

Do it! Vue.js 입문 - 03장

뷰 인스턴스

- 뷰로 화면을 개발할 때 필수적으로 생성해야 하는 기본 단위

뷰 인스턴스 생성 방법

- 아래와 같이 Vue 생성자를 이용하여 인스턴스를 생성

```
new Vue({  
  
});
```

뷰 인스턴스 옵션 속성

- 인스턴스를 생성할 때 아래와 같이 옵션 속성을 추가

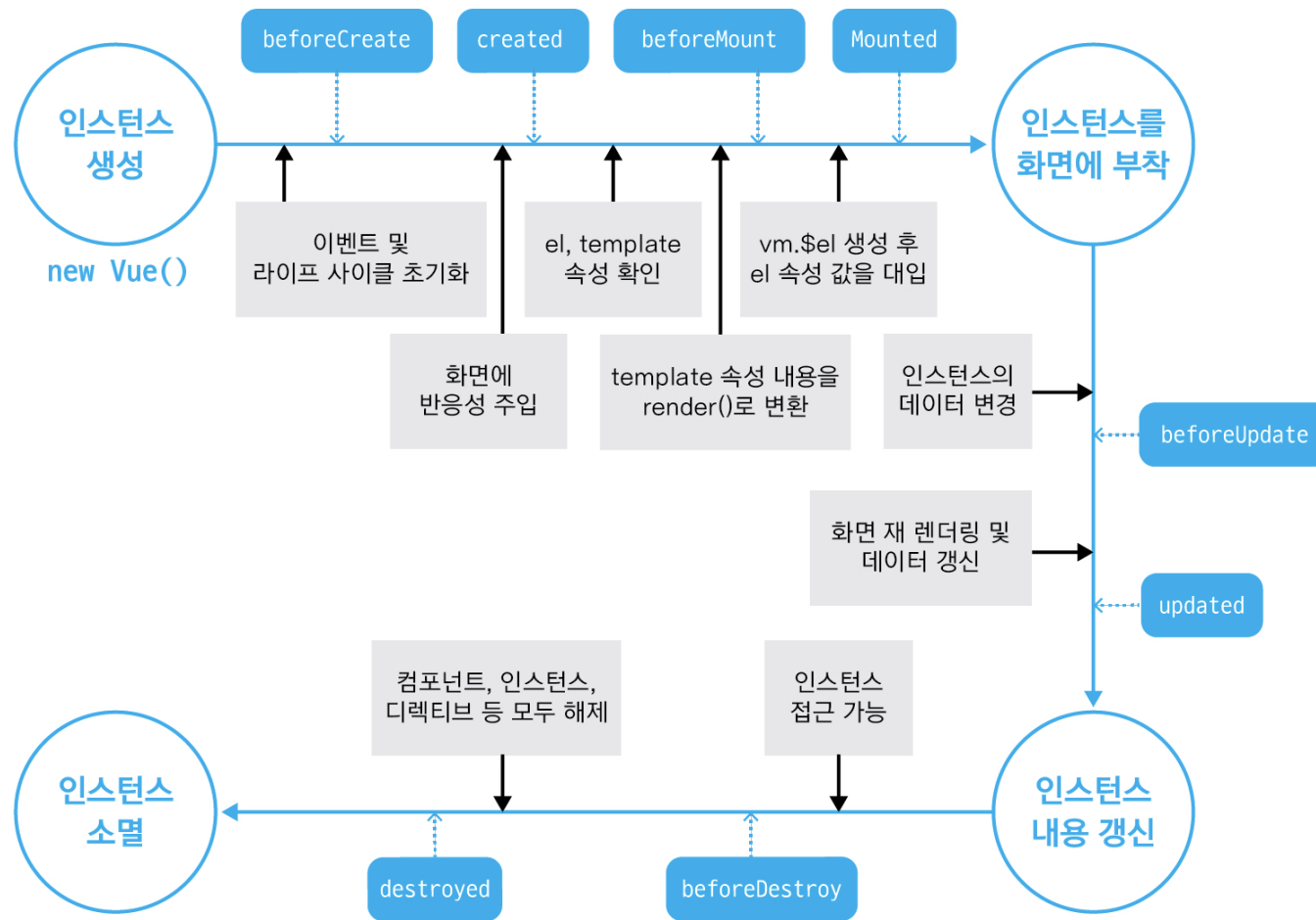
```
new Vue({  
  el: , // 인스턴스의 시작 지점  
  data: , // 인스턴스의 데이터 속성  
  template: , // 인스턴스의 템플릿 속성  
  methods: , // 이벤트 및 화면 동작 메서드  
  created: // 라이프 사이클 커스텀 로직  
});
```

