Vue.js란?

* JavaScript framework
  + Javascript
    - 브라우저의 구현에 사용되는 언어
    - 이미 실행 중인 page를 조작해 content가 좀 더 rich하게 될 수 있게 함
  + Framework
    - Third-party library
    - Set-of-rule이 있고 이걸 따른 방식으로 구현이 되어야 함
* Frontend 부분을 interactive & reactive하게 만드는 것을 좀 더 쉽게 할 수 있게 해줌
  + Reactive
    - Update screen dynamically
* Mobile app을 생각해보면 one-app인 것같이 느껴지게 new page가 load되는 시간이 매우 짧거나 없음 = very responsive
  + Request를 매번 보내는게 아니라 DOM을 수정하는 것으로 page에 변화를 생기도록 함

Different ways of utilizing vue

* Control parts of HTML page
  + Page를 구성하는 특정 element를 control, called widgets
* Entire frontend of web
  + Single-Page-Application approach라고 함
  + Server에 request를 한 번 보내서 page를 받은 다음, 그 이후의 작동은 vue가 담당

Vue alternative

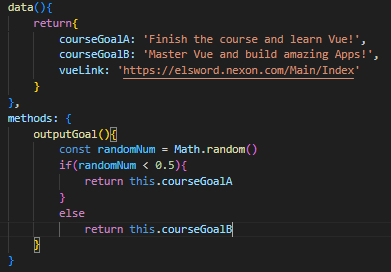
* Vue
  + Component based
* React
  + Lean & focused component-based UI library
* Angular
  + Component based
  + Use TypeScript

작성순서

* Vue.createApp()을 통해 Vue를 이용해 app을 생성하겠다고 명시
  + State가 data를 통해 관리되며 data는 function()의 형태로 사용되어야 함.
  + 그리고 data는 return을 통해 어떤 값이 반환되는지 확인되어야 함

Vue methods

* CreateApp 부분에서 data와 같이 정의, app에서 사용할 함수 같은 것들의 정의가 진행됨
* Methods : { … } 의 형식으로 작성됨
* … 부분에 사용할 함수를 선언 및 작성하게 됨
* Html 파일에서 {{ … }} 형식으로 methods에 작성되어 있는 함수 호출이 가능



* 이런 식으로 선언되어 있을 때, methods에서 this.변수명을 사용하는 것으로 data에 정의되어 있는 변수값을 사용할 수 있음
* 이게 가능한 이유는 Vue에서 위에서 정의한 data에서 return하는 data들을 모두 모아서 global Vue instance로 만들게 되는데 global object이기 때문에 어디서든 접근이 가능하게 됨.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

* 이런 식으로 vue component와 html component 값을 bind해서 사용할 수 있음. 이 경우, reset input 버튼을 누르게 되면 input의 value에 해당되는 vue에서 정의된 data 중 name 변수의 값이 ‘’으로 바뀌게 됨

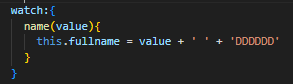
Methods

* method에서 정의된 함수는 hard-code된 부분이 아니기 때문에 vue component에서 변화가 생길 때마다 재로딩이 되기 때문에 특정 경우에서 효율적이지 못함
  + 화면에서 변화가 많이 일어나는 경우 등
* Event binding OR data binding에 사용
  + Data binding – method가 component가 re-render될 때마다 실행됨
  + Event나 data가 지속적으로 revaluate되어야 할 때 사용

Computed

* Method와 마찬가지로 method (함수)를 정의하게 됨. 하지만 실행은 methods에서 정의된 method와 다른 방식으로 진행됨
* () 없이 computed에 정의되어 있는 method의 이름을 호출하는 방식으로 사용
* Data binding에 사용
* 다른 data에 의존하는 data에 사용

Watch

* Methods에서와 마찬가지로 method를 정의
* Data에 정의된 variable의 이름과 같은 이름을 갖는 method를 선언 및 정의
* Variable의 property가 바뀔 때마다 정의된 logic을 실행하게 됨
* 직전의 variable value를 parameter로 가지게 됨
  + 
* watch해야하는 variable이 여러 개가 되는 경우에는 computed를 사용하는게 효율적일 수 있음
  + watch의 경우 한 번에 하나만 watch를 하기 때문에 각 variable에 대해서 method를 선언해줘야 함
* not used directly in template
* changed data에 반응하게 코드를 실행하는 것을 허용
* non-data update에 사용됨

Vue dynamic styling

v-if & v-else

* v-if와 v-else문은 if가 쓰여진 바로 다음 줄에 else가 나와야 else 부분이 유효

v-show

* v-if & v-else와 같이 특정 조건에 어떤 내용을 보여줄지 결정할 수 있는 문법
* v-if & v-else와 같이 사용은 안됨. 여러 조건이 있다면 v-show를 여러 번 사용해야 함
* v-if & v-else와 다른 점은 if & else는 element를 조건이 바뀔 때 DOM에 추가/삭제하는 반면 v-show는 CSS로 element를 show/hide 시킴
  + 따라서, if & else를 주로 사용하되 toggle이나 계속 show/hide되는 element라면 v-show를 사용하는 것을 고려

v-for

* V-for를 사용하는 것으로 반복문처럼 data 접근이 가능
* V-for=” 조건 “ 의 형태로 사용되는데 조건은 “variable name” in “array” 의 형태로도 사용 가능
  + 조건에 (변수 이름, index)로 몇 번째 index인지 까지 출력할 수 있음

Vue에서 element의 삭제

* For 반복문을 통해 생성한 element나 다른 element들의 경우, 삭제를 하게 되면 해당 위치의 element가 삭제되는 것이 아닌 뒤에 있는걸 지우고 거기 있던 내용을 앞의 element에 덮어 씌우는 방식임
* V-key를 사용하는 것으로 해결할 수 있음
  + Default html attribute가 아니라 vue에서 지원하는 attribute임
* Ref
  + Default html attribute가 아니라 vue에서 지원하는 attribute임
  + This.$refs.{{접근하고 싶은 ref의 변수명}}의 형식으로 ref에 정의한 이름에 접근 가능

Vue에서 파일 작성 요령

* Html 파일에는 최대한 적은 정보만 있도록 작성

Vue가 DOM을 update하는 법

* Vue instance
  + Data, computed properties, methods
  + Data & computed properties는 change될 수 있음
* Browser DOM
  + DOM에 vue로 생성한 template이 render 됨
  + Update가 반영되지만 entire page를 re-render하지는 않음 (바뀐 부분만 부분적으로 re-render)
* Virtual DOM
  + Js-based, exist only in memory
  + Data가 바뀔 때마다 old virtual DOM과 비교해서 다른 부분이 있을 경우 그 부분만 부분적으로 re-render를 진행

Vue instance lifecycle

텍스트, 스크린샷, 직사각형, 폰트이(가) 표시된 사진

자동 생성된 설명

Vue에서 component 사용법

* createApp을 하면서 .mount() 전에 .component()를 추가
* component는 mini-app이라고 생각하면 됨