

2025년 상반기 K-디지털 트레이닝

컴포넌트 심화

[KB] IT's Your Life

- ✓ **style-test 프로젝트를 생성하세요.**
 - 관련 패키지 설치
 - 새로운 작업을 위한 기본 골격을 위해 필요 없는 파일 삭제 및 수정

- ✓ **다음 컴포넌트 3개를 추가하세요.**
 - Child1.vue
 - Child2.vue
 - Child3.vue

✓ Child1, Child2, Child3 컴포넌트의 내용을 다음처럼 작성하세요.

○ 각 컴포넌트의 이름 출력

```
<template>
  <div class="child">
    <h2>Child1</h2>
  </div>
</template>

<script>
export default {
  name: 'Child1',
};
</script>

<style>
.child {
  background-color: yellow;    // Child2는 blue, Child3는 orange
  border: solid 1px black;
  margin: 1.5em;
  padding: 1em;
}
</style>
```

- ✓ App.vue에 Child1, Child2, Child3 컴포넌트를 지역 컴포넌트로 배치하세요.

```
<template>
  <div>

  </div>
</template>

<script>
export default {
  name: 'App',

};
</script>
```

- ✓ Child1, Child2의 `<style>`을 `<style scoped>`로 변경하고, 브라우저 콘솔의 DOM을 확인하세요.

☑ Child3.vue을 <style>을 module로 설정한 경우 다음 코드를 완성하세요.

```
<template>
  <div :class="_____ ">
    <h2>Child3</h2>
  </div>
</template>

<script>
export default {
  name: 'Child3',
  created() {
    console.log(this._____);
  },
};
</script>

<style module>
.child {
  background-color: orange;
  border: solid 1px black;
  margin: 1.5em;
  padding: 1em;
}
</style>
```

✓ slot-test 프로젝트를 생성하세요.

- 관련 패키지 설치
- 새로운 작업을 위한 기본 골격을 위해 필요 없는 파일 삭제 및 수정

✓ 다음 컴포넌트를 추가하세요.

- CheckBox1.vue
- NoSlotTest.vue

✓ CheckBox1 컴포넌트를 다음과 같이 정의하세요.

```
<template>
  <div>
    <input type="checkbox"
      :value="id"
      :checked="checked"
      @change="$emit('check-changed', {id, checked: $event.target.checked })" />
    <span v-if="checked === true" style="color:blue; text-decoration:underline;">
      <i>{{label}}</i>
    </span>
    <span v-else style="color:gray">{{label}}</span>
  </div>
</template>

<script>
export default {
  name : "CheckBox1",
  props : ["id", "checked", "label"]
}
</script>
```


✓ NoSlotTest 컴포넌트를 다음과 같이 정의하세요.

```
<script>
import CheckBox1 from './CheckBox1.vue';

export default {
  name : "NoSlotTest",
  components : { CheckBox1 },
  data() {
    return {
      items : [
        { id:"V", checked:true, label:"Vue" },
        { id:"A", checked:false, label:"Angular" },
        { id:"R", checked:false, label:"React" },
        { id:"S", checked:false, label:"Svelte" },
      ]
    }
  },
  methods : {
    CheckBoxChanged(e) {
      let item = this.items.find((item)=> item.id === e.id);
      item.checked = e.checked;
    }
  }
}
</script>
```

- ✓ NoSlotTest 컴포넌트를 다음과 같이 정의하세요. 필요한 경우 코드를 완성하세요.

```
<template>
  <div>
    <h3>당신이 경험한 프론트 엔드 기술은?(첫번째:Slot사용(X))</h3>
    <CheckBox1 _____ :key="item.id"
      _____="item.id"
      _____="item.label"
      _____="item.checked"
      @_____="CheckBoxChanged">
    </CheckBox1>
  </div>
</template>
```

✓ App.vue를 다음과 같이 정의하고, 결과를 확인하세요.

```
<template>
  <div>
    <NoSlotTest />
  </div>
</template>

<script>
import NoSlotTest from './components/NoSlotTest.vue'

export default {
  name: 'App',
  components: { NoSlotTest }
}
</script>
```

당신이 경험한 프론트 엔드 기술은?(첫번째:Slot사용(X))

☒ [Vue](#)

☐ Angular

☐ React

☐ Svelte

✔ 다음과 같이 CheckBox2 컴포넌트를 정의하세요.

```
<template>
  <div>
    <input
      type="checkbox"
      :value="id"
      :checked="checked"
      @change="$emit('check-changed', { id, checked: $event.target.checked })"
    />
    <slot>Item</slot>
  </div>
</template>

<script>
export default {
  name: 'CheckBox2',
  props: ['id', 'checked'],
};
</script>
```

- ✓ NoSlotTest와 동일하게 동작하도록 SlotTest 컴포넌트를 정의하세요.
 - 템플릿 파트를 slot을 이용하는 형태로 변경하세요.
- ✓ App.vue를 다음과 같이 수정하고 결과를 확인하세요.

```
<template>
  <div>
    <NoSlotTest />
    <SlotTest />
  </div>
</template>

<script>
import NoSlotTest from './components/NoSlotTest.vue'
import SlotTest from './components/SlotTest.vue'

export default {
  name: 'App',
  components: { NoSlotTest, SlotTest }
}
</script>
```