

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>컴퍼넌트_지역컴포넌트_템플릿_props_$emit_반응형</title>

<link rel="stylesheet" href="/css/reset.css">
<link rel="stylesheet" href="/css/style.css">

<!-- 오픈소스 -->
<script src="/js/lib/jquery.min.js"></script>
<script src="/vue/lib/vue.js"></script>

</head>
<body>
<div id="wrap">

<!-- 헤더영역 시작 -->

<div id="header">
<div class="wrap">
  <div class="gap">
    <div class="container">

      <!-- logo template -->
      <logo-component v-for='item in imglogo' :propslogo="item"></logo-component>

      <!-- Navigation template -->
      <nav id="nav">
        <ul>
          <nav-component v-for='item in gnb' v-bind:propsnav='item'></nav-component>
        </ul>
      </nav>

    </div>
  </div>
</div>
```

```
</div>
</div>
```

```
<script>
```

```
var logoComponent = {
  props:['propslogo'],
  template:`\
    <div class="logo">\
      <a href="javascript:" class="logo-btn">\
        \
      </a>\
    </div>\
  `
}

var navComponent = {
  props:['propsnav'],
  template:`\
    <li>\
      <a href="javascript:" class="main-btn">{{ propsnav.menu }}</a>\
    </li>\
  `
}

var header = new Vue({
  el:'#header',
  data:{
    imglogo:[
      {img: './img/logo_image.png'}
    ],
    gnb:[
      {menu:'ABOUT'},
      {menu:'PORTFOLIO'},
      {menu:'SERVICES'},
      {menu:'TESTIMONIAL'},
      {menu:'BLOG'},
      {menu:'CONTACT'}
    ]
  }
})
```

```
    },
    components:{
      'logo-component': logoComponent,
      'nav-component': navComponent
    }
  }
};
```

```
</script>
```

```
<!-- 헤더영역 끝 -->
```

```
<!-- 메인영역 시작 -->
```

```
<div id="main">
<section1-component v-on:emitevent='updateFiledata' v-bind:propssec1='filedata'></section1-component>
</div>
```

```
<script>
```

```
var section1Component = {
  props:['propssec1'],
  template:`
    <section id="section1">\
      <div class="wrap">\
        <div class="gap">\
          <div class="container">\
            <h4>WE WORK HARD, WE PLAY HARD</h4>\
            <h2>\
              SALIENT IS DIGITAL<br>\
              BRILLIANCE\
            </h2>\
            <h3>CREATIVE SOLUTIONS TO IMPROVE YOUR BUSINESS</h3>\
            <p>\
```

```

        <span><a href="javascript:" v-on:click="popup(\./popup/main\' + propssec1 + \'.html\', \'Window Popup\' + propssec1 + \' \')">OUR
SERVICES</a></span>\
    </p>\
</div>\
</div>\
</div>\
</section>\
,
,
methods:{
  popup:function(y, z){
    this.$emit('emitevent'); //에밋 메서드 실행 이벤트 생성

    var url = y;
    var name = z;
    var option = "width=500, height=600, top=0, left=0";

    // 팝업창
    window.open(url, name, option);

    console.log( y, z );
    //1. 하위 컴포넌트에서 이벤트를 발생시켜서

    //2. 상위 컴포넌트에 이벤트를 전달할 이벤트이름(emitevent)을
    // 생성(에밋 $emit 이라는 메서드로 이벤트를 생성)한다.

    //3-1. 생성된 이벤트이름을 상위 컴포넌트의 데이터속성을 할당하고 활성화한다.
    // 예] v-on:하위 컴포넌트에서 생성된 이벤트 이름='상위 컴포넌트의 데이터변경을 위해서
    // 메서드 안에 메서드 함수(updateFiledata)를 호출한다.'
    // 예] v-on:emitEvent='updateFiledata'

    //3-2 상위 컴포넌트의 메서드속성을 이용 호출할 메서드함수를 만든다.
    // 메서드 함수 변경 내용은
    // 함수 호출 시 100씩 증가하는 알고리즘 구문
    // 예] this.filedata +=100 ;

    //4 상위 컴포넌트에 데이터 속성중 filedata의 값을 100씩 증가(변경)된 값을 프롭스[props]를 통해서
    // 하위 컴포넌트에 값이 전달되게한다.
    //5. 그리고 전달된 데이터 값을 템플릿에서 속성으로 받아서 변경된 파일명을

    //6. 팝업창으로 띄운다.
    // 단 팝업창은 너비 500픽셀, 높이 500픽셀, 탑 0 , 좌측 300픽셀 네비게이션 안나오게 하여 띄운다.

```

```

        // 예] window.open("팝업파일경로/파일명.html", "팝업창이름", "width=500, height=600, top=0, left=300, location=no");
        // 예] window.open("./popup/main100.html", "Window Popup 100", "width=500, height=600, top=0, left=300, location=no");
    }
}
}

```

```

var main = new Vue({
  el:'#main',
  data:{
    filedata:100
  },
  components:{
    'section1-component': section1Component
  },
  methods:{
    updateFiledata: function(){
      if( this.filedata >= 300 ){
        this.filedata = 100;
      }
      else{
        this.filedata +=100 ;
      }
    }
  },
  mounted:function(){

```

```

    //mounted 메서드 : 화면에 부착이 완료된 상태

```

```

    //창너비(window.innerWidth)와 창높이(window.innerHeight)를 반응형에 적용
    //그리고 섹션1(#section1)의 창높이와 창너비를 설정한다.
    //단, 화면크기가 변경 시 즉시 반응하여 적용되도록 한다.
    //addEventListener 이벤트 리스너 사용
    //예] window.addEventListener('resize', 반응형함수호출);

```

```

    //자바스크립트 반응형

```

```

    function javascriptFn(){
      var section1 = document.querySelector('#section1');
      var winH = window.innerHeight; //창높이
    }

```

```
function resizeFn(){
    winH = window.innerHeight; //창높이
    // section1.style = 'height:' + winH + 'px';
    section1.style.height = winH + 'px';
}
resizeFn(); //마운트 완료시 실행 1회
window.addEventListener('resize',resizeFn);
//창 크기 변경 시 즉각 반응하여 함수 호출 실행하는 이벤트 리스너
}
//javascriptFn(); //자바스크립트로 실행 시
```

```
function jQueryFn(){

    //제이쿼리 반응형
    var section1 = $('#section1');
    var winH = $(window).innerHeight();

    function resizeFn(){
        winH = $(window).innerHeight();
        section1.css({ height: winH });
    }
    resizeFn();
    $(window).resize(function(){
        resizeFn();
    });

}
jQueryFn(); //제이쿼리로 실행시
```

```
    }
});

</script>

<!-- 메인영역 끝 -->
```

```
</div>  
</body>  
</html>
```