SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

삼성 청년 SW 아카데미

Vue.js

SAMSUNG SW ACADEMY FOR YOUTH

FrontEnd Framework

Vue Directive

- Vue Template Syntax
- Vue Instance option

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

Vue Template Syntax

Vue Template Syntax

Confidential

Template Syntax

- Vue.js는 렌더링 된 DOM을 기본 Vue 인스턴스의 데이터에 선언적으로 바인딩 할 수 있는 HTML 기반 템플릿 구문 사용

- 보간법 (Interpolation)
- 디렉티브 (Directive)

Vue Template Syntax

Confidential

Interpolation (보간법) - Text (문자열)

- 데이터 바인딩의 가장 기본 형태 “Mustache” 구문 (이중 중괄호)을 사용한 텍스트 보간.
- `<span>메시지: {{ msg }}</span>`
- Mustache 태그는 데이터 객체의 msg 속성 값으로 대체
  (해당 값이 변경되면 갱신)

- v-once 디렉티브를 사용하여 데이터 변경 시 업데이트 되지 않는 일회성 보간을 수행
- `<span v-once>다시는 변경하지 않습니다: {{ msg }}</span>`

https://kr.vuejs.org/v2/guide/syntax.html#문자열

Confidential

Interpolation (보간법) - Raw HTML (원시 HTML)

- 이중 중괄호는 HTML이 아닌 일반 텍스트로 해석
  실제 HTML을 출력하기 위해서는 v-html 디렉티브 사용
-
```
<p>Using mustaches: {{ rawHtml }}</p>
<p>Using v-html directive: <span v-html="rawHtml"></span></p>
```

Using mustaches: This should be red.

Using v-html directive: This should be red.

- 웹사이트에서 임의의 HTML을 동적으로 렌더링하려면 XSS 취약점으로 쉽게 이어질 수 있으므로 매우 위험할 수도 있음. 신뢰할 수 있는 콘텐츠에서만 HTML 보간을 사용할 것

Confidential

Interpolation (보간법) - JavaScript 표현식 사용

- Vue.js는 모든 데이터 바인딩 내에서 JavaScript 표현식의 모든 기능을 지원.
- ```
  {{ number + 1 }}
  {{ ok ? 'YES' : 'NO' }}
  {{ message.split('').reverse().join('') }}

  ```

- 한가지 제한사항은 각 바인딩에 하나의 단일표현식 만 포함될 수 있으므로 아래처럼 작성X
- ```
  <!-- 아래는 구문입니다, 표현식이 아닙니다. -->
  {{ var a = 1 }}
  <!-- 조건문은 작동하지 않습니다. 삼항 연산자를 사용해야 합니다. -->
  {{ if (ok) { return message } }}
  ```

Vue Template Syntax

Confidential

Directive (디렉티브)

- v- 접두사가 있는 특수 속성
- 속성값은 단일 JavaScript 표현식이 됨. (v-for 예외)
- 역할은 표현식의 값이 변경될 때 사이드 이펙트를 반응적으로 DOM에 적용

- v-text
- v-html
- v-show
- v-if
- v-else
- v-else-if
- v-for
- v-on
- v-bind
- v-model
- v-slot
- v-pre
- v-cloak
- v-once

Vue Template Syntax

Confidential

Directive (디렉티브) : v-text

- 엘리먼트의 textContent를 업데이트.
- 일부를 갱신해야 한다면 {{ }} 를 사용해야함.

-
```
<span v-text="msg"></span>
<!-- 같습니다 -->
<span>{{ msg }}</span>
```

Confidential

Directive (디렉티브) : v-bind

- HTML 요소의 속성에 Vue 상태 데이터를 값으로 할당
- Object 형태로 사용하면 value가 true인 key가 class 바인딩 값으로 할당.
- 약어 제공 :
  v-bind:herf === :href

- `<a v-bind:href="url"> ... </a>`

Confidential

Directive (디렉티브) : v-model

- HTML form 요소의 input 엘리먼트 또는 컴포넌트에 양방향 바인딩 처리
- 수식어
  - .lazy : input 대신 change 이벤트 이후에 동기화
  - .number : 문자열을 숫자로 변경
  - .trim : 입력에 대한 trim 진행
- form 엘리먼트 초기 ‘value’와 ‘checked’, ‘selected’속성을 무시함.

