

2025년 상반기 K-디지털 트레이닝

스타일 적용

[KB] IT's Your Life

1 HTML의 스타일 적용

✓ 기존 HTML의 스타일 적용

- 클래스명, 속성명 지정시 케밥 표기법으로 지정
- Javascript 표현식에서는 '-' 표기가 연산자이므로 사용할 수 없음
→ 케밥 표기법 대신 카멜케이스 표기법 사용

```
<style>
    .test { background-color:aqua; color:brown; border:solid 1px black; }
</style>
<button class="test">버튼 1</button>
<button class="test">버튼 2</button>
<button class="test">버튼 3</button>
```

○ 스타일 적용 순서

요소의 기본 스타일 → .test 스타일 → .over 스타일 → 인라인 스타일

1 HTML의 스타일 적용

✓ 인라인 스타일

○ v-bind:style로 작성

- 데이터 속성에 자바스크립트 객체로 작성

케밥 표기법(kebob casing)	카멜 표기법(camel casing)
font-size	fontSize
background-color	backgroundColor

1 HTML의 스타일 적용

06-02.html

```
<div id="app">
  <button
    :style="style1"
    @mouseover.stop="overEvent"
    @mouseout.stop="outEvent"
  >
    테스트
  </button>
</div>
```

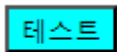


```
<script src="https://unpkg.com/vue"></script>
<script>
  let vm = Vue.createApp({
    name: "App",
    data() {
      return {
        style1: {
          backgroundColor: "aqua",
          color: "black",
        },
      };
    },
    methods: {
      overEvent() {
        this.style1.backgroundColor = "purple";
        this.style1.color = "yellow";
      },
      outEvent() {
        this.style1.backgroundColor = "aqua";
        this.style1.color = "black";
      },
    },
  }).mount("#app");
</script>
```

1 HTML의 스타일 적용

06-03.html

```
<div id="app">
  <button
    :style="{backgroundColor, color}"
    @mouseover.stop="overEvent"
    @mouseout.stop="outEvent"
  >
    테스트
  </button>
</div>
```



```
<script src="https://unpkg.com/vue"></script>
<script>
  let vm = Vue.createApp({
    name: "App",
    data() {
      return {
        backgroundColor: "aqua",
        color: "black",
      };
    },

    methods: {
      overEvent() {
        this.backgroundColor = "purple";
        this.color = "yellow";
      },
      outEvent() {
        this.backgroundColor = "aqua";
        this.color = "black";
      },
    },
  }).mount("#app");
</script>
```

1 HTML의 스타일 적용

06-04.html

```
<div id="app">
  <button :style="[myColor, myLayout]">테스트</button>
</div>

<script src="https://unpkg.com/vue"></script>
<script>
  let vm = Vue.createApp({
    name: 'App',
    data() {
      return {
        myColor: {
          backgroundColor: 'purple',
          color: 'yellow',
        },
        myLayout: {
          width: '150px',
          height: '80px',
          textAlign: 'center',
        },
      };
    },
  }).mount('#app');
</script>
```



2 CSS 클래스 바인딩

✓ CSS 클래스 지정

- `v-bind:class="..."` 또는 축약표현 `:class="..."`
 - 클래스명 문자열을 바인딩 방법
 - `true/false` 값을 가진 객체를 바인딩 하는 방법

2 CSS 클래스 바인딩

- ✓ CSS 클래스명 문자열을 바인딩하는 방법

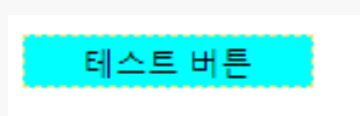
2 CSS 클래스 바인딩

06-05.html

```
<style>
  .buttonColor {
    background-color: aqua;
    color: black;
  }
  .buttonLayout {
    text-align: center;
    width: 120px;
  }
  .staticBorder {
    border: khaki dashed 1px;
  }
</style>

<div id="app">
  <button class="staticBorder" :class="myColor">
    테스트 버튼</button>
</div>
```

```
<script src="https://unpkg.com/vue"></script>
<script type="text/javascript">
  var vm = Vue.createApp({
    name: "App",
    data() {
      return {
        myColor: "buttonColor buttonLayout",
      };
    },
  }).mount("#app");
</script>
```



테스트 버튼

2 CSS 클래스 바인딩

06-06.html

```
<div id="app">
  <button class="staticBorder" :class="[myColor, myLayout]">테스트 버튼</button>
</div>

<script src="https://unpkg.com/vue"></script>
<script type="text/javascript">
  var vm = Vue.createApp({
    name: "App",
    data() {
      return {
        myColor: "buttonColor",
        myLayout: "buttonLayout",
      };
    },
  }).mount("#app");
</script>
```

