

프론트엔드 개발자를 위한 Vue.js 시작하기

Vue.js </> 시작하기

Component Option과 Data Binding



한국기술교육대학교
온라인평생교육원

학습내용

- Component Option
- Data Binding

학습목표

- **Component Option**을 이해하고 사용하는 방법에 대해 설명할 수 있다.
- **Component template**에 data를 바인딩할 수 있다.

Component Option

Component Option

1 components

- Child Component를 등록하기 위한 옵션

```
components: {  
  MySub  
}
```

Component Option

2 template

template

- Component의 UI Template
- 데이터가 바인딩 되어야 하는 부분을 {{ }}으로 표현

Component Option

Component Option

2 template

- Component의 데이터가 변경되면 Template에 의해 출력되는 부분도 변경

```
template: `
  <div>
    <div>message : {{ msg }}</div>
  </div>
`
```

Component Option

3 data

data

- Component를 위한 data 설정
- data에 설정된 값을 template, method 등에서 이용

Component Option

Component Option

3 data

- data 옵션은 함수 형식으로 선언

```
data() {
  return {
    msg: 'hello',
    arrayData: [
      { message: 'hello' },
      { message: 'world' }
    ]
  }
},
```

Component Option

4 methods

- Component에 메서드를 등록하기 위한 옵션

```
<p>Reversed message: "{{ reversedMessage() }}"</p>
.....
methods: {
  reversedMessage: function () {
    return this.msg.split('').reverse().join('')
  }
}
```

Component Option

Component Option

5 computed

computed

- 템플릿 표현식의 복잡함을 줄이기 위해서 제공
- 표현식을 함수로 정의한 후 computed 속성에 등록하면 템플릿에서 Getter Property로 이용

Component Option

5 computed

computed

- 캐싱 제공

methods

- 캐싱 기능이 없음

Component Option

Component Option

5 computed

- 값이 바뀌지 않으면 반복 호출되지 않음

```
<p>Computed reversed message: "{{ reversedMessage }}"</p>
.....
computed: {
  reversedMessage: function () {
    return this.msg.split('').reverse().join('')
  }
}
```

Component Option

5 computed

- 기본적으로는 Getter 함수만 제공이 되지만 원한다면 Setter 함수 추가 가능
- Setter 함수도 Property로 이용함
fullName='kim'

```
fullName: {
  get: function () {
    .....
  },
  set: function (newValue) {
    .....
  }
}
```

Component Option

Component Option

6 watch

● Data 변경 감지

```
watch: {
  firstName: function (val) {
    this.fullName = val + ' ' + this.lastName
  },
  lastName: function (val) {
    this.fullName = this.firstName + ' ' + val
  }
}
```


Data Binding

Data Binding

1 Component Data Binding

- Component Data를 화면에 출력함
- Data는 `data() {}`에 명시
- Template에서 Data Binding은 `{{ }}` 표현식 이용

Data Binding

1 Component Data Binding

```
<template>
  <div>
    Hello,
    {{ name }}
  </div>
</template>
<script>
export default {
  data(){
    return {
      name: ' vue...'
    }
  }
};
</script>
```

Data Binding

Data Binding

2 Tag Attribute 값에 Data Binding

- HTML 속성 값에 데이터 바인딩은 **{{ }}**을 사용할 수 없음
- v-bind**를 이용해 속성 값을 지정함

```
<div v-bind:id="dynamicId"></div>
.....
data(){
  return {
    dynamicId: 'a1'
  }
}
```

Data Binding

2 Tag Attribute 값에 Data Binding

- 값이 Boolean이 대입되는 속성 경우 데이터 값이 **false**이며, **undefined, null**이면 속성이 추가되지 않음

```
<button v-
bind:disabled="isButtonDisabled">Button</button>
.....
data(){
  return {
    isButtonDisabled: true
  }
}
```

Data Binding

Data Binding

3 Data Binding 관련 Directive

v-once

초기 데이터 출력 후 데이터 변경에도 업데이트 되지 않는 Template

```
<span v-once>This will never change:
{{ msg }}</span>
```

v-html

rawHtml로 출력

```
<p>Using mustaches: {{ rawHtml }}</p>
<p>Using v-html directive:
<span v-html="rawHtml"></span></p>
```