Confidential

Directive (디렉티브) : v-show

- 조건에 따라 엘리먼트를 화면에 표시
- 항상 렌더링되고 DOM에 남아있음.
- 단순히 엘리먼트에 display CSS 속성을 토글하는것.
- 조건이 바뀌면 트랜지션 호출
- `<h1 v-show="ok">Hello!</h1>`

Confidential

Directive (디렉티브) : v-if, v-else-if, v-else

- 표현식 값의 참 거짓을 기반으로 엘리먼트를 조건부 렌더링 함.
- 엘리먼트 및 포함된 디렉티브/컴포넌트 는 토글 하는 동안 삭제되고 다시 작성됨.
- <template> 엘리먼트를 이용하여 v-if 사용가능, 최종 결과에는 <template>엘리먼트는 포함 X

```
<span v-if="age < 10">무료</span>
<span v-else-if="age < 20">7000원</span>
<span v-else-if="age < 65">10000원</span>
<span v-else>3000원</span>
```

Confidential

Directive (디렉티브) : v-for

- 원본 데이터를 기반으로 엘리먼트 또는 템플릿 블록을 여러 번 렌더링.
- 디렉티브의 값은 반복되는 현재 엘리먼트에 대한 별칭을 제공하기 위해 alias in expression 을 사용
-
```
<div v-for="item in items">
    {{ item.text }}
</div>
```
- 또는, 인덱스(아니면 객체의 경우 키)의 별칭 사용 가능
-
```
<div v-for="(item, index) in items"></div>
<div v-for="(val, key) in object"></div>
<div v-for="(val, name, index) in object"></div>
```
- v-for의 기본 동작은 엘리먼트를 이동하지 않고 그 자리에서 패치 시도, 강제로 엘리먼트의 순서를 바꾸려면 특수 속성 key 설정

Confidential

Directive (디렉티브) : v-for

- v-for의 기본 동작은 엘리먼트를 이동하지 않고 그 자리에서 패치 시도, 강제로 엘리먼트의 순서를 바꾸려면 특수 속성 key 설정
-
```
<div v-for="item in items" :key="item.id">
    {{ item.text }}
</div>
```
- v-if와 함께 사용하는 경우, v-for는 v-if보다 높은 우선순위를 갖는다.
  따라서 되도록이면 같이 작성하는 것은 피하기

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

Vue Template Syntax

Confidential

Directive (디렉티브) : v-on

- 엘리먼트에 이벤트 리스너를 연결
- 이벤트 유형은 전달인자로 표시
- 약어 제공 @
  v-on:click === @click

- ```

      <button v-on:click="counter += 1">Add 1</button>
      위 버튼을 클릭한 횟수는 {{ counter }} 번 입니다.

  ```

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

Vue Template Syntax

Confidential

Directive (디렉티브) : v-cloak

- Vue Instance가 준비될 때까지 Mustache 바인딩을 숨기는 데 사용.
- [v-clock] { display: none } 과 같은 CSS 규칙과 함께 사용.
- Vue Instance가 준비되면 v-cloak 는 제거됨.

