1.
2. APIprops,event,slot,API
3. #appVue3Teleport

- .
-
- render
-
-
- TS
-
- ...

Vue

Vue

- -> props
- ->
- provie/inject ->
- Event Busonemit
- $children
- Vuex

Vue

1. createdbeforeCreatedatapropsmethods,createdcreated
2. mountedcreatedmountedcreatedmounted

Vuedata

VueVuedatadata

data

computedwatch

computed

- computed
- computedcomputed

watch

- watch
- watch
    - immediatewatchimmediate
    - deepwatch

- computedwatchVueinitWatcherinitComputedwatchcomputedwatchcomputed Watcher
- computedWatcherlazy Watcher
- computedWatcherdirty

Vue

- VueObject.defineProperty
- VueReactdiffVuegetterWatcherWatcherWatcherDep

Vue

Vue2

Object.definePropertyVue2

Vue2push()pop()shift()unshift()splice()sort()reverse()Vue

$setAPI

MVVMVue

MVVMModel-ViewView-Model

Vuev-model

ReactReactReact

MVVMUIAndroid JetpackData BindingIOSSwift UIMVCMVPMVVM

Vue

- -> ->
- Vue
- input

DOMdiff

DOMJSDOMUIVueVNode

DOM

- JSJSSSR
- DOMDOMJSDOMdiffDOMSvelteDom

Vue diff

- SnabbdomVuediffSnabbdomdiff
- VuediffvnodeDOMDOM
- key diff vnode key key vnodeVueDOMDOM
- Vuediff
- keyDOM

Vue

template3parse -> optimize -> generate

1. parse() AST
2. optimize() class style Vue
3. generate() AST

Vue

- JestMochaVue
- @vue/test-utilsVue

-
- APIprops,event,slot
-
-
-
-
- watch
- Vuex
-

Button

```
// Button
import { mount } from '@vue/test-utils'
import Button from '@/components/Button.vue'

describe('Button', () => {
  it('renders default button', () => {
    //
    const wrapper = mount(Button)

    // html
    expect(wrapper.html()).toContain('<button class="btn">Button</button>')
  })

  it('renders primary button', () => {
    // primaryButton
    const wrapper = mount(Button, {
      propsData: {
        primary: true
      }
    })

    // html
    expect(wrapper.html()).toContain('<button class="btn btn-
primary">Button</button>')
  })

  it('triggers click event', () => {
    // mock
    const mockFn = jest.fn()

    // Buttonmock
    const wrapper = mount(Button, {
      propsData: {
        onClick: mockFn
      }
    })

    //
    wrapper.trigger('click')

    // mock
    expect(mockFn).toHaveBeenCalled()
  })
})
```

VueDOMloading

v- v-mydirective

- bind
- inserted UI
- updateUI
- componentUpdatedDOM
- unbind

```
Vue.directive('focus', {
  inserted: function (el) {
    el.focus()
  }
})
```

```
<input v-focus>
```

keep-alive

1. keep-aliveVuekeep-alivetab

keep-alive

```
<template>
  <div>
    <keep-alive>
      <router-view v-if="$route.meta.keepAlive"></router-view>
    </keep-alive>
    <router-view v-if="!$route.meta.keepAlive"></router-view>
  </div>
</template>
```

2.

- keep-alive
- LRULRUkeep-aliveLRUkeep-alive

nextTick

nextTickdom

1. VueVueDom
2. nextTicknextTicksetTimeout
3. VuenextTickPromisePromisesetImeediate -> setTimeout

SSR

- SSRVueCSR
- SSRSEO
- SSR

- SSRVueNuxtReactNext
- SSRAPI
  SSRSEOSEO

Vue3Vue2

- ProxyObject.defineProperty
- Block
- slot
- Tree ShakingAPI

APIReactive API

- API
- refAPI
- Vue3Vue2

TypeScript

Vue3TypeScriptVue3APITS

VueWebGLCanvas

- Monorepo
- APIReactive API
-

- Teleport
-

Vue

1. iffor
2. Object.freeze()
3. forkeykey
4.
5. props
6. if/else
7.