01_트랜지션기본.html

```html
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>트랜지션</title>
  <link rel="stylesheet" href="../css/main.css">
  <style>
    /* 1초 동안 투명도 변화 */
    .v-enter-active, .v-leave-active {
      transition: opacity 1s;
    }
    /* 더 이상 보이게 되지 않았을 때의 투명도 */
    .v-enter, .v-leave-to {
      opacity: 0;
    }
  </style>
</head>
<body>
  <div id="app">
    <p><button v-on:click="show=!show">변경</button></p>
    <transition>
      <div v-show="show">
        트랜지션 요소
      </div>
    </transition>
  </div>

  <!-- Vue.js -->
  <script src='https://cdnjs.cloudflare.com/ajax/libs/vue/2.6.9/vue.min.js'></script>
  <script>
    var app = new Vue( {
      el: '#app',
      data: {
        show: true
      }
    } );
  </script>
</body>
</html>
```

02_클래스이름과프리픽스.html

```html
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>트랜지션</title>
  <link rel="stylesheet" href="../css/main.css">
  <style>
    /* 1초 동안 투명도 변화 */
    .gt-enter-active, .gt-leave-active {
      transition: opacity 1s;
    }
    /* 더 이상 보이게 되지 않았을 때의 투명도 */
    .gt-enter, .gt-leave-to {
      opacity: 0;
    }
  </style>
</head>
<body>
  <div id="app">
    <p><button v-on:click="show=!show">변경</button></p>
    <transition name="gt">
      <div v-if="show">
        트랜지션 요소
      </div>
    </transition>
  </div>

  <!-- Vue.js -->
  <script src='https://cdnjs.cloudflare.com/ajax/libs/vue/2.6.9/vue.min.js'></script>
  <script>
    var app = new Vue( {
      el: '#app',
      data: {
        show: true
      }
    } );
  </script>
</body>
</html>
```

03_초기렌더링트랜지션.html

```html
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>트랜지션</title>
  <link rel="stylesheet" href="../css/main.css">
  <style>
    /* 1초 동안 투명도 변화 */
    .v-enter-active, .v-leave-active {
      transition: opacity 2s;
    }
    /* 더 이상 보이게 되지 않았을 때의 투명도 */
    .v-enter, .v-leave-to {
      opacity: 0;
    }
  </style>
</head>
<body>
  <div id="app">
    <!-- appear 속성을 추가하면 최초 렌더링시 트랜지션이 적용된다! -->
    <transition appear>
      <div v-if="show">
        처음 렌더링 시 트랜지션 적용
      </div>
    </transition>
  </div>

  <!-- Vue.js -->
  <script src='https://cdnjs.cloudflare.com/ajax/libs/vue/2.6.9/vue.min.js'></script>
  <script>
    var app = new Vue( {
      el: '#app',
      data: {
        show: true
      }
    } );
  </script>
</body>
</html>
```

04_enter와leave.html

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>트랜지션</title>
  <link rel="stylesheet" href="../css/main.css">
  <style>

    .v-enter-active,
    .v-leave-active {
      transition: opacity 1s, transform 1s;
    }
    /* 왼쪽에서 오른쪽으로 나타나기*/
    .v-enter {
      opacity: 0;
      transform: translateX(-30px);
    }
    /* 아래로 사라지기 */
    .v-leave-to {
      opacity: 0;
      transform: translateY(30px);
    }
  </style>
</head>
<body>
  <div id="app">
    <p><button v-on:click="show=!show">변경</button></p>
    <transition>
      <div v-show="show">
        트랜지션 요소
      </div>
    </transition>
  </div>

  <!-- Vue.js -->
  <script src='https://cdnjs.cloudflare.com/ajax/libs/vue/2.6.9/vue.min.js'></script>
  <script>
    var app = new Vue( {
      el: '#app',
      data: {
        show: true
      }
    } );
  </script>
</body>
</html>
```

05_그룹화.html

```html
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>트랜지션</title>
  <link rel="stylesheet" href="../css/main.css">
  <style>
    .v-enter-active, .v-leave-active {
      transition: opacity 1s;
    }
    .v-enter, .v-leave-to {
      opacity: 0;
    }

    /* 오류 수정 */
    .v-leave-active {
      position: absolute;
    }
  </style>
</head>
<body>
  <div id="app">
    <p><button v-on:click="show=!show">변경</button></p>
    <transition>
      <div v-if="show" key="a">TRUE</div>
      <div v-else key="b">FALSE</div>
    </transition>
  </div>

