

[WEB UI Study]

WEB Animation 2탄

NHN NEXT 우재우

지난 시간에 우리는 JS Animation을 공부했어요

`requestAnimationFrame` 기억나니?!

안 나니? 나니?!?!?

오늘은 CSS Animation에 대해 알아보시다



다들 CSS 은근 무시하고 있었죠?

CSS Animation 성능을 무시하지 말라!

나 GPU 쓰는 녀석임

Hardware(GPU) Accelerated CSS

HTML5에 들어오면서 특정 스펙이 대해 일반 연산은 CPU로, graphics processing은 GPU로 처리하게 했습니다. 마치 좌뇌로 수학을 풀고, 우뇌로 그림을 그리는 느낌?!
아래는 GPU가 적용되는 HTML5 스펙입니다.

- General layout compositing
- CSS3 transitions
- CSS3 3D transforms
- Canvas Drawing
- WebGL 3D Drawing

참고: <http://www.html5rocks.com/en/tutorials/speed/html5/>

뭐 이정도만 보여줘도 충분하겠죠?

<http://www.creativebloq.com/css3/animation-with-css3-712437>

찬반이 있지만
꽤나 강력해서 많이 씁니다
CSS Animation!!!

CSS Animation의 키워드!

@keyframes

일단 따라해봅시다

인터넷에 가장 많이 돌아다니는 예제입니다. <h1> 엘리먼트를 오른쪽에서 왼쪽으로 옮기는 애니메이션입니다. <h1> tag를 만들고, CSS 파일에 넣어봅시다.

```
h1 {  
  animation-duration: 3s;  
  animation-name: slidein;  
}  
  
@keyframes slidein {  
  from {  
    margin-left: 100%;  
    width: 300%  
  }  
  
  to {  
    margin-left: 0%;  
    width: 100%;  
  }  
}
```

animation-name은 내 맘대로!

다른 속성들은 나중에 보더라도 이걸 보고 갑시다. animation-name 속성은 맘대로 설정할 수 있어요~ 그리고 name 별로 animation 설정해주면 되겠죠?

```
h1 {  
  animation-duration: 3s;  
  animation-name: smith_babo;  
}  
  
@keyframes smith_babo {  
  from {  
    margin-left: 100%;  
    width: 300%;  
  }  
  
  to {  
    margin-left: 0%;  
    width: 100%;  
  }  
}
```

이렇게도 됩니다!

from/to 대신에 0%/100%로 바꿔 쓸 수 있어요!

```
h1 {  
  animation-duration: 3s;  
  animation-name: slidein;  
}  
  
@keyframes slidein {  
  0% {  
    margin-left: 100%;  
    width: 300%;  
  }  
  
  100% {  
    margin-left: 0%;  
    width: 100%;  
  }  
}
```

그렇다면...

심지어 중간과정을 설정해줄 수도 있다는 어마무시한 사실!!!

아래 코드에 50% 상황을 하나 설정해봅시다.

```
h1 {  
  animation-duration: 3s;  
  animation-name: slidein;  
}  
  
@keyframes slidein {  
  0% {  
    margin-left: 100%;  
    width: 300%;  
  }  
  50% {  
    font-size: 100px;  
  }  
  100% {  
    margin-left: 0%;  
    width: 100%;  
  }  
}
```

뭐라고?!?! 이것도 된다고?!

마우스 오버되면 :hover 썼던거 기억나나요? 여기에도 animation 먹일 수 있어요~

```
div: hover {  
  animation-duration: 3s;  
  animation-name: smith;  
}
```

```
@keyframes smith {  
  0% {  
    opacity: 1;  
  }  
  100% {  
    opacity: 0;  
  }  
}
```

더 하기 전에 하나만 더! transform

CSS에는 transform 속성이 있습니다. 그리고 transform은 엘리먼트의 좌표정보를 바꿀 때 쓸 수 있죠. 예를 들면 엘리먼트를 돌리고, 옮기고, 키우고, 줄이고, 기울이는 등등에 적용할 수 있습니다. 그러면 우리가 원하는 빙글빙글 효과나, 왔다갔다 효과를 만들 수 있겠죠?? 자세한건 각자 더 공부해보시길 ㅎㅎㅎ

쓰는 법은 다음과 같습니다. 적당한 div하나 만들어서 적용해봅시다.

```
div {  
    transform: rotate(120deg) translate(100px, 200px);  
}
```

그러면 마우스를 올렸을 때 상자가 한바퀴 빙글 도는 애니메이션을 만들어볼까요?

