SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

삼성 청년 SW 아카데미

Vue.js

SAMSUNG SW ACADEMY FOR YOUTH

Vue Event

- Vue Life Cycle
- Vue Event Handling
- Vue Bindings

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

Vue Life Cycle

Vue Life Cycle

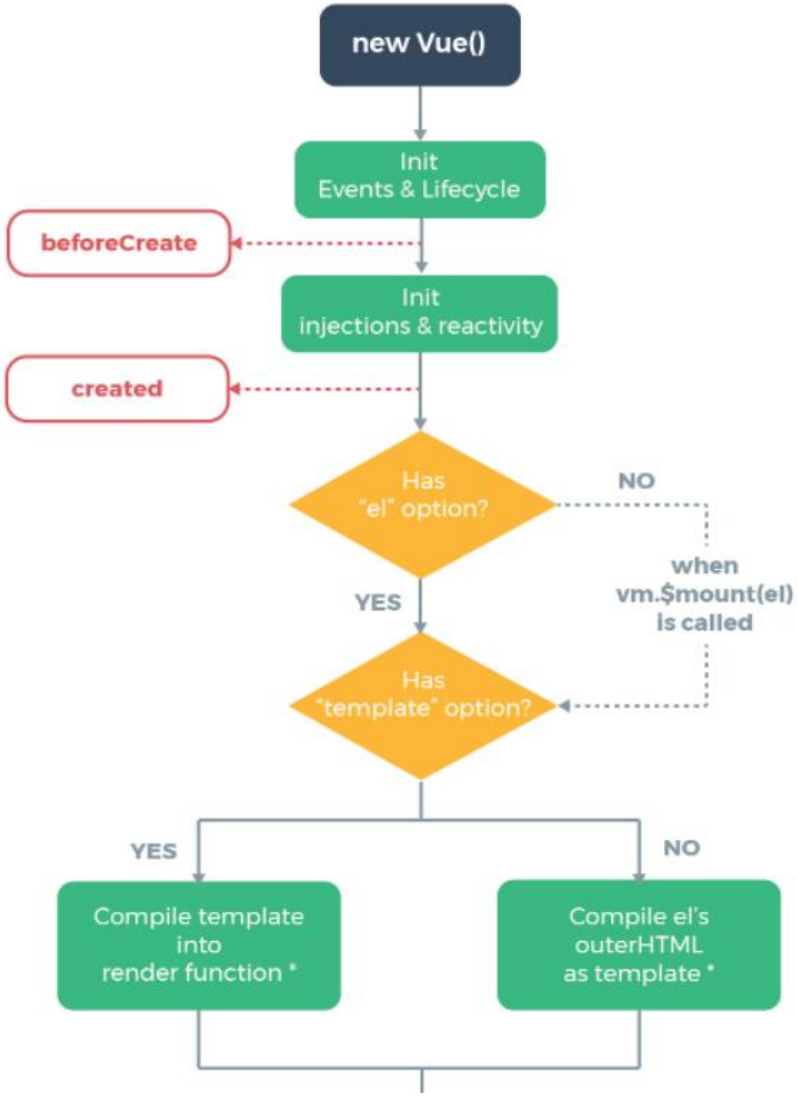
Instance Life Cycle Hooks

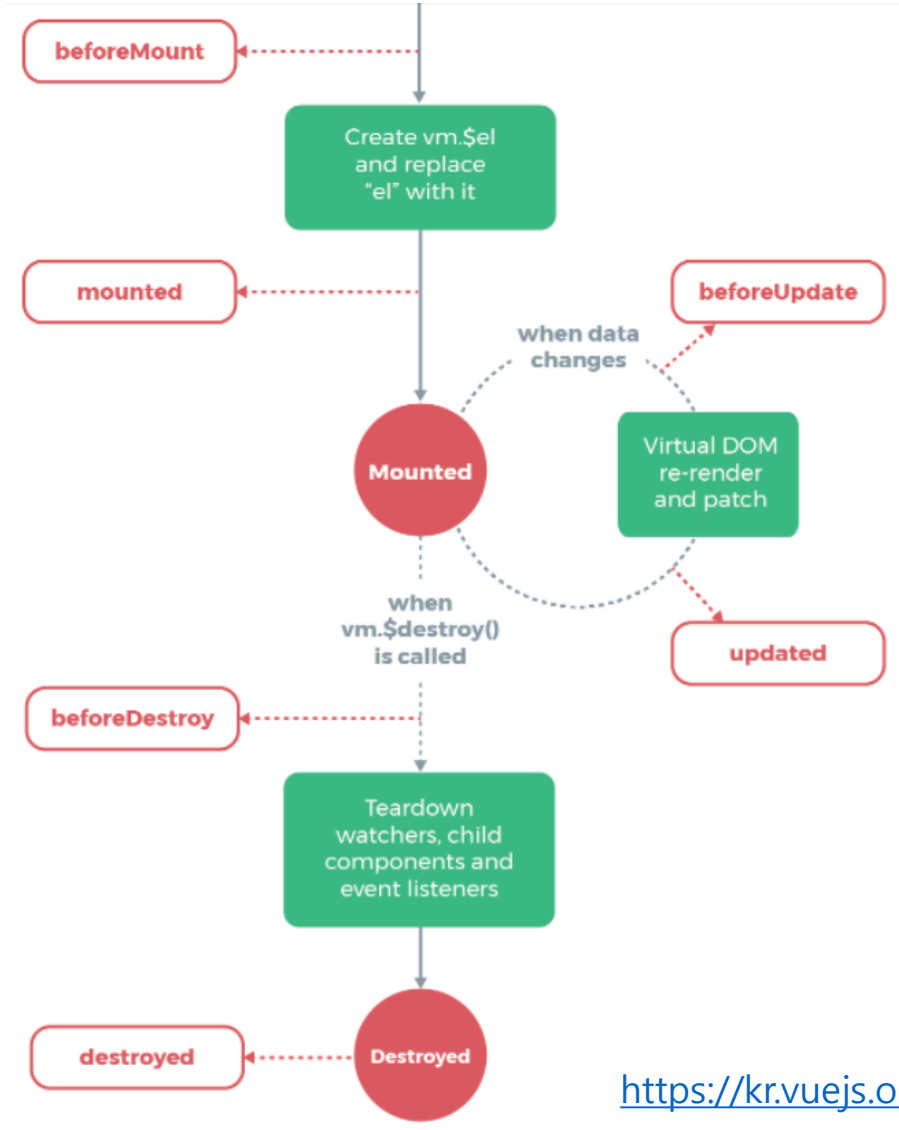
- 각 Vue 인스턴스는 생성될 때 일련의 초기화 단계를 걸친다.
  - 데이터 관찰 설정이 필요한 경우
  - 템플릿을 컴파일 하는 경우
  - 인스턴스를 DOM에 마운트 하는 경우
  - 데이터가 변경되어 DOM을 업데이트 하는 경우 등.
- 그 과정에서 사용자 정의 로직을 실행할 수 있는 Life Cycle Hooks도 호출
- 크게 4개의 파트 (생성, 부착, 갱신, 소멸)로 나뉜다.
- 모든 Life Cycle Hooks은 자동으로 this 컨텍스트를 인스턴스에 바인딩하므로 데이터, 계산된 속성 및 메소드에 접근할 수 있다. (화살표 함수를 사용해 라이프 사이클 메소드를 정의하면 X)

Vue Life Cycle

Confidential

Instance Life Cycle Hooks





