

프론트엔드 개발자를 위한 Vue.js 시작하기

Vue.js </> 시작하기

Directive



한국기술교육대학교
온라인평생교육원

학습내용

- Directive 선언
- Directive 등록

학습목표

- Directive를 선언하고 Lifecycle 함수를 이용하는 방법에 대해 설명할 수 있다.
- Directive의 Global, Local 등록 사용 방법에 대해 설명할 수 있다.

Directive 선언

Directive 선언

1 Directive

- Component는 동적 UI 구성을 목적으로 하며 태그로 이용됨
- Directive는 속성으로 이용되며 속성이 적용된 DOM에 대한 추가 작업을 목적으로 함

Directive 선언

1 Directive

- Vue.js에서 제공하는 Directive가 있으며 개발자가 직접 만들어 사용할 수 있음

```
<div v-show="false">I am Show</div>
```

Directive 선언

Directive 선언

2 Custom Directive

- Custom Directive는 Object Literal로 만들어짐
- Directive의 Lifecycle 함수를 이용해 Directive에서 실행할 로직을 담음

```
var testDirective = {
  mounted(el) {
    el.focus()
  }
}
```

Directive 선언

3 Directive Lifecycle

- Directive의 Lifecycle은 Directive가 사용되는 Element(Component)의 Lifecycle과 유사하지만 차이가 있음

beforeMount()

Directive가 Element에 바인딩 되면 호출되며 한번만 호출됨

mounted()

Element가 부모 DOM에 삽입되면 호출

beforeUpdate()

Element가 업데이트 되기 전에 호출

Directive 선언

Directive 선언

3 Directive Lifecycle

- Directive의 Lifecycle은 Directive가 사용되는 Element(Component)의 Lifecycle과 유사하지만 차이가 있음

updated()

Element가 업데이트 되면 호출

beforeUnmount()

Element가 마운트 해제되기 전에 호출

unmounted()

Directive가 Element에서 제거되면 호출

Directive 선언

3 Directive Lifecycle

- Directive의 Lifecycle은 Directive가 사용되는 Element(Component)의 Lifecycle과 유사하지만 차이가 있음

```
var testDirective = {
  beforeMount() { },
  mounted() { },
  beforeUpdate() { },
  updated() { },
  beforeUnmount() { },
  unmounted() { },
}
```

Directive 선언

Directive 선언

4 el

- Lifecycle 함수의 매개변수로 전달
- Directive가 사용된 Element 객체

Directive 선언

4 el

- el 객체를 이용해 Directive가 적용된 Element의 조작 작업 가능

```
mounted(el) {  
  el.innerHTML = `  
    el.tagName : ${el.tagName} <br/>  
    el.parentNode.tagName : ${el.parentNode.tagName}  
  <br/>  
`  
}
```

Directive 선언

Directive 선언

5 binding

- Lifecycle 함수의 매개변수로 전달
- Directive와 관련된 다양한 정보 추출 가능함

Directive 선언

5 binding

value

- Directive에 대입된 값
- v-myDirective="2"이면 2값이 대입

oldValue

- 변경되기 전의 값
- beforeUpdate()와 updated()에서만 사용 가능

Directive 선언

Directive 선언

5 binding

arg

- Directive에 적용된 인자
- v-myDirective:foo이면 “foo” 대입

modifiers

- Directive에 적용된 수식어를 포함하는 객체
- v-myDirective.foo.bar이면
{foo:true, bar:true}

Directive 선언

5 binding

```
mounted(el, binding) {
  el.innerHTML = `
    binding.value : ${binding.value} </br/>
    binding.arg : ${binding.arg} <br/>
    binding.modifiers :
    ${JSON.stringify(binding.modifiers)}
  `
}
```


Directive 등록

Directive 등록

1 Global 등록 사용

- application instance에 Directive 등록
- application 전역에서 Directive 이용
- application instance의 directive() 함수로 등록

Directive 등록

1 Global 등록 사용

- 첫 번째 매개변수가 Directive 속성명

```
var app = createApp(MyComponent)
app.directive('testDirective', TestDirective)
app.mount('#app')
```

Directive 등록

Directive 등록

1 Global 등록 사용

- 등록된 Directive 이용

```
<template>
  <div>
    <input v-testDirective />
  </div>
</template>
```

