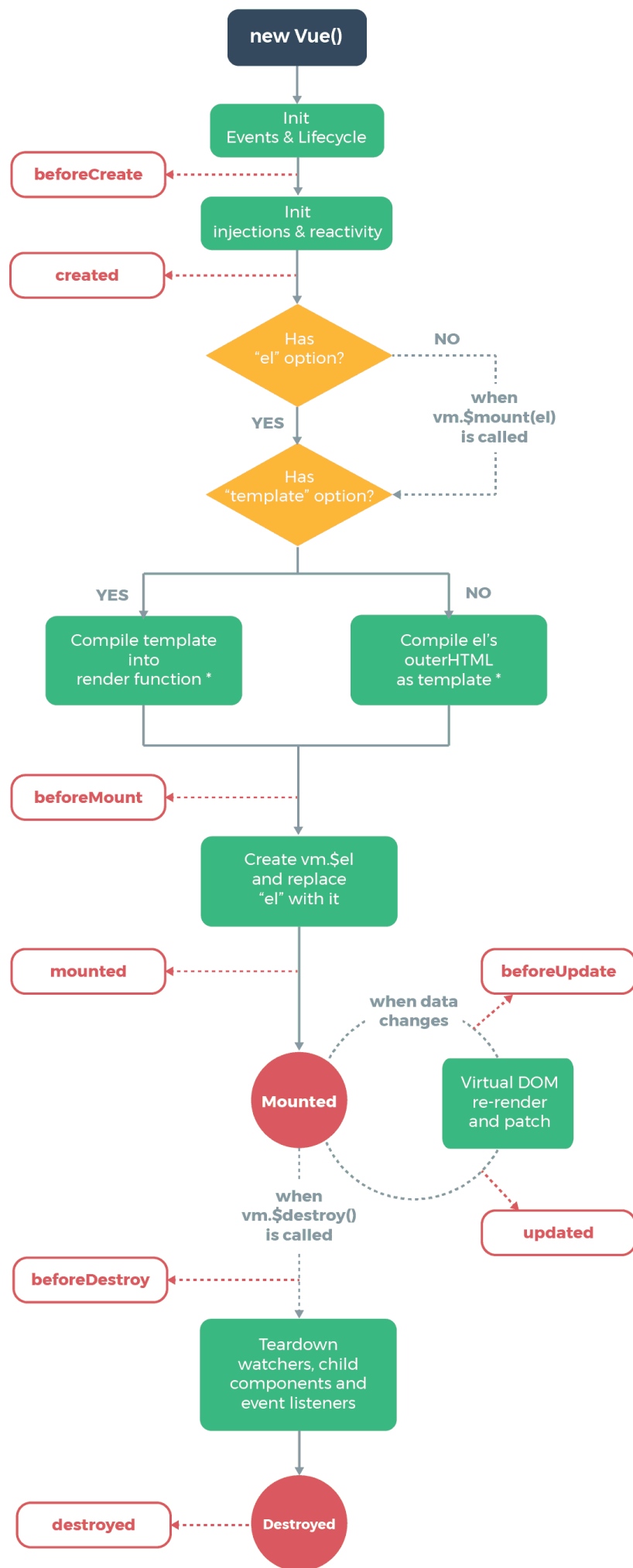




Life Cycle (라이프사이클)

Vue 인스턴스가 생성된 후 우리 눈에 보여지고, 사라지기까지의 단계를 일컫는 말입니다.



* template compilation is performed ahead-of-time if using a build step, e.g. single-file components

```
✖ [Vue warn]: Error in beforeCreate hook: "TypeError: this.requestApi is not a function"
found in
--> <Anonymous>
      <VMain>
        <VApp>
          <App> at src/App.vue
        <Root>

✖ TypeError: this.requestApi is not a function
    at VueComponent.beforeCreate (Notice2.vue?418b:38:1)
    at invokeWithErrorHandling (vue.runtime.esm.js?2b0e:1863:1)
    at callHook (vue.runtime.esm.js?2b0e:4235:1)
    at VueComponent.Vue._init (vue.runtime.esm.js?2b0e:5018:1)
    at new VueComponent (vue.runtime.esm.js?2b0e:5168:1)
    at createComponentInstanceForVnode (vue.runtime.esm.js?2b0e:3304:1)
    at init (vue.runtime.esm.js?2b0e:3133:1)
    at merged (vue.runtime.esm.js?2b0e:3322:1)
    at createComponent (vue.runtime.esm.js?2b0e:6022:1)
    at createElm (vue.runtime.esm.js?2b0e:5969:1)

created Notice2.vue?418b:88
before mounted Notice2.vue?418b:88
CSSStyleDeclaration {accentColor: '', additiveSymbols: '', alignContent: '', alignItems: '', alignSelf: '', ...}
Notice2.vue?418b:50
mounted Notice2.vue?418b:88
2초후 created Notice2.vue?418b:90
2초후 before mounted Notice2.vue?418b:90
2초후 mounted Notice2.vue?418b:90
clicked Notice2.vue?418b:76
before update Notice2.vue?418b:88
update Notice2.vue?418b:88
before update Notice2.vue?418b:88
update Notice2.vue?418b:88
2초후 before update Notice2.vue?418b:90
2초후 update Notice2.vue?418b:90
2초후 before update Notice2.vue?418b:90
2초후 update Notice2.vue?418b:90
```

1. Creation 컴포넌트 초기화

☐ beforeCreate

- Vue 인스턴스가 초기화 된 직후에 발생합니다. 컴포넌트가 DOM에 추가되기도 전
- data, event, watcher에도 접근하기 전이라 `data`, `method` 에도 접근할 수 없습니다.

