실습 환경 구성

node.js 14.17.0

vs code 플러그인

- Vetur

- Night Owl

- Material Icon Theme

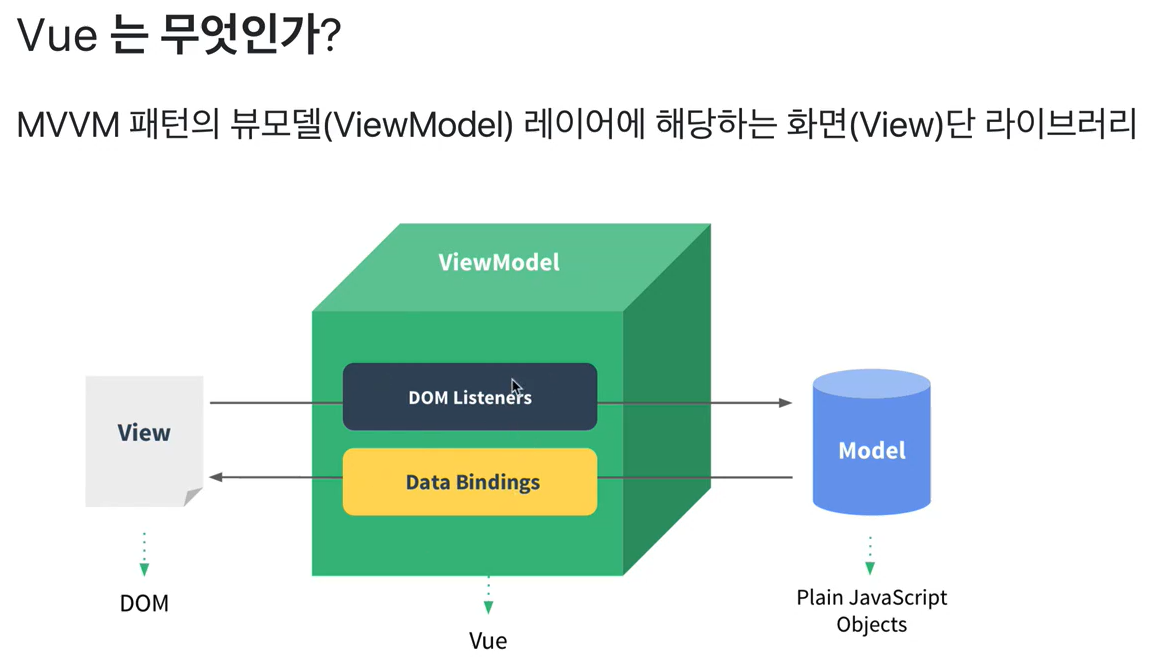
- Live Server

- ESLint

- Prettier

- Auto Close Tag

- Atom Keymap



Objcet.defineProperty()를 이용한 div 내용 동적 변경. (reactivity)

// *객체 동작 재정의*

        // *Object.defineProperty(대상 객체, 객체의 속성, {*

        // *정의할 내용*

        // *});*

*Object.defineProperty*(viewModel, 'str', {

            // *게터*

*get*: function(){

*console.log*('getter');

            },

            // *세터*

*set*: function(newValue){

*console.log*('setter', newValue);

                // *객체의 속성이 변경될때마다, div도 수정.*

*div.*innerHTML = newValue;

            }

        });

즉시실행함수

// *즉시실행함수 자바스크립트 문법.*

      (function () {

          ...

