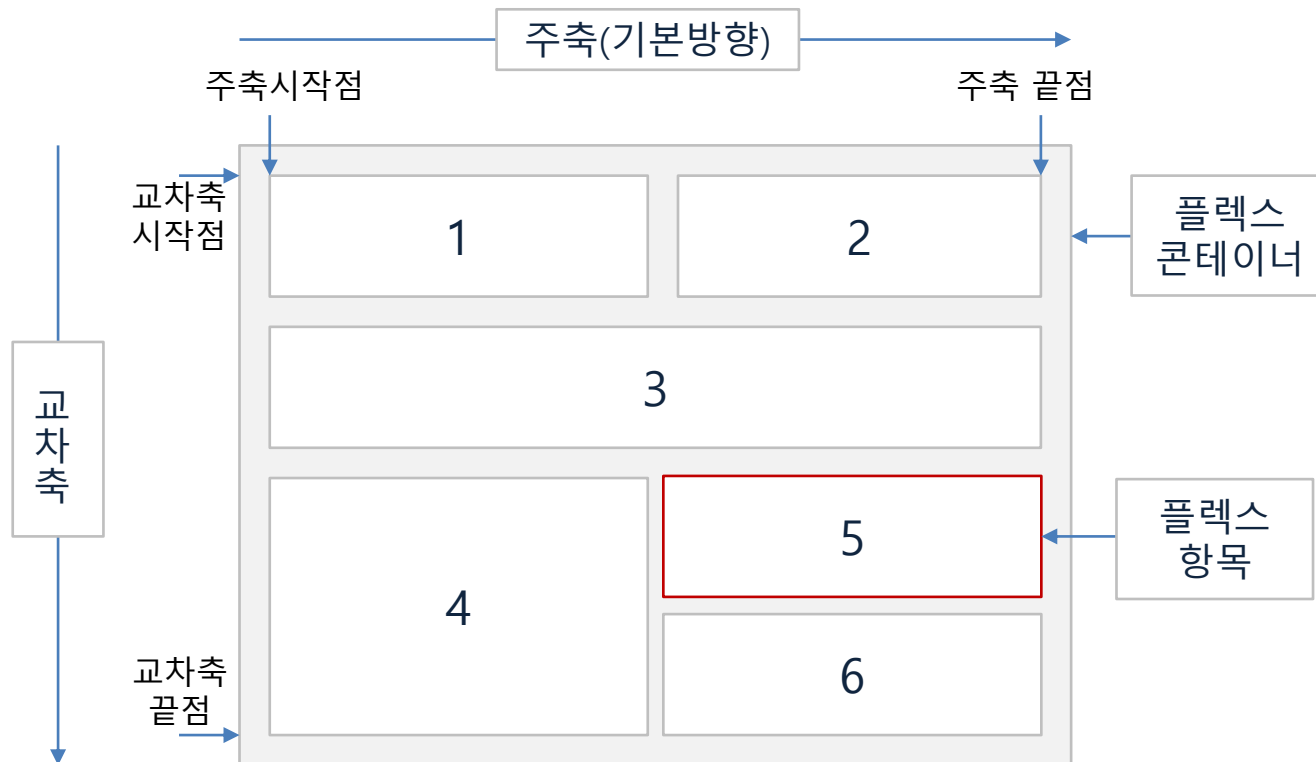
media526.html



- ① 레이아웃 구상하기
 - 모바일 화면
 - 태블릿 화면
 - 데스크톱 화면
- ② 미디어 쿼리 중단점 결정하기
- ③ 태그로 구성
- ④ 모바일 화면 css 작성
- ⑤ 태블릿 화면 css 작성
 - 미디어 중단점(768px 이상 ~ 1719px 이하)
- ⑥ 데스크톱 화면 css 작성
 - 미디어 중단점(1720px 이상)

❖ 플렉스 박스 레이아웃(flex box layout)

- 그리드 레이아웃을 기본으로 플렉스 박스를 원하는 위치에 배치하는 것
- 플렉스 박스를 이용하면 여유 공간에 따라 너비나 높이, 위치를 자유롭게 변형할 수 있어 편리하게 사용할 수 있다



❖ 플렉스 박스 레이아웃

속성	설명	기본형
display	플렉스 컨테이너 지정	display: flex inline-flex
flex-direction	주축 방향 지정	flex-direction: row row-inverse column column-inverse (p539)
flex-wrap	항목 한 줄/여러 줄 배치	flex-wrap: no-wrap wrap wrap-reverse
flex-flow	방향과 배치 함께 지정	flex-flow: <방향> <줄 배치>
order	항목 배치 순서 바꾸기	order: 0 숫자 (0 : 입력 순서, 숫자 순서로 배치) (p542)
flex	플렉스 항목 크기 조절	flex: [<flex-grow> <flex-shrink> <flex-basis>] auto(항목 너비값) initial

flex : 2 2 0; flex: [<flex-grow: 늘릴 비율> <flex-shrink: 줄일 비율> <flex-basis: 기본값>]

❖ 플렉스 박스 레이아웃

속성	설명	기본형
justify-content	주축 기준 배치 방법	justify-content: flex-start flex-end center space-between space-around (p544)
align-items	교차 축 배치 방법	align-items: stretch flex-start flex-end center base-line(글자 기준선) (p545)
align-self	교차 축 배치 방법 (특정 플렉스 항목 배치 가능)	align-self: auto(부모 속성 상속) stretch flex-start flex-end center base-line
align-content	여러 줄일 때 배치 방법 (교차축)	align-content: flex-start flex-end center space-between space-around

❖ 플렉스 박스 이용 사이트 구성하기(p547)

flex547.html



- ① 플렉스 박스 레이아웃 구상하기
 - 모바일 화면
 - 태블릿 화면
 - 데스크톱 화면
- ② 태그로 구성/기본 css작성
- ③ 미디어 쿼리 작성하기
- ④ 플렉스 컨테이너 지정하기
- ⑤ 플렉스 항목 너비 지정하기
- ⑥ 브라우저 확인하기

HTML5 & CSS3

수업 마칩니다

수고했습니다