☐ created

- `data` 를 반응형으로 추적할 수 있게 되며 `computed`, `methods`, `watch` 등이 활성화되어 접근이 가능하게 됩니다. 하지만 아직까지 DOM에는 추가되지 않은 상태

2. Mounting DOM 삽입단계

☐ beforeMount

- DOM에 부착하기 직전에 호출되는 **beforeMount** 훅
- 가상 DOM이 생성되어 있으나 실제 DOM에 부착되지는 않은 상태

☐ mounted

- 가상 DOM의 내용이 실제 DOM에 부착되고 난 이후에 실행되므로, `this.$el` 을 비롯한 `data`, `computed`, `methods`, `watch` 등 모든 요소에 접근이 가능

** 부모와 자식 컴포넌트 간의 mounted 훅 순서

일반적인 경우에, 부모와 자식 컴포넌트 간의 mounted 훅 순서는 위의 그림과 같습니다. 부모 컴포넌트의 mounted 훅은 항상 자식 컴포넌트의 mounted 훅 이후에 발생한다는 것을 알 수 있습니다.

```
mounted(function() {
  this.$nextTick(function() {
    // 모든 화면이 렌더링된 후 실행합니다.
  })
})
```

자식 컴포넌트가 서버에서 비동기로 데이터를 받아오는 경우처럼, 부모의 mounted 훅이 모든 자식 컴포넌트가 마운트 된 상태를 보장하지는 않습니다. 따라서 이때는 `this.$nextTick` 을 이용한다면, 모든 화면이 렌더링 된 이후에 실행되므로 마운트 상태를 보장할 수 있습니다.

3. Updating

컴포넌트에서 사용되는 반응형 속성들이 변경되거나 어떠한 이유로 재렌더링 발생 시 실행된다.

디버깅이나 프로파일링 등을 위해 컴포넌트 재 렌더링 시점을 알고 싶을 때 사용하면 된다.

☐ beforeUpdate

컴포넌트의 데이터가 변하여 업데이트사이클이 시작될 때 실행된다. 정확히는 DOM이 패치되기 전 데이터가 변경될 때 호출된다.

- 업데이트 전에 기존 DOM에 접근할 수 있다.
- 재 렌더링 전이 새 상태의 데이터를 얻을 수 있고 더 많은 변경이 가능하다

☐ updated - UI가 변경되는 시점

데이터가 변경되어 가상 DOM이 재렌더링되고 패치된 후에 호출 됩니다.

- updated가 호출될 때 컴포넌트의 DOM이 업데이트되므로 여기서 DOM의 종속적인 연산을 할 수 있습니다.
- 그러나 여기서 상태를 변경하면 무한루프에 빠질 수 있다.
- 모든 자식 컴포넌트의 재 렌더링 상태를 보장하지 않는다.

전체 화면이 재렌더링 될 때까지 기다리려면, updated 내부에서 `vm.$nextTick`을 사용합니다.

4. Destruction 해체 단계

☐ beforeDestroy

- 해체 되기 직전에 호출된다.

☐ destroyed

- 해체 된 후에 호출된다.
- Vue 인스턴스의 모든 디렉티브가 바인딩 해제되고모든 이벤트 리스너가 제거되면 모든 하위 Vue인스턴스도 삭제된다.

```
beforeCreate() {
  console.log("before create")
},
created() {
  console.log("created")
},

beforeMount() {
  console.log("before mount")
},

beforeUpdate() {
  console.log("before update")
},
updated() {
  console.log("update")
},

beforeDestory() {
  console.log("before destroy")
},
destroyed() {
  console.log("before destroyed")
},

mounted() {
  console.log('mounted')
  isUpdated = true;
  this.loadData();

  if (!localStorage.getItem("guideView")) {
    this.showGuideTutor = true;
    localStorage.setItem("guideView", true);
  }
  if (this.$store.state.isCompleted) {
    this.showCompleteAlert = true;
    this.alertMessage =
      " 🎉성공적으로 분석되었습니다! \n 이제 보트는 휴지통에 버려주세요";
  }
}
```

beforeCreate

created

beforeMount

mounted

before create	Home.vue?76f2:254
created	Home.vue?76f2:257
before mount	Home.vue?76f2:262
mounted	Home.vue?76f2:285
before update	Home.vue?76f2:267
update	Home.vue?76f2:271
▶ {data: {…}, status: 200, statusText: 'OK', headers: {…}, config: {…}, …} interceptors.js?a3fd:24	
before update	Home.vue?76f2:267
update	Home.vue?76f2:271

beforeUpdate

update

before update	Home.vue?76f2:267
update	Home.vue?76f2:271
before update	Home.vue?76f2:267
update	Home.vue?76f2:271
before update	Home.vue?76f2:267
update	Home.vue?76f2:271

beforeDestroy

before destroyed	Home.vue?76f2:280
▶ {data: {…}, status: 200, statusText: 'OK', headers: {…}, config: {…}, …} interceptors.js?a3fd:24	
data load	Myepage.vue?f55f:87