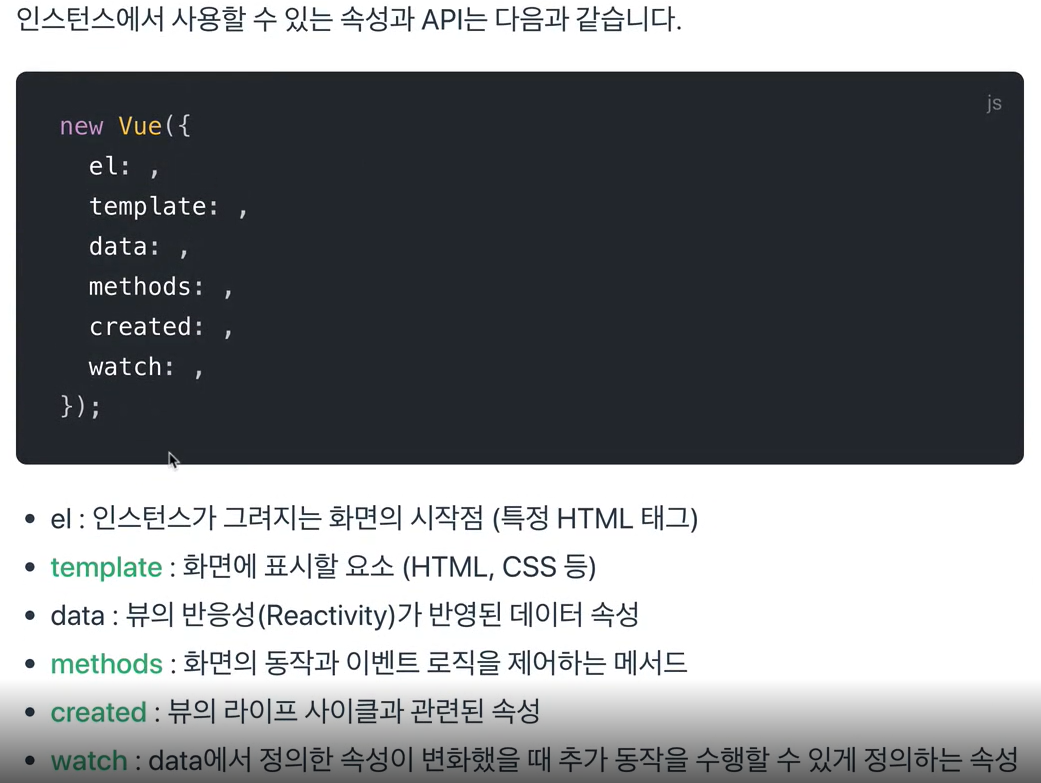
      })();

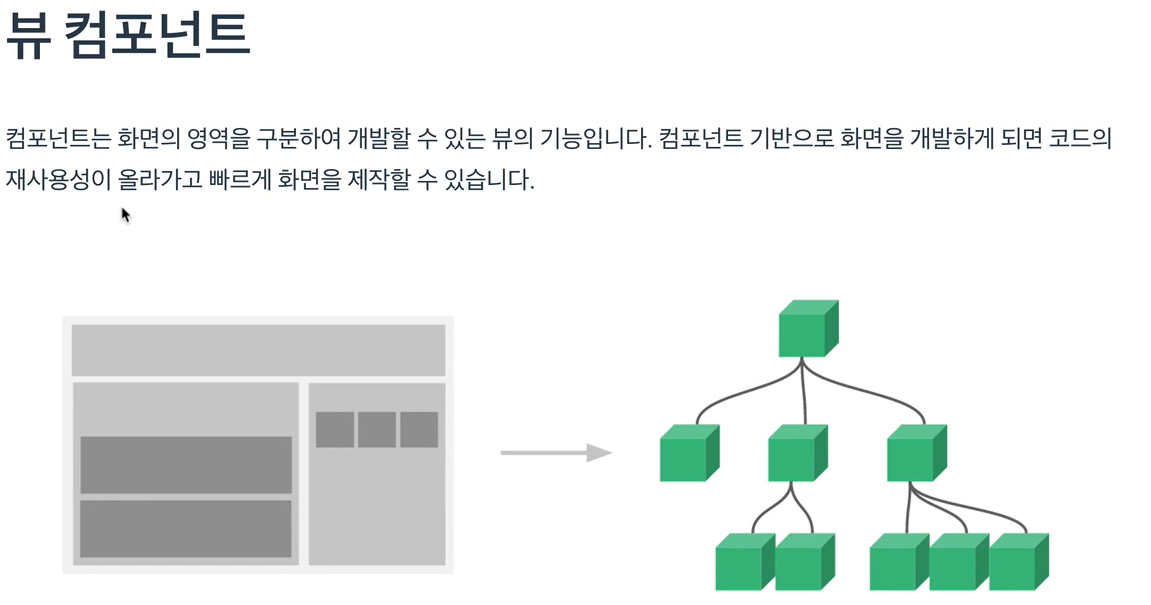
예제의 init과 render와 같은 함수가 애플리케이션에 노출되지 않도록 또 다른 스코프에 넣어 줌.