뷰 인스턴스 적용 과정

1. 뷰 라이브러리 로딩
2. 인스턴스 객체 생성
3. 특정 화면 요소에 인스턴스를 부착
4. 인스턴스에 정의한 내용이 화면 요소로 변환
5. 최종 화면 요소를 사용자가 확인

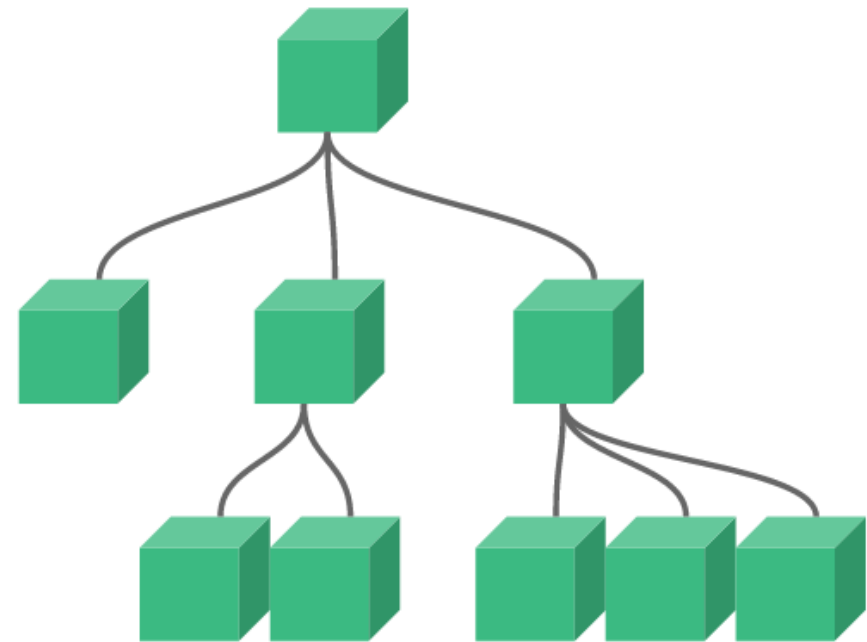
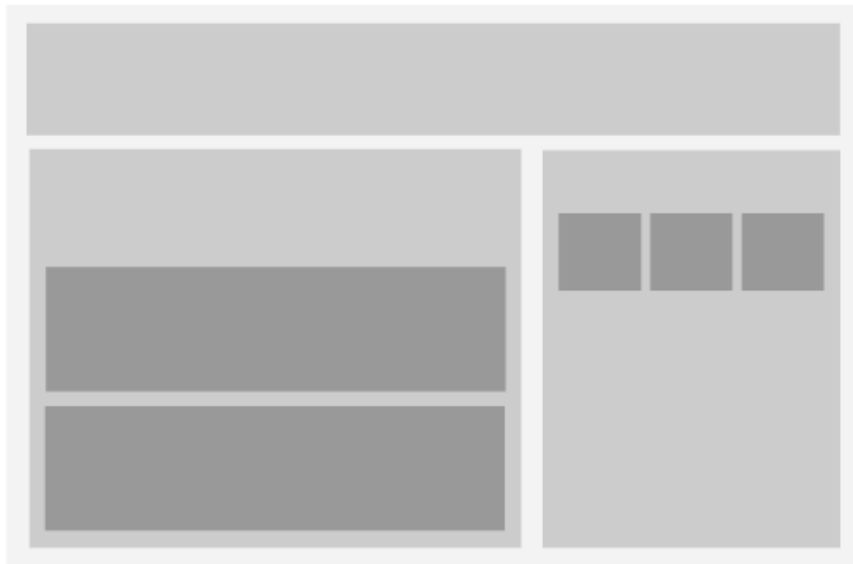
뷰 인스턴스 라이프 사이클

- 뷰 인스턴스가 생성되고 소멸되기까지 아래의 과정을 거침



뷰 컴포넌트

- 컴포넌트란 화면을 구성하는 특정 영역을 의미
- 여러 개를 조합하여 레고 블록처럼 화면을 구성할 수 있음



전역 컴포넌트 등록

- 모든 인스턴스에서 사용 가능한 컴포넌트 등록 방법

```
Vue.component('컴포넌트 이름', {  
  // 컴포넌트 내용  
});
```


