

# Vue 기본 문법 살펴보기

## Vue3 js 테스트

### CDN 방식 기본 구조

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <title>Vue3</title>
  <script src="https://unpkg.com/vue@3/dist/vue.global.js"></script>
</head>
<body>
  <div id="counter">
    <button type="button">Counter:0</button>
  </div>
  <script>
    const app = Vue.createApp({
      // 옵션 추가 가능
    })
  </script>
</body>
</html>
```

### vue 인스턴스 생성 및 컨테이너 위치 지정

```
// vue 인스턴스(객체) 생성
const app = Vue.createApp({ })

// vue 식별자를 통한 컨테이너 위치 지정
app.mount("#counter")
```

### vue 변수를 html에 렌더링 하기

```
<body>
  <div id="app">
    <button type="button">Counter: {{ cnt }}</button>
  </div>
  <script>
    const app = Vue.createApp({
      // 옵션을 통해 vue활용한 웹 어플리케이션 조작
      data() {
```

```

        return {
          cnt: 200
        }
      }
    })

    //인스턴스를 통해서 렌더링할 컨테이너 위치를 지정할 수 있음
    // app.mount("컨테이너_요소_선택터")
    app.mount("#app")
  </script>
</body>

```

## event 발생으로 요소 제어하기

```

<body>
  <div id="counter">
    <button type="button" v-on:click="cnt++">Counter: {{ cnt }}</button>
  </div>
  <script>
    const app = Vue.createApp({
      // 옵션을 통해 vue활용한 웹 어플리케이션 조작
      data() {
        return {
          cnt: 200
        }
      }
    })

    //인스턴스를 통해서 렌더링할 컨테이너 위치를 지정할 수 있음
    // app.mount("컨테이너_요소_선택터")
    app.mount("#counter")
  </script>
</body>

```

## Vue의 두 가지 핵심 기능

- 선언적 렌더링(Declarative Rendering)
  - 템플릿 구문( {{ 데이터 }})을 활용하여 데이터를 선언적으로 출력(렌더링) 할 수 있도록 함
- 반응성(Reactivity)
  - JavaScript 상태 변경을 자동으로 추적하고 변경이 발생하면 DOM을 효율적으로 업데이트함

## 바인딩(v-binding)

엘리먼트(html 요소) 속성에 데이터를 바인딩(연결) 할 수 있음

```

<body>
  <div id="app">

```

```

<button type="button" v-on:click="cnt++">Counter: {{ cnt }}</button><br>
<input type="text" v-bind:placeholder="message">
</div>
<script>
const app = Vue.createApp({
  // 옵션을 통해 vue활용한 웹 어플리케이션 조작
  data() {
    return {
      cnt: 200,
      message: '헬로 vue'
    }
  }
})

//인스턴스를 통해서 랜더링할 컨테이너 위치를 지정할 수 있음
// app.mount("컨테이너_요소_선택터")
app.mount("#app")
</script>
</body>

```

## 디렉티브 속성

- **v-bind** 속성은 데이터(상태) 속성에 바인딩할 때 사용하는 특수 속성임
- 반응형 동작 : 바인딩 된 DOM은 **placeholder** 속성을 Vue 인스턴스의 **message** 속성으로 최신 상태를 유지함
- **v-** 접두어가 붙은 특수 속성을 디렉티브(directive)라고 함

## 이벤트 핸들링(v-on)

**v-on** 디렉티브 : Vue 인스턴스의 메서드(methods)를 호출하는 이벤트 리스터를 추가할 수 있음

```

<body>
  <div id="app">
    <button type="button" v-on:click="cnt++">Counter: {{ cnt }}</button><br>
    <input type="text" v-bind:placeholder="message">
    <hr>
    <button v-on:click="hello">클릭</button>
  </div>
  <script>
    const app = Vue.createApp({
      // 옵션을 통해 vue활용한 웹 어플리케이션 조작
      data() {
        return {
          cnt: 200,
          message: '헬로 vue'
        }
      },
      methods: {
        hello() {
          alert("헬로 뷰~")
        }
      }
    })
    app.mount("#app")
  </script>
</body>

```

## 양방향 바인딩(v-model)

**v-model** 디렉티브

form 내의 요소인 (input, select 등)의 입력과 앱의 상태(data)를 양방향으로 바인딩함

```
<body>
<div id="app">
  <p>{{ bndMessage }}</p>
  <input type="text" v-bind:value="username"> <br>
  <input type="text" v-model="bndMessage">
</div>
<script>
  const app = Vue.createApp({
    // 옵션을 통해 vue활용한 웹 어플리케이션 조작
    data() {
      return {
        username: "코즈랩",
        bndMessage: "보라색",
      }
    }
  })
</script>
</body>
```

## 조건문 바인딩(v-if)

**v-if** 디렉티브

html 요소의 표시 여부를 제어할 수 있음

```
<body>
<button type="button" v-on:click="visible=true">보여주기</button>
<p v-if="visible">요소 보기</p>
</body>
<script>
  const app = Vue.createApp({
    // 옵션을 통해 vue활용한 웹 어플리케이션 조작
    data() {
      return {
        visible: false,
      }
    },
  })
  app.mount("#app")
</script>
```

# 반복문 바인딩(v-for)

**v-for** 디렉티브

배열에서 데이터를 가져와 아이템 목록을 표시할 때 사용함

```
<div id="app">
  <h3>v-for 디렉티브(반복문)</h3>
  <ul>
    <li v-for="item in items">{{ item }}</li>
  </ul>
  <ul>
    <li v-for="subject in subjects">{{ subject }}
      <span v-for="(sub, idx) in subject"> {{ sub }}&nbsp;&nbsp;&nbsp;</span>
    </li>
  </ul>
</div>
<script>
  const app = Vue.createApp({
    // 옵션을 통해 vue활용한 웹 어플리케이션 조작
    data() {
      return {
        items:["프론트엔드", "백엔드", "머신러닝", "빅데이터"],
        subjects:{ "프론트엔드": ["html", "css", "js", "vue"], "백엔드":["python", "django", "mysql"]}
      }
    }
  })
  app.mount("#app")
</script>
```

## Vue는

- User Interface를 개발하기 위한 JavaScript Framework
- html, css, javascript를 기반으로 단순, 복잡한 사용자 UI를 효율적으로 개발하는데 도움을 줌