변수의 유효범위를 관리.

뷰 인스턴스

new *Vue*();





인스턴스를 처음 등록하면 root가 됨.

전역 컴포넌트 등록

*Vue.component*('app-header',{

            template: '<h1>Header</h1>'

        })

지역 컴포넌트 등록

new *Vue*({

            el: '#app',

            components: {

                'app-footer': {

                    template: '<footer>value</footer>'

                }

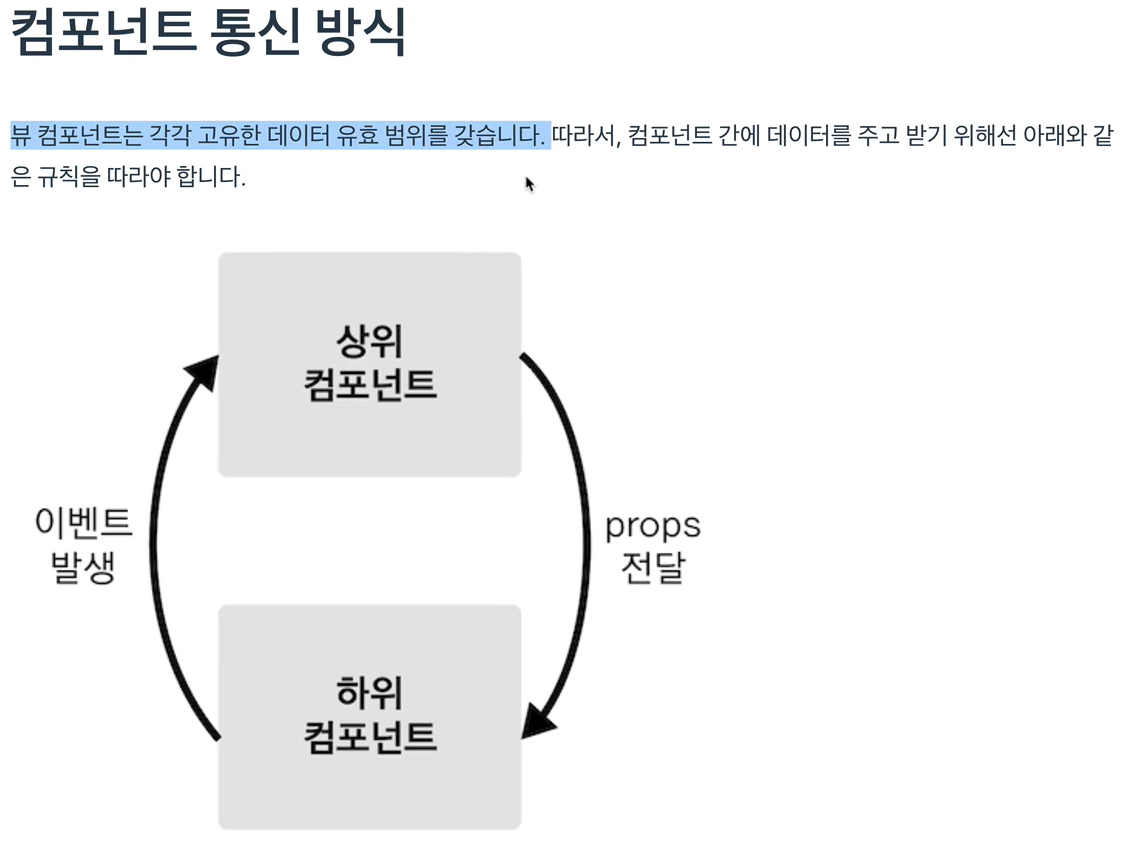
            }

        });

지역 컴포넌트는, components 속성으로 아래에 딸린 컴포넌트거 어떤게 있는지 조회 가능.