  <!-- Vue.js -->
  <script src='https://cdnjs.cloudflare.com/ajax/libs/vue/2.6.9/vue.min.js'></script>
  <script>
    var app = new Vue( {
      el: '#app',
      data: {
        show: true
      }
    } );
  </script>
</body>
</html>
```

06_enter와mode시점변경.html

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>트랜지션</title>
  <link rel="stylesheet" href="../css/main.css">
  <style>
    .v-enter-active, .v-leave-active {
      transition: opacity 1s;
    }
    .v-enter, .v-leave-to {
      opacity: 0;
    }

    .trans {
      width: 100px;
      border: 2px solid lightcoral;
      float: left;
      margin-right: 10px;
    }
  </style>
</head>
<body>
  <div id="app">
    <p><button v-on:click="show=!show">변경</button></p>

    <div class="trans">
      <h3>normal</h3>
      <transition>
        <div v-if="show" key="a">TRUE</div>
        <div v-else key="b">FALSE</div>
      </transition>
    </div>

    <div class="trans">
      <h3>out-in</h3>
      <transition mode="out-in">
        <div v-if="show" key="a">TRUE</div>
        <div v-else key="b">FALSE</div>
      </transition>
    </div>

    <div class="trans">
      <h3>in-out</h3>
      <transition mode="in-out">
        <div v-if="show" key="a">TRUE</div>
        <div v-else key="b">FALSE</div>
      </transition>
    </div>
  </div>

  <!-- Vue.js -->
```

06_enter와mode시점변경.html

```
  <script
  src='https://cdnjs.cloudflare.com/ajax/libs/vue/2.6.9/vue.min.js'></script>
  <script>
    var app = new Vue( {
      el: '#app',
      data: {
        show: false
      }
    } );
  </script>
</body>
</html>
```

07_키변화로트랜지션발동.html

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>트랜지션</title>
  <link rel="stylesheet" href="../css/main.css">
  <style>
    .trans {
      width: 50px;
      height: 50px;
      background-color: deeppink;
      color: white;
      text-align: center;
      line-height: 50px;
      font-size: 30px;
    }

    .v-enter-active {
      transition: opacity 1s;
    }

    /* 딜레이(0.2s)를 부여하여 문자 투명도 조정 */
    .v-leave-active {
      transition: opacity 0.8s ease 0.2s;
      position: absolute;
    }
    .v-enter, .v-leave-to {
      opacity: 0;
    }
  </style>
</head>
<body>
  <div id="app">
    <p><button v-on:click="count++">변경</button></p>
    <div class="trans">
      <transition>
        <div v-bind:key="count">{{ count }}</div>
      </transition>
    </div>
  </div>

  <!-- Vue.js -->
  <script src='https://cdnjs.cloudflare.com/ajax/libs/vue/2.6.9/vue.min.js'></script>
  <script>
    var app = new Vue( {
      el: '#app',
      data: {
        count: 0
      }
    } );
  </script>
```

```
</body>
</html>
```

08_이동트랜지션-1.html

```html
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>트랜지션</title>
  <link rel="stylesheet" href="../css/main.css">
  <style>
    /*
    요소가 추가/삭제/정렬 순서 변경으로 요소가 움직(이동)일 때 해당 요소에 .v-move 트랜지션 클래스가 적용된다!

    1초 동안 요소 이동 */
    .v-move {
      transition: transform 10s;
    }
  </style>
</head>
<body>
  <div id="app">
    <p><button v-on:click="order=!order">변경</button></p>

    <!-- transition-group으로 사용할 태그는 속성으로 지정 -->
    <transition-group tag="ul" class="list">
      <li v-for="item in sortedList" v-bind:key="item.id">
        {{ item.name }} {{ item.price }}원
      </li>
    </transition-group>
  </div>

  <!-- Vue.js -->
  <script src='https://cdnjs.cloudflare.com/ajax/libs/vue/2.6.9/vue.min.js'></script>
  <!-- lodash -->
  <script src='https://cdnjs.cloudflare.com/ajax/libs/lodash.js/4.17.21/lodash.min.js'></script>
  <script>
    new Vue( {
      el: '#app',
      data: {
        order: false,
        list: [
          { id: 1, name: '사과', price: 1000 },
          { id: 2, name: '바나나', price: 4000 },
          { id: 3, name: '수박', price: 5000 },
          { id: 4, name: '딸기', price: 3000 }
        ]
      },
      computed: {
        // order 값에 따라 리스트의 순서를 반전하는 산출 속성
        sortedList: function () {
          // Lodash의 orderBy 메서드 사용
```

08_이동트랜지션-1.html