아까 안 본 animation- 속성들을 살펴봅시다

animation 속성들

animation-name

animation의 이름

animation-delay

이벤트 발생 후 몇 초 후에 재생할지 지정

animation-duration

애니메이션을 몇 초 동안 재생할지 지정

animation-fill-mode

애니메이션이 끝나고 시작되는 상태를 결정

animation-timing-function

중간 상태들의 전환을 어떤 시간간격으로 진행할지 지정

animation-iteration-count

애니메이션 반복 횟수 지정

animation-direction

애니메이션 진행 방향을 설정

animation-play-state

애니메이션을 멈추거나 다시 시작

애니메이션 무한반복

animation-iteration-count: infinite

애니메이션 거꾸로 재생

animation-direction: reverse

애니메이션 그냥 재생 + 다음엔 거꾸로 재생

animation-direction: alternate



다 다루고 싶지만 너무 귀찮네요 ㅋㅋㅋ

아래 링크에서 각자 공부하는 걸로!!! ㅋ

https://developer.mozilla.org/ko/docs/Web/CSS/Using_CSS_animations

그래서 CSS Animation과 JS Animation 중에
무엇을 쓸 것인가?

엄마가 좋아? 아빠가 좋아?

그때 그때 달라요^^

CSS Animation과 JS Animation은 언제??

1. 아주 단순하고 간단한 애니메이션은 CSS로 처리한다!

확실히 CSS로 코딩하면 JS보다 코드량이나 구현 난이도에서 확연한 차이를 보입니다. 쉬운 길이 있다면 쉽게쉽게 ㅎㅎㅎ 그리고 CSS Animation은 JS Thread와는 별개로 작동하기 때문에 성능면에서도 효율적일 수 있습니다. 그래서 메뉴 옆에 간단한 이미지 애니메이션들은 CSS로 가볍게 처리해줍시다.

2. 아주 세밀한 애니메이션이나 동적으로 처리할 로직이 필요하다면 JS로 처리한다!

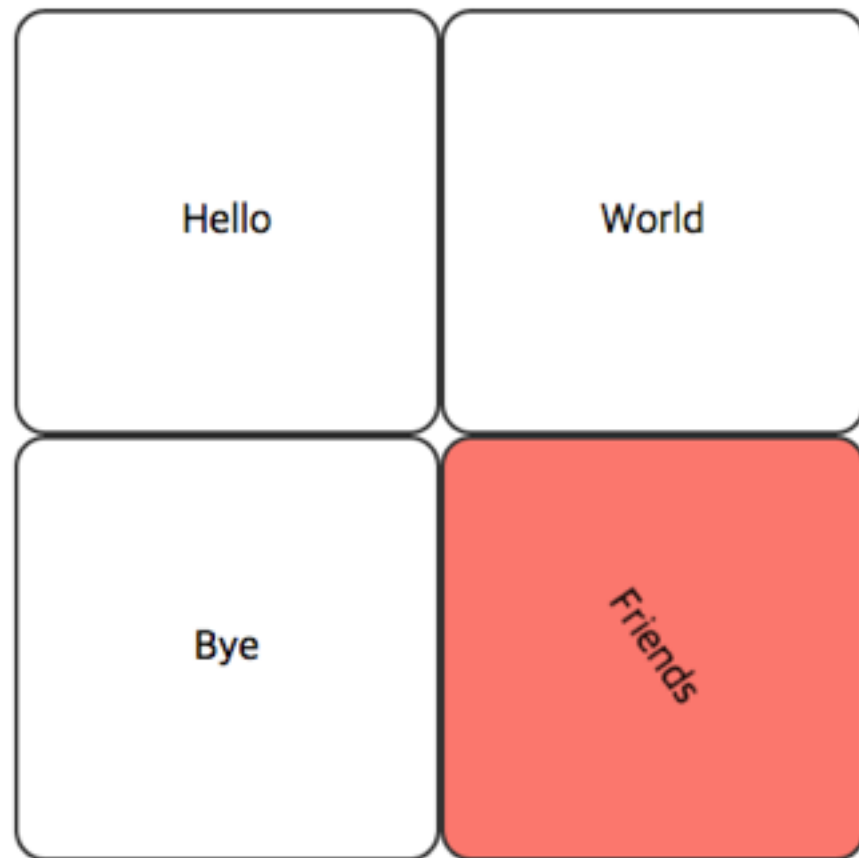
CSS는 애니메이션이 진행되는 도중에는 상태를 저장하거나 특정 값으로의 변경이 어렵습니다. 원하는 정도의 감속, 가속 등의 물리처리도 JS로 구현하는 것이 훨씬 정확합니다. 중간에 값이 동적으로 변한다면 역시 JS로 만들어줘야겠죠?

지금부터 실습할 내용은 Javascript로
CSS Animation을 제어하는 방법입니다.

전에도 우리 같이 본 적 있죠?

이제 직접 코딩해봐요~

오늘의 실습자료



네 개의 박스가 있고, 박스를 클릭하면
박스 색깔이 바뀌면서
안에 있는 문자가 빙글빙글 돌아갑니다.
다시 한 번 클릭하면 애니메이션 해제!

힌트1) 당연히 `addEventListener`를 써야겠죠?

힌트2) JS로 클릭된 상태를 관리하기위해서는

```
element.classList.add( ' animate ' );
```

```
element.classList.remove( ' animate ' );
```

힌트3) 추가된 `animate` class에 대해서

CSS Animation이 설정되어 있으면 끝!