

## 전역 컴퍼넌트 생성

```
<!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_전역컴퍼넌트_템플릿</title>

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

  <style>
    #wrapper {width:1200px;margin:0 auto;text-align:center;background:#f0f0f0;}
    #wrapper #header h1 {color:#000;}
    #wrapper #header h2 {color:#279;}
    #wrapper #header h2 mark {background:#ad5;}
  </style>
</head>
<body>
<div id="wrapper">

  <!-- HTML -->
  <div id="header">
    <h1> {{ msg }} </h1>
    <!-- 상위 컴포넌트에서 하위 컴포넌트에게 통신은 Properties(속성) props 이용하여 전달 -->
    <!-- 하위 컴포넌트에서 상위 컴포넌트에게 통신은 Event(이벤트) $emit을 이용하여 전달 -->
    <nav-component></nav-component> <!-- 가상 요소(JSX) 사용자 정의 태그 컴퍼넌트 이름 -->
  </div>

  <!-- SCRIPT -->
  <script>

    // 전역컴포넌트 생성
    Vue.component('nav-component',{
      template:'<h2><mark> 전역컴포넌트 템플릿 속성 </mark></h2>'
    });

    // 뷰 인스턴스 생성
```

```

    var header = new Vue({
      el: '#header',
      data: {
        msg: '전역 컴포넌트 생성'
      }
    });
  </script>
</div>
</body>
</html>

```

## 컴포넌트 통신

### 상위 컴퍼넌트에서 하위 컴퍼넌트로 데이터 전달(통신)

#### [전역 컴퍼넌트 & PROPS]

```

<!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_전역컴퍼넌트_템플릿</title>

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

  <style>
    #wrapper {width:1200px;margin:0 auto;text-align:center;background:#f0f0f0;}
    #wrapper #header h1 {color:#000;}
    #wrapper #header h2 {color:#279;}
    #wrapper #header h2 mark {background:#ad5;}
  </style>

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

  <!-- HTML -->
  <div id="header">

```

```
<h1> {{ msg }} </h1>
<!-- 상위 컴포넌트에서 하위 컴포넌트에게 통신은 Properties(속성) props 이용하여 전달 -->
<!-- 하위 컴포넌트에서 상위 컴포넌트에게 통신은 Event(이벤트) $emit을 이용하여 전달 -->
<nav-component v-bind:txtbind='txt'></nav-component> <!-- 가상 요소(JSX) 사용자 정의 태그 컴퍼넌트 이름 -->
</div>
```

```
<h1 data-teme='gallery'></h1>
```

```
<!-- SCRIPT -->
<script>
```

```
// 전역컴포넌트 생성
Vue.component('nav-component',{
  props:['txtbind'], //바인딩(v-bind:속성이름)할 속성 지정
  template:'<h2><mark><em> {{ txtbind }} </em></mark></h2>'
});
```

```
// 뷰 인스턴스 생성
var header = new Vue({
  el:'#header',
  data:{
    msg:'전역 컴포넌트 생성',
    txt:'상위 컴포넌트에서 하위 컴포넌트로 통신 성공!!!'
  }
});
```

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

## 컴포넌트 통신

[상위 컴퍼넌트에서 버튼 클릭 이벤트 이용 데이터 값 변경하여 하위 컴퍼넌트에 전달하기]

*상위 컴퍼넌트에서 하위 컴퍼넌트로 데이터 전달(통신)*

[전역 컴퍼넌트 & PROPS]

```
<!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_전역컴퍼넌트_템플릿</title>

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

  <style>
    #wrapper {width:1200px;margin:0 auto;text-align:center;background:#f0f0f0;}
    #wrapper #main .button-box {color:#279;padding:50px 0;}
    #wrapper #main .button-box button {cursor:pointer;background:#333;color:#eee;font-size:20px;width:300px;height:50px;}
    #wrapper #main .button-box button:hover {background:#000;}
    #wrapper #main .button-box button:active {background:#922;}
    #wrapper #main .button-box p {font-size:40px;color:#379;color:#000;padding:50px 0;color:#379;}
  </style>
</head>
<body>
<div id="wrapper">

  <!-- HTML -->
  <div id="main">
    <div class="button-box">
      <button v-on:click='countFn'>클릭하세요!!!</button>
      <babo-component v-bind:count='num'></babo-component>
    </div>
  </div>

  <!-- SCRIPT -->
<script>

// 전역컴퍼넌트
Vue.component('babo-component',{
  props:['count'],
  template:'<p> {{ count }} </p>' //하위 컴퍼넌트 props의 속성 이름 'count'로 전달 받음.
});

// 뷰 인스턴스 생성
var main = new Vue({
  el:'#main',
  data:{
    num:0 //루트의 num값이 하위 컴포넌트에 전달(통신)됨.
  },