https://kr.vuejs.org/v2/guide/instance.html#lifecycle

Vue Life Cycle

Confidential

Instance Life Cycle Hooks

Life Cycle Hooks	설명
beforeCreate	인스턴스가 방금 초기화 된 후 데이터 관찰 및 이벤트 / 감시자 설정 전에 동기적으로 호출
created	인스턴스가 작성된 후 동기적으로 호출. 데이터 처리, 계산된 속성, 메서드, 감시/이벤트 콜백 등과 같은 옵션 처리 완료 마운트가 시작되지 않았으므로 DOM 요소 접근 X
beforeMount	마운트가 시작되기 전에 호출
mounted	지정된 엘리먼트에 Vue 인스턴스 데이터가 마운트 된 후에 호출
beforeUpdated	데이터 변경될 때 virtual DOM이 렌더링, 패치 되기 전에 호출
updated	Vue에서 관리되는 데이터가 변경되어 DOM이 업데이트 된 상태
beforeDestroy	Vue 인스턴스가 제거되기 전에 호출
destroyed	Vue 인스턴스가 제거된 후에 호출

https://kr.vuejs.org/v2/api/index.html#옵션라이프사이클훅

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

Vue Event Handling

Vue Event Handling

- **Vue Event**
  - Listening to Event
  - Method Event Handlers
  - Methods in Inline Handlers
  - Event Modifiers
  - Key Modifiers

