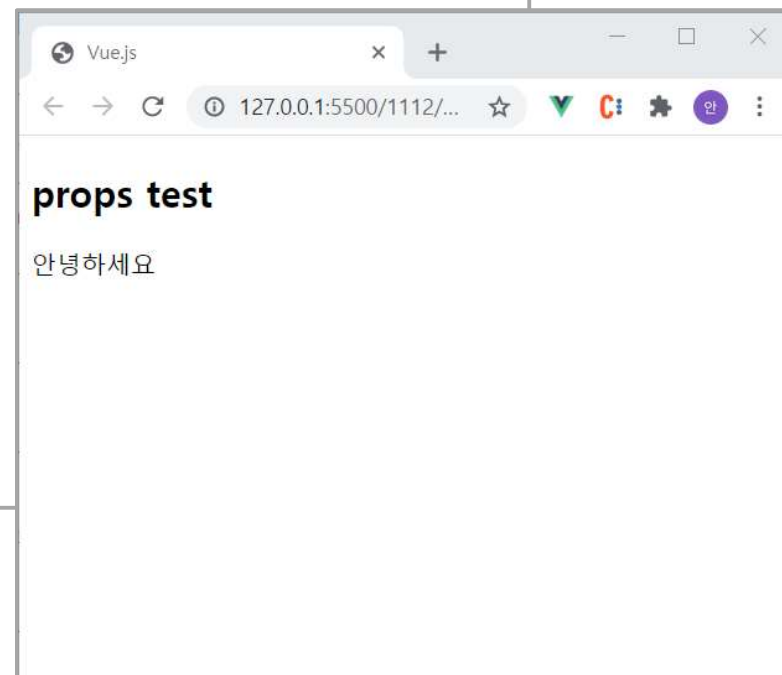


✓ literal props.

Confidential

```
<div id="app">
  <h2>props test</h2>
  <child-component propsdata="안녕하세요"></child-component>
</div>
<script>
  //하위 컴포넌트
  Vue.component('child-component', {
    props: ['propsdata'],
    template: '<span>{{ propsdata }}</span>',
  });
  new Vue({
    el: '#app',
  });
</script>
```



✓ 렌더링 과정.

1. new Vue()로 상위 컴포넌트인 인스턴스를 하나 생성.
2. Vue.component()를 이용하여 하위 컴포넌트인 child-component를 생성.
3. <div id="app"> 내부에 <child-component>가 있기 때문에 하위 컴포넌트가 된다. 처음 생성한 인스턴스 객체가 #app의 요소를 가지기 때문에 부모와 자식 관계가 성립한다.
4. 하위 컴포넌트에 props 속성을 정의한다. ['propsdata']
5. html에 컴포넌트 태그(child-component)를 추가한다.
6. 하위 컴포넌트에 v-bind 속성을 사용하면 상위 컴포넌트의 data의 key에 접근이 가능하다. (message)
7. 상위 컴포넌트의 message 속성 값인 String 값이 하위 컴포넌트의 propsdata로 전달된다.
8. 하위 컴포넌트의 template 속성에 정의된 ' {{ propsdata }}'에게 전달된다.