애니메이션 기법

Vue는 진입/퇴장 및 리스트 전환을 처리하기 위해 `<Transition>` 및 `<TransitionGroup>` 컴포넌트를 제공합니다. 하지만 Vue 애플리케이션에서도 웹에서 애니메이션을 사용하는 다른 많은 방법들이 있습니다. 여기에서는 몇 가지 추가적인 기법에 대해 논의하겠습니다.

클래스 기반 애니메이션

DOM에 진입/퇴장하지 않는 요소의 경우, CSS 클래스를 동적으로 추가하여 애니메이션을 트리거할 수 있습니다:

```js
const disabled = ref(false)

function warnDisabled() {
  disabled.value = true
  setTimeout(() => {
    disabled.value = false
  }, 1500)
}
```

```template
<div :class="{ shake: disabled }">
  <button @click="warnDisabled">Click me</button>
  <span v-if="disabled">This feature is disabled!</span>
</div>
```

```css
.shake {
  animation: shake 0.82s cubic-bezier(0.36, 0.07, 0.19, 0.97) both;
  transform: translate3d(0, 0, 0);
}

@keyframes shake {
  10%,
  90% {
    transform: translate3d(-1px, 0, 0);
  }
```

```
  80% {
    transform: translate3d(2px, 0, 0);
  }

  30%,
  50%,
  70% {
    transform: translate3d(-4px, 0, 0);
  }

  40%,
  60% {
    transform: translate3d(4px, 0, 0);
  }
}
```

Click me

상태 기반 애니메이션

일부 전환 효과는 값을 보간하여 적용할 수 있습니다. 예를 들어, 상호작용이 발생하는 동안 스타일을 요소에 바인딩하는 방식입니다. 예를 들어 다음과 같습니다:

```js
const x = ref(0)

function onMousemove(e) {
  x.value = e.clientX
}
```

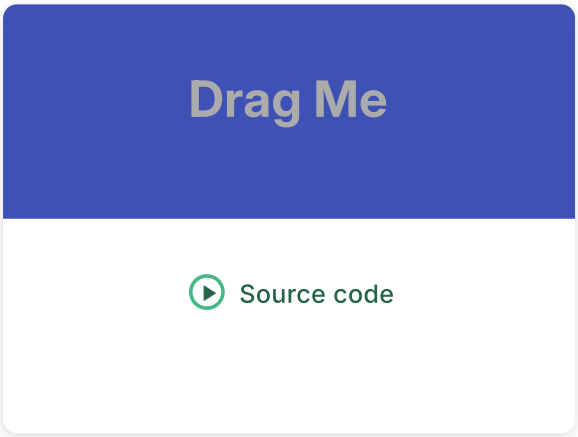
```template
<div
  @mousemove="onMousemove"
  :style="{ backgroundColor: `hsl(${x}, 80%, 50%)` }"
  class="movearea"
>
  <p>Move your mouse across this div...</p>
  <p>x: {{ x }}</p>
</div>
```

```css
.movearea {
  transition: 0.3s background-color ease;
}
```

Move your mouse across this div...

x: 0

색상 외에도 스타일 바인딩을 사용하여 transform, width, height 등을 애니메이션할 수 있습니다. 심지어 스프링 물리학을 사용하여 SVG 경로도 애니메이션할 수 있습니다. 결국 이들 모두는 속성 데이터 바인딩이기 때문입니다:



Watcher를 이용한 애니메이션

약간의 창의력을 발휘하면, watcher를 사용하여 어떤 수치 상태에 기반한 모든 것을 애니메이션할 수 있습니다. 예를 들어, 숫자 자체를 애니메이션할 수 있습니다:

```js
import { ref, reactive, watch } from 'vue'
import gsap from 'gsap'

const number = ref(0)
const tweened = reactive({
  number: 0
})

// 참고: Number.MAX_SAFE_INTEGER(9007199254740991)보다 큰 입력의 경우,
// JavaScript 숫자 정밀도의 한계로 인해 결과가 부정확할 수 있습니다.
watch(number, (n) => {
  gsap.to(tweened, { duration: 0.5, number: Number(n) || 0 })
})
```

```template
숫자를 입력하세요: <input v-model.number="number" />
<p>{{ tweened.number.toFixed(0) }}</p>
```

Type a number: 0

0

Playground에서 직접 실행해보기

GitHub에서 이 페이지 편집

Previous
Vue와 웹 컴포넌트