Vue Event Handling

Vue Listening to Event (이벤트 청취)

- v-on 디렉티브를 사용하여 DOM 이벤트를 듣고 트리거 될 때 JavaScript를 실행 할 수 있음.

```
<div id="app">
    <button v-on:click="counter += 1">클릭</button>
    <p>위 버튼을 클릭한 횟수는 {{counter}} 번 입니다.</p>
</div>
<script>
    new Vue({
        el: '#app',
        data: {
            counter: 0
        }
    });
</script>
```

Vue Event Handling

Vue Method Event Handlers

- 많은 이벤트 핸들러의 로직은 복잡하여 v-on 속성 값으로 작성하기는 간단하지 않음.
- 때문에 v-on이 호출하고자 하는 method의 이름을 받는 이유

```
<div id="app">
    <button v-on:click="greet">Greet</button>
</div>
<script>
    new Vue({
        el: '#app',
        data: {
            name: 'Vue.js'
        },
        methods: {
            greet: function (event) {
                alert(`Hello ${this.name} !`)
                console.dir(event)
            }
        }
    })
</script>
```

Vue Event Handling

Vue Method in Inline Handlers

- 메소드 이름을 직접 바인딩 하는 대신 인라인 JavaScript 구문에 메소드를 사용할 수도 있음.
- 때로 인라인 명령문 핸들러에서 원본 DOM 이벤트에 엑세스 해야할 수도 있음.
  $event 변수를 사용해 메소드에 전달할 수 있음.

```
<div id="example-3">
    <button v-on:click="say('hi')">Say hi</button>
    <button v-on:click="say('what')">Say what</button>
</div>

new Vue({
    el: '#example-3',
    methods: {
        say: function (message) {
            alert(message)
        }
    }
})
```

Vue Event Handling

Vue Event Modifiers

- 이벤트 핸들러 내부에서 event.preventDefault() 등을 호출하는 것은 보편적인 일이다.
- 메소드 내에서 쉽게 작업을 할 수 있지만, methods는 DOM의 이벤트를 처리하는 것 보다 data 처리를 위한 로직만 작업하는 것이 좋음.
- v-on 이벤트에 이벤트 수식어를 제공 (.으로 표시된 접미사)
- 체이닝 가능
- .stop : event.stopPropagation() 호출 - 클릭 이벤트 전파 X
- .prevent : event.preventDefault() 호출 - 제출 이벤트가 페이지를 다시 로드 X
- .capture : 캡처 모드에서 이벤트 리스너를 추가함.
- .self : 이벤트가 이 엘리먼트에서 전달된 경우에만 처리 됨.
- .once : 단 한번만 처리.
- .passive : DOM 이벤트를 { passive : true } 와 연결함.

https://kr.vuejs.org/v2/api/#v-on

Vue Key Modifier

- Vue는 키 이벤트를 수신할 때 v-on에 대한 키 수식어를 추가할 수 있음.

```
<input v-on:keyup.13="submit">
```

- .enter (.13)
- .tab
- .delete (“Delete” 와 “Backspace” 키 모두 캡처)
- .esc
- .space
- .up / .down / .left / .right
- 전역 config.keyCodes 객체를 통해 사용자 지정 키 수식어 별칭 지정 가능

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

Vue Bindings

Vue Bindings

ref, $refs

- $refs : ref 속성이 등록된 자식 컴포넌트와 DOM 엘리먼트 객체
  템플릿이나 계산 된 속성에서 사용X
- ref : 엘리먼트 또는 자식 컴포넌트에 대한 참조를 등록하는데 사용.

```
<div id="app">
      아이디 : <input type="text"  v-model="id"  ref="id">
</div>
<script>
      new Vue({
            ..
            this.$refs.id
      });
</script>
```