Data Binding

4 표현식 출력

- Template의 {{ }} 혹은 v-bind 안에 자바스크립트 표현식이 가능
- 단일 표현식만 가능

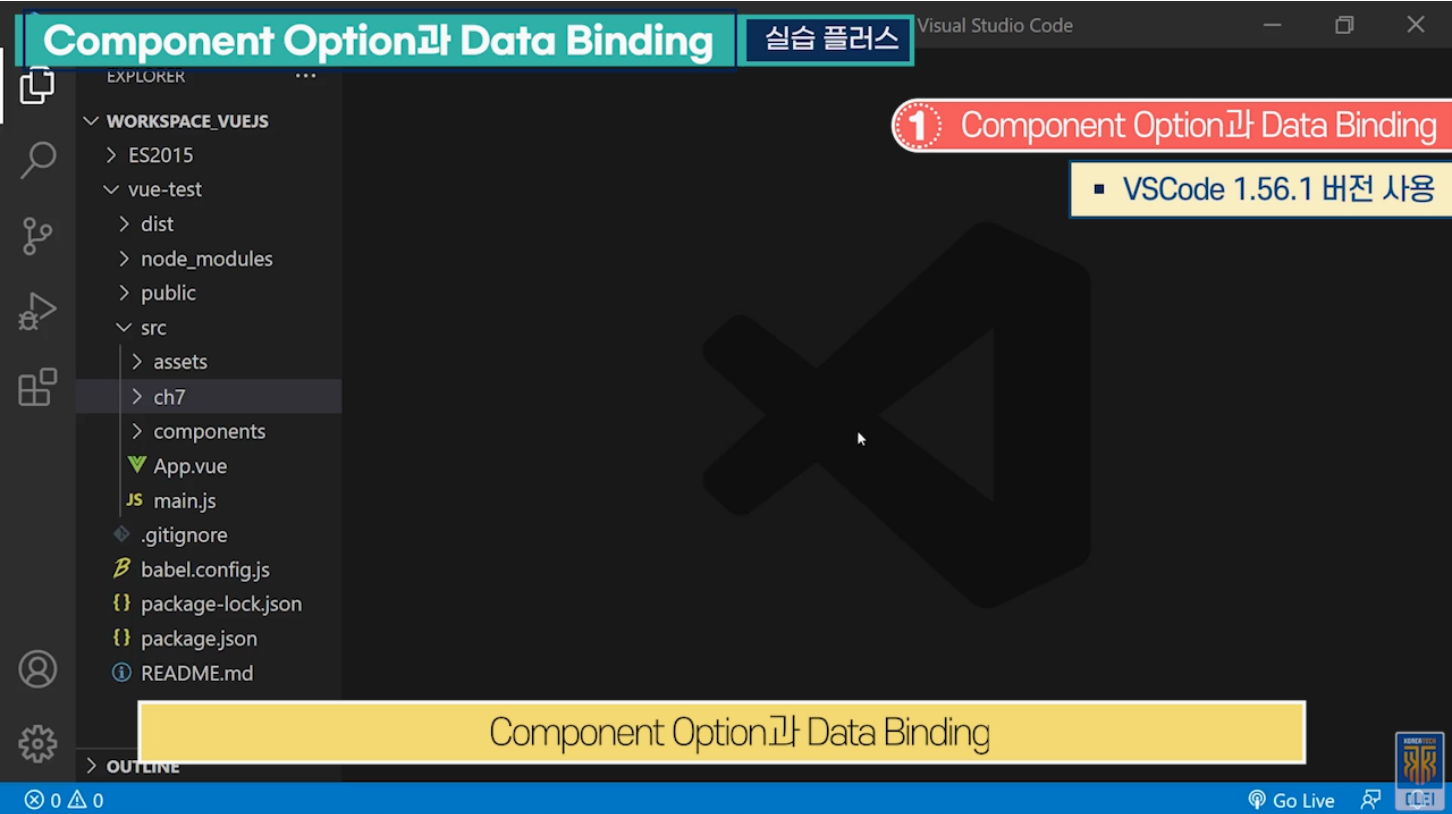
Data Binding

Data Binding

4 표현식 출력

```
{{ number + 1 }}  
{{ ok ? 'YES' : 'NO' }}  
{{ msg.split('').reverse().join('') }}  
  
<div v-bind:id="'list-' + number"></div>  
.....  
data(){  
  return {  
    number: 10,  
    ok: 'YES'  
  }  
}
```

실습 플러스



실습단계
src → ch8 서브 폴더 생성
서브 폴더 생성 후 Vue 파일 생성
Component 정의
script 부분을 먼저 작성하고 template 작성
Vue data 선언
computed : 템플릿 표현식의 복잡함을 줄이기 위해 제공하는 옵션
어떤 함수가 몇 번 호출되었는지 확인하기 위해 console 작성
methods : 함수 스타일로 정의됨
watch 옵션 정의
Vue 화면 구성을 위한 템플릿 정의
data 값 출력
v-once가 있어서 초기 값은 동일하게 출력되지만 변경된 값의 반영 여부가 다름
button 정의
속성 값에 대한 출력이기 때문에 v-bind 이용

실습 플러스

실습단계
선언되어 있는 computed 값을 출력
method 호출하여 값을 출력
테스트를 위해 한 번 더 출력하게 복사해서 지정
computed와 method를 2번 호출하기 위한 의도로, 반복 호출되는지 혹은 한 번만 호출되는지 확인
유저에게 입력을 받고 변경되는 데이터를 watch에서 인식하여 작성된 값으로 변경 가능한지 확인
옵션 정의, 옵션 테스트를 위한 코드 완성
이전에 생성한 ch7의 html, js, vue.config.js 파일 복사하여 이용
main.js 파일 수정하여 사용
테스트 진행
브라우저에 Component 출력
computed : 코드에서 2번 이용해도 데이터가 변경되지 않았기 때문에 한 번만 호출됨 method : 데이터가 변경되지 않아도 2번 호출 가능
초기화된 값 반영