전역 컴포넌트 코드 예제

```
<div id="app">  
  <my-cmp></my-cmp>  
</div>
```

```
Vue.component('my-cmp', {  
  template: '<p>Global Component</p>'  
});  
  
new Vue({  
  el: '#app'  
});
```

지역 컴포넌트 등록

- 특정 컴포넌트에서만 사용 가능한 컴포넌트 등록 방법

```
new Vue({  
  components: {  
    '컴포넌트 이름': 컴포넌트 내용  
  }  
});
```

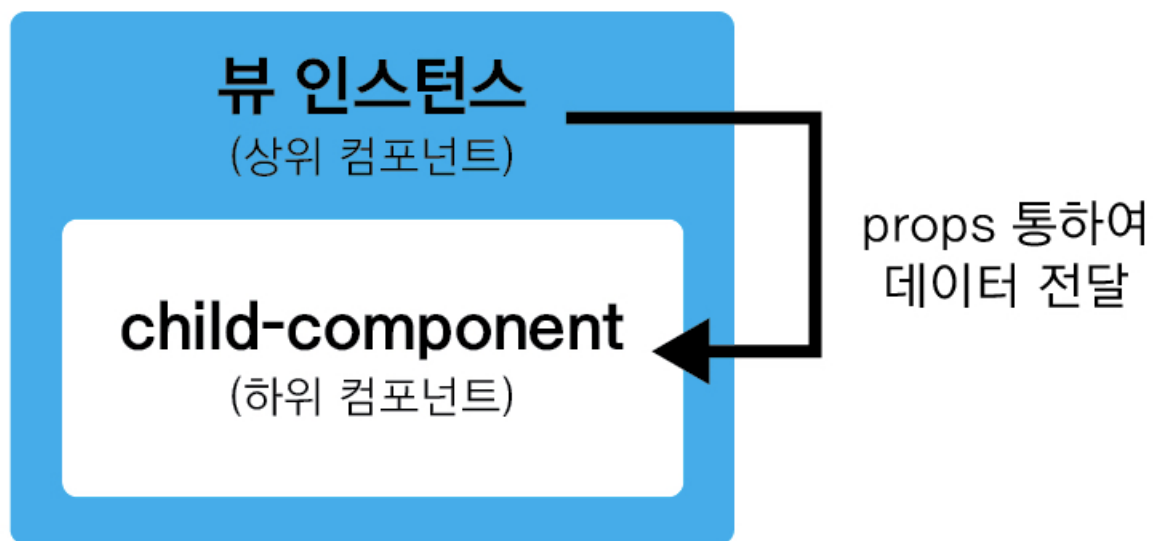
전역 컴포넌트와 지역 컴포넌트 등록 예제

- 아래 링크에서 제공하는 코드를 실행하여 확인

전역, 지역 컴포넌트 등록 코드

컴포넌트 통신 방법

- 뷰 컴포넌트 간에 통신 방법은 아래의 구조를 따름



props 속성

- 상위 컴포넌트에서 하위 컴포넌트로 전달하는 데이터 속성

```
<child-cmp v-bind:propsdata="message"></child-cmp>
```

```
// 하위 컴포넌트
Vue.component('child-cmp', {
  props: ['propsdata'],
  template: '<p> {{ propsdata }} </p>'
});

// 상위 컴포넌트
new Vue({
  data: {
    message: 'parent data'
  }
});
```

이벤트 발생

- 하위 컴포넌트에서 상위 컴포넌트로 통신하는 방법

```
// 하위 컴포넌트  
this.$emit('이벤트 명');
```

```
<child-cmp v-on:이벤트명="상위 컴포넌트의 메서드 명"></child-cmp>
```

이벤트 버스

- 관계가 없는 컴포넌트 간에 통신하는 방법

```
// 이벤트 버스를 위한 별도의 인스턴스 생성
var eventBus = new Vue();

// 이벤트 발생하는 부분
eventBus.$emit('이벤트 명', 인자1, 인자2);

// 이벤트 수신하는 부분
eventBus.$on('이벤트 명', function () {
  // 이벤트를 수신했을 때의 동작 정의
});
```

끝