***부모와 자식 컴포넌트의 데이터의 흐름**

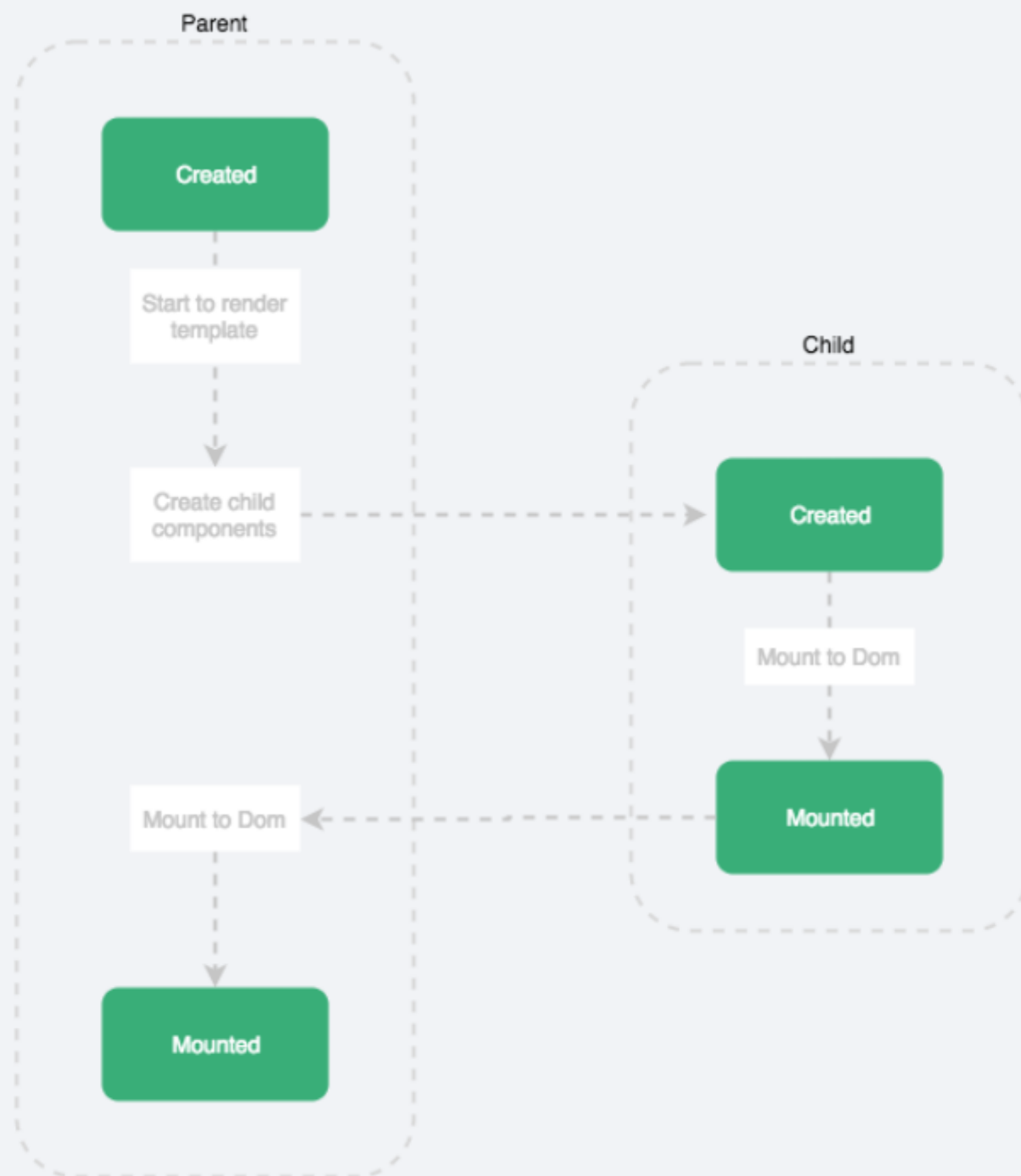
Parent created → Child created → Child mounted → Parent mounted

즉, data 호출 시점과 data를 props를 통해 자식컴포넌트에 넘겨줄때를 알아야한다.

mounted혹은 새로운 컴포넌트가 DOM에 추가가 되고 난 후에 실행되는 혹입니다.

컴포넌트 간에 계층 관계가 있을 경우에 created와 반대로, mounted 혹은 자식 컴포넌트가 부모 컴포넌트보다 먼저 실행됩니다.

아래 그림은 부모 컴포넌트와 자식 컴포넌트 간에 created 혹과 mounted 혹의 실행 시점을 표시한 그림입니다.



Mounted에서는 자식컴포넌트가 부모 컴포넌트보다 먼저 실행됩니다.