2 CSS 클래스 바인딩

06-07.html

```

<style>
  .buttonColor {
    background-color: aqua;
    color: black;
  }
  .buttonLayout {
    text-align: center;
    width: 120px;
  }
  .staticBorder {
    border: khaki dashed 1px;
  }
</style>

<div id="app">
  <input type="checkbox" v-model="isMyLayout" />
  레이아웃 적용 여부<br />
  <button class="staticBorder"
    :class="[myColor, isMyLayout? myLayout: '']">
    버튼
  </button>
</div>

<script src="https://unpkg.com/vue"></script>

```

```

<script>
  let vm = Vue.createApp({
    name: 'App',
    data() {
      return {
        myColor: 'buttonColor',
        myLayout: 'buttonLayout',
        isMyLayout: false,
      };
    },
  }).mount('#app');
</script>

```

☐ 레이아웃 적용 여부
버튼

☒ 레이아웃 적용 여부
버튼

2 CSS 클래스 바인딩

- ✓ true/false 값을 가진 객체를 바인딩하는 방법
 - 3항 연산자로 클래스 지정 제어시 여러 개인 경우 복잡해짐
 - 객체로 바인딩하여
 - 속성명으로 클래스명을 지정
 - 속성값으로 해당 클래스 적용 여부를 true/false로 지정

2 CSS 클래스 바인딩

06-08.html

```

<style>
  .btnColor {
    background-color: aqua;
    color: black;
  }
  .btnAlign {
    text-align: center;
    width: 120px;
  }
  .btnBorder {
    border: khaki dashed 1px;
  }
</style>

<div id="app">
  <input type="checkbox" v-model="btnColor" />
  색상<br />
  <input type="checkbox" v-model="btnAlign" />
  정렬, 크기<br />
  <input type="checkbox" v-model="btnBorder" />

```

```

테두리<br />
  <button :class="{btnColor, btnAlign, btnBorder}">
    버튼</button>
</div>
<script src="https://unpkg.com/vue"></script>
<script>
  let vm = Vue.createApp({
    name: 'App',
    data() {
      return {
        btnColor: false,
        btnAlign: false,
        btnBorder: false,
      };
    },
  }).mount('#app');
</script>

```

☒ 색상
☒ 정렬, 크기
☐ 테두리

버튼

☐ 색상
☒ 정렬, 크기
☒ 테두리

버튼

2 CSS 클래스 바인딩

06-09.html

```
<style>
  .btnColor {
    background-color: aqua;
    color: black;
  }
  .btnAlign {
    text-align: center;
    width: 120px;
  }
  .btnBorder {
    border: khaki dashed 1px;
  }
</style>
```

```
<div id="app">
  <input type="checkbox" v-model="btnStyle.btnColor" />
  색상<br />
  <input type="checkbox" v-model="btnStyle.btnAlign" />
  정렬, 크기<br />
```

```
<input type="checkbox" v-model="btnStyle.btnBorder" />
테두리<br />
<button :class="btnStyle">버튼</button>
</div>
```

```
<script src="https://unpkg.com/vue"></script>
<script>
  let vm = Vue.createApp({
    name: 'App',
    data() {
      return {
        btnStyle: {
          btnColor: false,
          btnAlign: false,
          btnBorder: false,
        },
      };
    },
  }).mount('#app');
```

☒ 색상
☒ 정렬, 크기
☐ 테두리

버튼

☐ 색상
☒ 정렬, 크기
☒ 테두리

버튼

3 동적 스타일 바인딩

✓ 동적 스타일 바인딩

- 계산된 속성을 스타일 적용하기

3 동적 스타일 바인딩

06-10.html

```
<style>
  .score { border: solid 1px black; }
  .warning {
    background-color: skyblue;
    color: purple;
  }
  .warnimage {
    width: 18px;
    height: 18px;
    top: 5px;
    position: relative;
  }
</style>

<div id="app">
  <div>
    <p>1부터 100까지만 입력가능합니다.</p>
    <div>
      점수 : <input type="text" class="score" v-model.number="score" :class="info" />
      
    </div>
  </div>
</div>
```


3 동적 스타일 바인딩

06-10.html

```
<script src="https://unpkg.com/vue"></script>
<script type="text/javascript">
  var vm = Vue.createApp({
    name: "App",
    data() {
      return { score: 50 };
    },
    computed: {
      info() {
        return { warning: this.score < 1 || this.score > 100 };
      },
    },
  }).mount("#app");
</script>
```

1부터 100까지만 입력가능합니다.

점수 :

1부터 100까지만 입력가능합니다.

점수 : 