Directive 등록

2 Local 등록 사용

- Directive가 필요한 Component에 등록 사용
- Component의 directives 속성을 이용해 등록

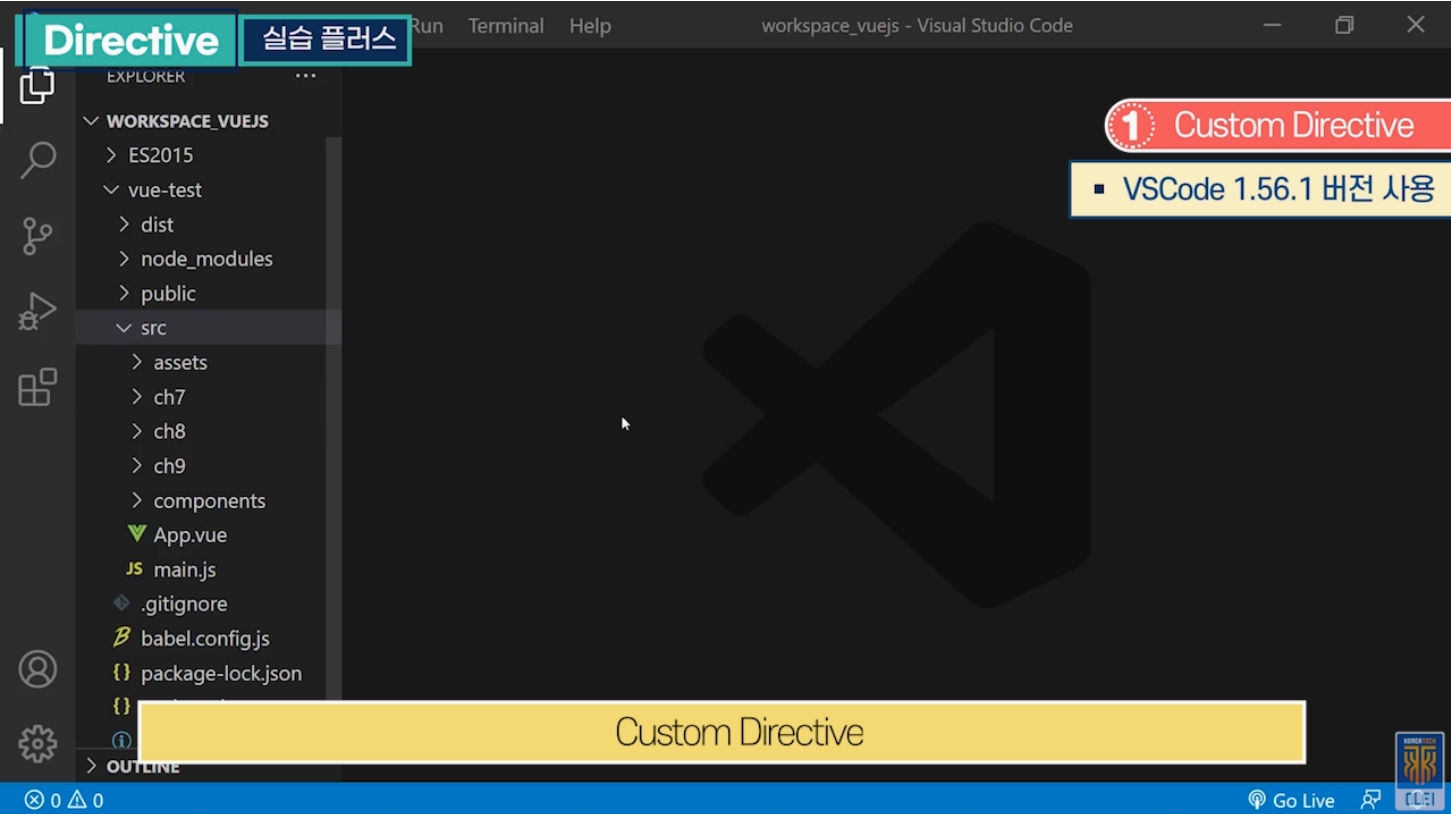
Directive 등록

Directive 등록

2 Local 등록 사용

```
export default {  
  data() {  
    return {  
      msg: "hello",  
      data1: 10  
    };  
  },  
  directives: {  
    myDirective  
  }  
};
```

실습 플러스



실습단계
src → ch10 폴더 생성
생성한 Directive를 Component에서 활용
생성한 폴더 내 myDirective.js 자바스크립트 파일 생성
Directive에 LifeCycle 함수 선언
console로 전달되는 정보 확인
LifeCycle 함수 선언 후 전달받은 Directive와 관련된 정보 확인
Tag에 정보 출력
Directive가 사용된 Tag 명 출력
Directive가 사용된 부모 Tag 정보
Directive에 설정된 값
Directive에 추가된 Argument 정보
Modifiers 값까지 출력
UI 효과 추가
외부에서 이용하기 위해 export 작업

실습 플러스

실습단계
Directive를 이용할 Component를 정의해 줘야 함
사용하고자 하는 Directive 파일 import
Component 정의
Directive 등록 방법 2가지 1. Global로 등록 2. Local로 등록
Local로 등록
HTML Tag에서 사용하기 위해 template 정의
Directive 속성 값 지정
테스트용 Directive에 부가적인 정보 추가
Directive 정의와 Directive를 이용한 Component 정의
이전에 생성한 ch9의 index.html, main.js, vue.config.js 파일 복사하여 이용
Vue 명령으로 실행
브라우저에 Component 출력
전달된 값 확인