```
          return _.orderBy(this.list, 'price', this.order ? 'desc' : 'asc')
        }
      }
    } );

  </script>
</body>
</html>
```

08_이동트랜지션-2.html

```html
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>트랜지션</title>
  <link rel="stylesheet" href="../css/main.css">
  <style>
    /* 박스 스타일 정의 */
    .list {
      width: 240px;
      padding: 0;
    }
    .item {
      display: inline-flex;
      justify-content: center;
      align-items: center;
      margin: 4px;
      width: 40px;
      height: 40px;
      background: #f5f5f5;
    }
    /* 트랜지션 전용 스타일 */
    .v-enter-active, .v-leave-active, .v-move {
      transition: all 1s;
    }
    .v-leave-active {
      position: absolute;
    }
    .v-enter, .v-leave-to {
      opacity: 0;
      background: #f9a3b1;
      transform: translateY(-30px);
    }
  </style>
</head>
<body>
  <div id="app">
    <p>
      <button v-on:click="doShuffle">셔플</button>
      <button v-on:click="doAdd">추가</button>
    </p>
    <transition-group tag="ul" class="list">
      <li v-for="(item, index) in list"
        v-bind:key="item"
        class="item"
        v-on:click="doRemove(index)">{{ item }}</li>
    </transition-group>
  </div>

  <!-- Vue.js -->
  <script
  src='https://cdnjs.cloudflare.com/ajax/libs/vue/2.6.9/vue.min.js'></script>
  <!-- lodash -->
  <script src='https://cdnjs.cloudflare.com/ajax/libs/lodash.js/4.17.21/lodash.min.js'></script>
  <script>
    new Vue( {
      el: '#app',
      data: {
        list: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
      },
      methods: {
        doShuffle: function () {
          this.list = _.shuffle(this.list)
        },
        doAdd: function() {
          var newNumber = Math.max.apply(null, this.list) + 1
          var index = Math.floor(Math.random() * (this.list.length + 1))
          this.list.splice(index, 0, newNumber)
        },
        doRemove: function (index) {
          this.list.splice(index, 1)
        }
      }
    } );
  </script>
</body>
</html>
```

08_이동트랜지션-3.html

```html
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>트랜지션</title>
  <link rel="stylesheet" href="../css/main.css">
  <style>
    /* 박스 스타일 정의 */
    .list {
      width: 240px;
      padding: 0;
    }
    .item {
      display: inline-flex;
      justify-content: center;
      align-items: center;
      margin: 4px;
      width: 40px;
      height: 40px;
      background: #f5f5f5;
    }
    /* 트랜지션 전용 스타일 */
    .v-enter-active, .v-leave-active, .v-move {
      transition: all 1s;
    }
    .v-leave-active {
      position: absolute;
    }
    .v-enter, .v-leave-to {
      opacity: 0;
      background: #f9a3b1;
      transform: translateY(-30px);
    }
  </style>
</head>
<body>
  <div id="app">
    <p>
      <button v-on:click="doShuffle">셔플</button>
      <button v-on:click="doAdd">추가</button>
    </p>
    <!-- before-leave 훅 추가 -->
    <transition-group tag="ul" class="list" v-on:before-leave="beforeLeave">
      <li v-for="(item, index) in list"
        v-bind:key="item"
        class="item"
        v-on:click="doRemove(index)">{{ item }}</li>
    </transition-group>
  </div>

  <!-- Vue.js -->
  <script
  src='https://cdnjs.cloudflare.com/ajax/libs/vue/2.6.9/vue.min.js'></script>
  <!-- lodash -->
  <script src='https://cdnjs.cloudflare.com/ajax/libs/lodash.js/4.17.21/lodash.min.js'></script>
  <script>
    new Vue( {
      el: '#app',
      data: {
        list: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
      },
      methods: {
        doShuffle: function () {
          this.list = _.shuffle(this.list)
        },
        doAdd: function() {
          var newNumber = Math.max.apply(null, this.list) + 1
          var index = Math.floor(Math.random() * (this.list.length + 1))
          this.list.splice(index, 0, newNumber)
        },
        doRemove: function (index) {
          this.list.splice(index, 1)
        },
        // leave 시 X&Y 위치 보정 메소드
        beforeLeave: function (el) {
          var style = window.getComputedStyle(el)
          el.style.left = el.offsetLeft - parseFloat(style.marginLeft, 10) + 'px'
          el.style.top = el.offsetTop - parseFloat(style.marginTop, 10) + 'px'
        }
      }
    } );
  </script>
</body>
</html>
```

09_SVG트랜지션.html

