

本讲目标

- 虚拟DOM
- Patch算法

什么是虚拟DOM

Object

```
anchor: null
▶ appContext: {config: {...}, mixins: Arr
children: null
component: null
dirs: null
dynamicChildren: null
dynamicProps: null
el: null
key: null
patchFlag: 0
props: null
ref: null
scopeId: null
```


为什么需要虚拟dom

- 性能
- 兼容性
- 跨平台
- 组件更新

何时会用到虚拟dom

```
const patch: PatchFn = (
```

```
  n1,
```

```
  n2,
```

```
  container,
```

```
  anchor = null,
```

```
  parentComponent = null,
```

```
  parentSuspense = null,
```

```
  isSVG = false,
```

```
  optimized = false
```

```
) => {
```

```
Evan You, 7 months ago
```

初始化时n1为null
更新时比对n1、n2

创建VNode

```
mount(rootContainer: HostElement, isHydrate?: boolean): any {  
  if (!isMounted) {  
    const vnode = createVNode(rootComponent as Component, rootProps)  
    // store app context on the root VNode.  
    // this will be set on the root instance on initial mount.  
    vnode.appContext = context  
    // HMR root reload
```


渲染VNode

```
const render: RootRenderFunction = (vnode, container) => {  
  if (vnode == null) {  
    if (container._vnode) {  
      unmount(container._vnode, null, null, true)  
    }  
  } else {  
    patch(container._vnode || null, vnode, container)  
  }  
  flushPostFlushCbs()  
  container._vnode = vnode  
}
```


渲染VNode

```
const mountComponent: MountComponentFn = (
  initialVNode,
  container,
  anchor,
  parentComponent,
  parentSuspense,
  isSVG,
  optimized
) => {
```

```
  setupRenderEffect(
    instance,
    initialVNode,
    container,
    anchor,
    parentSuspense,
    isSVG,
    optimized
  )
```