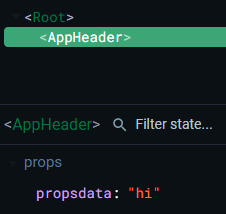
전역 컴포넌트는 플러그인처럼 전역적으로 사용해야 하는 것에만 사용.

컴포넌트 통신



props 전달 방법

<컴포년트명 *v-bind:props속성이름* = "상위 컴포넌트의 데이터 이름></ 컴포년트명>

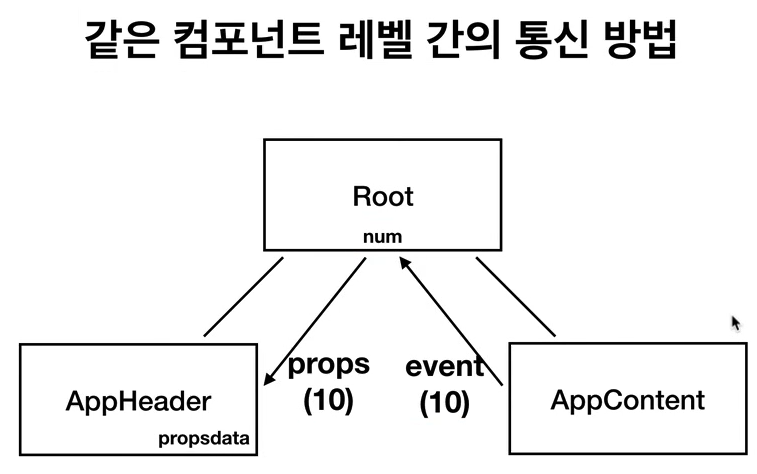


하위 컴포넌트에서 상위 컴포넌트로 event 전달

1. 컴포넌트의 methods 속성에 이벤트 메서드 구현
   * 메서드 내부에서 this.$emit(‘전달할 이벤트명’, 기타파라미터) 로 상위 컴포넌트에 이벤트 전달.
2. 위에서 작성한 메서드를 v-on:event명=”메서드명” 으로 커스텀 이벤트 발생.

같은 컴포넌트 레벨 간의 통신

상위 컴포넌트의 data를 공유해서 event, props 하는 방식으로 통신함.



뷰 라우터

뷰 라우터는 뷰 라이브러리를 이용해여 싱글페이지 애플리케이션을 구현할 때 사용하는 라이브러리 이다.

<!-- *script 순서를 vue, vue-router 순으로 해줘야함.* -->

<script *src*="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>

<script *src*="https://unpkg.com/vue-router@3.5.3/dist/vue-router.js"></script>

뷰 라우터 등록

        // *라우터 인스턴스 생성*

        varrouter=new *VueRouter*({

        // *라우터 옵션*

})

        // *인스턴스에 라우터 인스턴스를 등록*

        new *Vue*({

        router: router

        })

라우터 설정

varrouter=new *VueRouter*({

            // *페이지의 라우팅 정보*

*routes*:[

{

                    // *페이지의 url 이름*

*path*:'/login',

                    // *해당 url에서 표시될 컴포넌트*

*component*:LoginComponent,

*name*:'login'

},

{

*path*:'/main',

*component*:MainComponent

}

]

});

router-link

<router-link *to*="이동할 url">Login</router-link>

<router-link *to*="/main">Main</router-link>

이렇게 설정해 놓으면 a태그로 변환되어서 출력.