```html
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>트랜지션</title>
  <link rel="stylesheet" href="../css/main.css">
  <style>
    .v-enter-active, .v-leave-active {
      transition: all 1s;
    }
    .v-leave-active {
      position: absolute;
    }
    .v-enter, .v-leave-to {
      opacity: 0;
      transform: translateX(-20px);
    }

    button {
      display: block;
    }
  </style>
</head>
<body>
  <div id="app">
    <button v-on:click="toggle=!toggle">변경</button>
    <svg xmlns="http://www.w3.org/2000/svg" version="1.1">
      <!-- SVG 요소에 트랜지션 사용 -->
      <transition>
        <my-circle v-bind:fill="fill" v-bind:key="fill"></my-circle>
      </transition>
    </svg>
  </div>

  <!-- Vue.js -->
  <script src='https://cdnjs.cloudflare.com/ajax/libs/vue/2.6.9/vue.min.js'></script>
  <script>
    // SVG 요소 컴포넌트 정의
    Vue.component( 'my-circle', {
      template: '<circle cx="80" cy="75" r="50" v-bind:fill="fill"/>',
      props: {
        fill: String
      }
    } );

    new Vue( {
      el: '#app',
      data: {
        toggle: false
      },
      computed: {
```

09_SVG트랜지션.html

```
        fill: function () {
          return this.toggle ? 'lightpink' : 'skyblue'
        }
      }
    } );
  </script>
</body>
</html>
```

10_트랜지션훅.html

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>트랜지션</title>
  <link rel="stylesheet" href="../css/main.css">
  <style>
    .v-enter-active, .v-leave-active {
      transition: all 1s;
    }
    .v-enter, .v-leave-to {
      opacity: 0;
      transform: translateX(-20px);
    }
  </style>
</head>
<body>
  <div id="app">
    <p><button v-on:click="show=!show">변경</button></p>
    <transition
      v-on:before-enter="beforeEnter"
      v-on:enter="enter"
      v-on:after-enter="afterEnter"
      v-on:enter-cancelled="enterCancelled"
      v-on:before-leave="beforeLeave"
      v-on:leave="leave"
      v-on:after-leave="afterLeave"
      v-on:leave-cancelled="leaveCancelled">
      <div v-if="show">콘솔 로그 확인</div>
    </transition>
  </div>

  <!-- Vue.js -->
  <script
  src='https://cdnjs.cloudflare.com/ajax/libs/vue/2.6.11/vue.min.js'></script>
  <script>
    new Vue( {
      el: '#app',
      data: {
        show: true
      },
      methods: {
        // Enter
        beforeEnter: function (el) {
          // DOM에 요소가 추가되기 전
          console.log('before-enter')
        },
        enter: function (el, done) {
          // .v-enter가 있는 DOM에 요소가 추가된 후
          console.log('enter')
          setTimeout(done, 1000) // 1초 후에 enter를 종료하고 after-enter로 변경
        },
        afterEnter: function (el) {
          // 트랜지션이 종료되거나 enter에서 done() 호출 후
          console.log('after-enter')
        },
        enterCancelled: function (el) {
          // Enter 페이즈가 중간에 취소되었을 때
          console.log('enter-cancelled')
        },
        // Leave
        beforeLeave: function (el) {
          // 클래스가 추가되기 전
          console.log('before-leave')
        },
        leave: function (el, done) {
          // .v-leave가 추가된 후
          console.log('leave')
          setTimeout(done, 1000)
        },
        afterLeave: function (el) {
          // DOM에서 요소가 제거된 후 또는 leave에서 done() 호출 후
          console.log('after-leave')
        },
        // v-show와 함께 사용하는 경우에만 leaveCancelled 사용 가능
        leaveCancelled: function (el) {
          // Leave 페이즈가 중간에 취소되었을 때
          console.log('leave-cancelled')
        }
      }
    } );
  </script>
</body>
</html>
```