Vue Bindings

Class Bindings

- 데이터 바인딩은 엘리먼트의 클래스 목록과 인라인 스타일을 조작하기 위해 일반적으로 사용됨.
- v-bind 를 사용하여 처리 가능.
- 문자열 이외에 객체 또는 배열을 이용할 수 있음.

```
<div v-bind:class="myclasses"></div>
<script>
    new Vue({
            …
            data: {
                myclasses: {  'a': true, 'b': true, 'c': false }
            }
    });
</script>
```

```
<div v-bind:class="{ 'a': apply }"></div>
<script>
    new Vue({
            …
            data: {
                apply : true
            }
    });
</script>
```

https://kr.vuejs.org/v2/guide/class-and-style.html

Vue Bindings

Form Input Bindings

- v-model 디렉티브를 사용하여 form input과 textarea 엘리먼트에 양방향 데이터 바인딩을 생성 할 수 있음.
- text 와 textarea : value, input 이벤트 사용
- checkbox, radio : checked, change 이벤트 사용
- select : value, change 이벤트 사용
- v-model은 모든 form 엘리먼트의 초기 value와 checked 그리고 selected 속성을 무시함.

https://kr.vuejs.org/v2/guide/forms.html

Vue Bindings

Form : text, textarea

- text

```
<input v-model="message" placeholder="여기를 수정해보세요">
<p>메시지: {{ message }}</p>
```

- textarea

```
<span>여러 줄을 가지는 메시지:</span>
<p style="white-space: pre-line">{{ message }}</p>
<br>
<textarea v-model="message" placeholder="여러줄을 입력해보세요."></textarea>
```

- 텍스트 영역의 보간 (<textarea> {{ message }} </textarea>)은 작동X
  v-model을 사용.

https://kr.vuejs.org/v2/guide/forms.html

Vue Bindings

Form : checkbox

- 하나의 체크박스일 경우 boolean 값을 표현

```
<input type="checkbox" id="checkbox" v-model="checked">
<label for="checkbox">{{ checked }}</label>
```

- 여러 개의 체크박스는 같은 배열을 바인딩 할 수 있음.

```
<div id='example-3’>
  <input type="checkbox" id="jack" value="Jack" v-model="checkedNames">
  <label for="jack">Jack</label>
  <input type="checkbox" id="john" value="John" v-model="checkedNames">
  <label for="john">John</label>
  <input type="checkbox" id="mike" value="Mike" v-model="checkedNames">
  <label for="mike">Mike</label>
  <br>
  <span>체크한 이름: {{ checkedNames }}</span>
</div>
```

```
new Vue({
  el: '#example-3’,
  data: {
    checkedNames: []
  }
})
```

https://kr.vuejs.org/v2/guide/forms.html

Form : radio

- 라디오 박스일 경우 선택된 항목의 value 속성의 값을 관리

```
<input type="radio" id="one" value="One" v-model="picked">
<label for="one">One</label>
<br>
<input type="radio" id="two" value="Two" v-model="picked">
<label for="two">Two</label>
<br>
<span>선택: {{ picked }}</span>
```

https://kr.vuejs.org/v2/guide/forms.html

Vue Bindings

Form : select (단일)

- select box일 경우 선택된 겨항목의 value 속성의 값을 관리

```
<select v-model="selected">
  <option disabled value="">Please select one</option>
  <option>A</option>
  <option>B</option>
  <option>C</option>
</select>
<span>선택함: {{ selected }}</span>
```

```
new Vue({
  el: '...',
  data: {
    selected: ‘’
  }
})
```

- v-model 표현식의 초기 값이 어떤 옵션에도 없으면 <select> 엘리먼트는 ‘선택없음’ 상태로 렌더링

https://kr.vuejs.org/v2/guide/forms.html

Form : select (다중)

- select box일 경우 선택된 겨항목의 value 속성의 값을 관리

```
<select v-model="selected“ multiple>
  <option>A</option>
  <option>B</option>
  <option>C</option>
</select>
<br>
<span>선택함: {{ selected }}</span>
```

```
new Vue({
  el: '...’,
  data: {
    selected: ‘’
  }
})
```

- v-for를 이용한 동적 option 렌더링 가능

https://kr.vuejs.org/v2/guide/forms.html

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

다음 방송에서 만나요!

삼성 청년 SW 아카데미