

## Part 3 Vue.js 框架源码与进阶

---

- 任务一 Vue.js基础回顾
  - 任务二 Vue-Router 原理实现 （建议看文档笔记）
  - 任务三：Vue响应式原理
  - 任务四：Virtual Dom 虚拟DOM
    - 什么是虚拟DOM
    - 为什么使用虚拟DOM
    - 虚拟DOM的作用和虚拟DOM的库
    - 创建项目
    - 导入 Snabbdom
    - 模块
    - Snabbdom源码解析
    - h函数
      - h()函数介绍
        - 在使用Vue的时候见过h()函数 （Vue中的进行了提升可以进行组件机制）
- ```
1  new Vue({  
2    router,  
3    store,  
4    render: h => h(App)  
5  }).$mount('#app')
```
- h() 函数最早见于 hyperscript，使用Javascript创建超文本
  - Snabbdom中的h()函数不是用来创建超文本的，而是创建VNode
  - 函数重载
    - 概念
      - 参数个数或类型不同的函数
      - JavaScript中没用重载的概念
      - TypeScript中有重载，不过重载的实现还是通过代码调整参数
    - h函数的重载
      - 代码

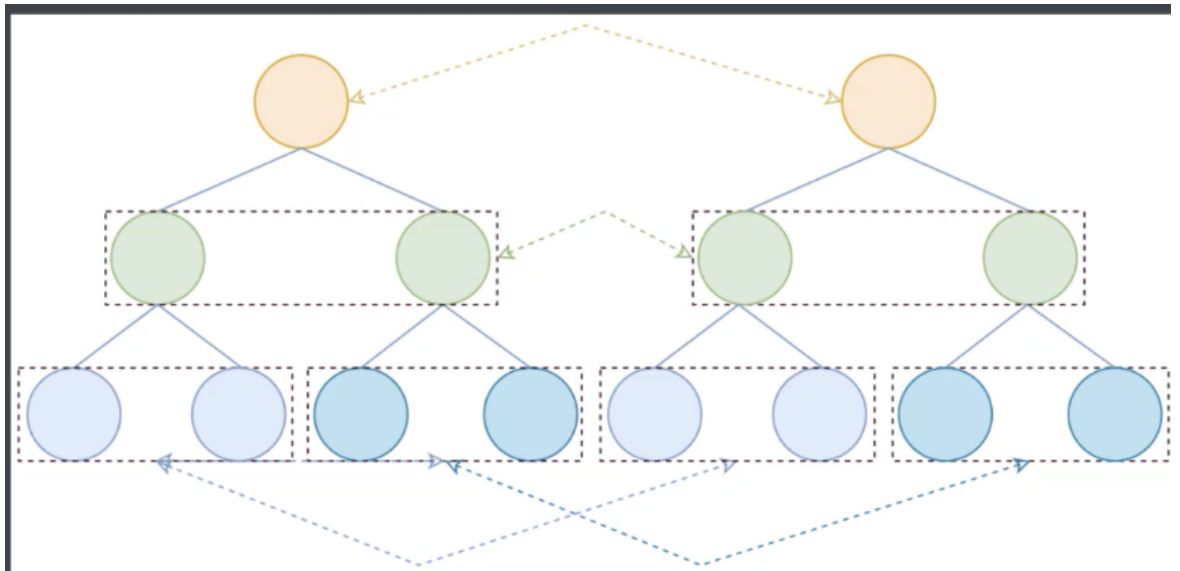
```

// h 函数的重载
export function h(sel: string): VNode;
export function h(sel: string, data: VNodeData | null): VNode;
export function h(sel: string, children: VNodeChildren): VNode;
export function h(sel: string, data: VNodeData | null, children: VNodeChildren): VNode;
export function h(sel: any, b?: any, c?: any): VNode { ...
};
// 导出模块
export default h;

```

## ● 必备快捷键

- vnode
- patch整体过程（VNode如何编程真实DOM的）
  - 打补丁，把新节点中变化的内容渲染到真实DOM，最后返回新节点作为下一次处理的旧节点
  - 对比新旧Vnode是否是相同节点（节点的key和sel相同）
  - 如果不是相同节点，删除之前的内容，重新渲染
  - 如果相同节点，再判断新的VNode是否有text，如果有并且和oldVnode的text不同，直接更新文本内容
  - 如果新的VNode有children，判断子节点是否有比那话，判断子节点的使用过程就是diff算法
  - diff过程只进行同层级的比较



## ● init函数

-