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

Vue Instance Option

Vue Instance Option

Confidential

Vue Method

- Vue Instance는 생성 관련된 데이터(data) 및 메서드의 정의 가능
- method 안에 data를 this.데이터이름 으로 접근

```
<div id="app">
    <button @click="greet">인사</button>
</div>
<script>
    new Vue({
        el: '#app',
        data() {
            return { name: "Yang" }
        },
        methods: {
            greet() {
                //메소드 안에서 사용하는 this 는 Vue 인스턴스를 가리킨다.
                alert(`Hello ${this.name} !`)
            }
        }
    });
</script>
```

Vue Instance Option

Confidential

Vue Filters

- Vue는 텍스트 형식화를 적용할 수 있는 필터를 지원함.
- filter를 이용하여 표현식에 새로운 결과 형식을 적용
- {{ Mustache }} 구문(이중 중괄호) 또는 v-bind 속성에서 사용이 가능
- 자바스크립트 표현식 마지막에 "¦" 심볼과 함께 추가 되어야 함.
- 필터는 체이닝이 가능

```
<!-- 중괄호 보간법 -->
{{ message ¦ capitalize }}

<!-- v-bind 표현 -->
<div v-bind:id="rawId ¦ formatId"></div>
```

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

Confidential

Vue filters

- 전역 필터

```
Vue.filter(
      'count', (val) => {
            if (val.length == 0) {
                return;
            }
            return `${val} : ${val.length}자`;
      }
);
```

- 지역 필터

```
new Vue({
     el: '#app',
     filters: {
          count(val) {
               return `${val} : ${val.length}자`;
          }
     }
});
```

Vue Instance Option

Confidential

Vue computed

- 특정 데이터의 변경사항을 실시간으로 처리할 수 있음.
- 캐싱을 이용하여 데이터의 변경이 없을 경우 캐싱된 데이터를 반환
- Setter와 Getter를 직접 지정할 수 있음.
- 작성은 method 형태로 정의하지만 Vue에서 프록시 처리하여 property처럼 사용.
- 화살표 함수를 사용하면 안됨.

```
<div id="app">
    <p>원본 메시지: "{{ message }}"</p>
    <p> {{ reversedMessage }}"</p>
</div>
```

```
var vm = new Vue({
   el: '#app',
   data: {
      message: '안녕하세요'
   },
   computed: {
      reversedMessage: function () {
         return this.message.split('').reverse().join('')
      }
    }
});
```

Vue Instance Option

Confidential

Vue computed & methods

- computed 속성 대신 methods에 함수 정의하여 사용가능 .(최종 결과는 같음)
- computed 속성은 종속 대상을 계산하여 저장해 놓는다.(캐싱)
  즉, 종속된 대상이 변경되지 않는 한 computed에 작성된 함수를 여러 번 호출해도 계산을 다시 하지 않고, 계산되어 있는 결과를 반환
- methods를 호출하면 렌더링을 다시 할 때마다 항상 함수를 실행

Vue Instance Option

Confidential

Vue watch

- Vue Instance의 특정 property가 변경될 때 실행할 callback 함수 설정
- 데이터를 감시
- 화살표 함수 사용 X

```
var vm = new Vue({
   el: '#app',
   data: {
      message: ‘Hello‘,
      reversedMessage: ‘’
   },
   watch: {
      message: function (newVal, oldVal) {
         this.reversedMessage = newVal.split(‘’).reverse().join(‘’);
      }
    }
});
```

```
<div id="app">
    <p>원본 메시지: "{{ message }}"</p>
    <p> {{ reversedMessage }}"</p>
    <input type=“text” v-model=“message”>
</div>
```

Vue Instance Option

Confidential

Vue computed & watch

- computed
  특정 데이터를 직접적으로 사용/가공하여 다른 값으로 만들 때 사용.
  계산해야 하는 목표 데이터를 정의하는 방식으로 SW 공학에서 이야기하는 '선언형 프로그래밍' 방식
- watch
  특정 데이터의 변화 상황에 맞추어 다른 data 등이 바뀌어야 할 때 주로 사용.
  감시할 데이터를 지정하고 그 데이터가 바뀌면 이런 함수를 실행하라 라는 방식으로 SW 공학에서 이야기하는 '명령형 프로그래밍' 방식

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

다음 방송에서 만나요!

삼성 청년 SW 아카데미