```

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

[상위 콤퍼넌트에서 반복문 이용 데이터(배열객체)를 하위 콤퍼넌트에 전달하기]  
*상위 콤퍼넌트에서 하위 콤퍼넌트로 데이터 전달(통신)*  
 [전역 콤퍼넌트 & PROPS]

```
<!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_전역컴퍼넌트_템플릿</title>

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

  <style>
    * {margin:0;padding:0;vertical-align:middle;border:0;}
    html,body {width:100%;height:100%;}
    html {overflow-y:scroll;}
    body {font:normal 16px/1 '돋움', 'arial';color:#555;background:#f6f6f6;}
    a {color:#555;text-decoration:none;}
    h1,h2,h3,h4,h5,h6 {font-weight:normal;font-size:16px;}
    ul,li {list-style:none;}
    .clearfix::after{content:'';display:block;clear:both;}
```

```

#wrapper {width:100%;margin:0 auto;}
#wrapper #section5 .notice {width:1170px;margin:0 auto;}
#wrapper #section5 .notice h1 {text-align:center;color:#279;padding:50px 0;font-size:50px;font-weight:normal;}
#wrapper #section5 .notice h2 {font-size:18px;font-weight:600;text-align:center;line-height:50px;border-top:1px solid #bbb;border-bottom:1px solid
#bbb;background:#f0f0f0;}
#wrapper #section5 .notice span {float:left;display:block;}
#wrapper #section5 .notice span.no {width:59px;}
#wrapper #section5 .notice span.subject {width:819px;}
#wrapper #section5 .notice span.date {width:119px;}
#wrapper #section5 .notice span.count {width:103px;}

#wrapper #section5 .notice ul {width:100%;text-align:center;}
#wrapper #section5 .notice ul li {border-bottom:1px solid #ccc;line-height:50px;}
#wrapper #section5 .notice ul li:hover {background:#f0f0f0;}
#wrapper #section5 .notice ul li span.subject {text-align:left;text-indent:20px;}
#wrapper #section5 .notice ul li span.subject a:hover {text-decoration:underline;color:#000;}
</style>
</head>
<body>
<div id="wrapper">

  <!-- HTML -->
  <div id="section5">
    <div class="notice">
      <h1> 공지사항 </h1>

      <h2 class="clearfix">
        <span class='no'>NO</span>
        <span class='subject'>제목</span>
        <span class='date'>날짜</span>
        <span class='count'>조회수</span>
      </h2>

      <ul>
        <!--
          반복문 경우는 item 을 propsbind 에 바인딩 그리고
          템플릿에서 아이템에 속성을 .연결 item.no item.제목
        -->
        <notice-component v-for='row in notice' v-bind:propsbind='row'></notice-component>
      </ul>
    </div>
  </div>

  <!-- SCRIPT -->
  <script>

    Vue.component('notice-component',{

```

```

    props:['propsbind'],
    template:"\
      <li class='clearfix'>\
        <span class='no'>{{ propsbind.no }}</span>\
        <span class='subject'><a href='javascript:boid(0);'>{{ propsbind.제목 }}</a></span>\
        <span class='date'>{{ propsbind.날짜 }}</span>\
        <span class='count'>{{ propsbind.조회수 }}</span>\
      </li>\
    "
  });

```

```

var section5 = new Vue({
  el:'#section5',
  data:{
    notice:[
      {no:301, 제목:"시스템 개선 및 점검 안내", 날짜:"2021-02-25", 조회수:7658},
      {no:300, 제목:"SSG PAY 결제 서비스 일시 이용 중지 안내", 날짜:"2021-02-15", 조회수:45220},
      {no:299, 제목:"스타벅스 홈페이지/애플리케이션 일부 서비스 종료 안내", 날짜:"2021-02-15", 조회수:49113},
      {no:298, 제목:"시스템 개선 및 점검 안내", 날짜:"2021-02-10", 조회수:32728},
      {no:297, 제목:"스타벅스 Web/App 결제 수단 변경 안내", 날짜:"2021-02-10", 조회수:144261},
      {no:296, 제목:"설 연휴 스타벅스 딜리버스 운영 시간 안내", 날짜:"2021-02-09", 조회수:66721},
      {no:295, 제목:"설 연휴 스타벅스 매장 영업시간 변경 안내", 날짜:"2021-02-04", 조회수:117654},
      {no:294, 제목:"시스템 개선 및 점검 안내", 날짜:"2021-02-03", 조회수:7099},
      {no:293, 제목:"개인정보 처리방침 개정안내", 날짜:"2021-01-28", 조회수:7839},
      {no:292, 제목:"위치정보 이용약관 개정안내", 날짜:"2021-01-28", 조회수:5061}
    ]
  }
});

```

</script>

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