# Vue3

1. 风格
   1. 选项式API（vue2）
   2. 组合式API（vue3）
2. 指令
   1. CNPM INSTALL
3. vscode:扩展插件volar
4. 文本插值：Mustache 语法 即双大括号 '{{ }}'
5. 原始HTML：v-html
6. 属性绑定：v-bind
   1. 简写：

<div v-bind:id="dynamicID" v-bind:class="dynamicClass" >测试</div>未简写
​<div :id="dynamicID" :class="dynamicClass" >测试</div>简写

* 1. 动态绑定多个值
     1. 代码

<div v-bind="objectOfAttrs">动态绑定多个值</div>
objectOfAttrs:{
class:'active',
id:'testId'}
​

* + 1. 网页：
       1. show

<div class="active" id="testId">动态绑定多个值</div>

1. 条件渲染：v-if、v-else、v-else-if、v-show
   1. v-if:用于条件性的渲染一块内容，条件为真(turn)
   2. v-if VS v-show
      1. v-if:是‘真实的’按条件渲染，因为他确保了在切换时，条件区块的事件监听器和子组件都会被销毁和重建
         1. 也是‘惰性’的：如果初次条件渲染为false，则不会做任何事，条件区块只有当条件首次变为trun时才被渲染
         2. 有更高的切换开销
         3. 建议使用：绑定条件很少改变绑定条件
   3. v-show:元素无论初始条件如何，始终被渲染 只有CSS display 会被切换
      1. 有更高的初始渲染开销
      2. 建议使用：频繁切换绑定条件
2. 列表渲染：v-for
   1. v-for：支持可选的第二个参数表示当前的位置索引

<p v-for="(item,index) in names">{{item}}-{{index}}</p>

* + 1. 可以使用’of‘作为分隔符来代替in ，这更接近JavaScript的迭代器语法

<p v-for="item of names">{{item}}</p>

* + 1. 遍历一个对象的所有属性
  1. 代码

<script>export default {
data(){
return{
names:[
"百战程序员","尚学堂","IT" ],
result:[{
'id':'123123',
'title':'awd',
'avator':'<https://images.shejidaren.com/wp-content/uploads/2019/09/31475-6.jpg'> },
{
'id':'4444',
'title':'sfs',
'avator':'<https://images.shejidaren.com/wp-content/uploads/2019/09/31475-6.jpg'> },
{
'id':'6666',
'title':'fdn',
'avator':'<https://images.shejidaren.com/wp-content/uploads/2019/09/31475-6.jpg'> },
],
userInfo:{
name:'iwen',
age:20,
sex:'男' }
}
}
}
</script><template><h3>列表渲染</h3> <p v-for="item in names">{{item}}</p> <p v-for="item of names">{{item}}</p> <div v-for="item in result"> <p>{{item.id}}</p> <p>{{item.title}}</p> <img :src="item.avator" width="50px"></img> </div> <div>支持可选-下标</div> <div> <p v-for="(item,index) in names">{{item}}-{{index}}</p> </div> <div>遍历对象</div> <div> <p v-for="item of userInfo"> {{item}}
</p><!-- 还可以有三个参数--> <p v-for="(value,key,index) of userInfo"> {{value}}-{{key}}-{{index}}
</p> </div></template>

1. v-for:key属性

<div> <p v-for="(item,index) of names" key="index">{{item}}</p></div>

1. 事件处理:v-on：click 简写 @ click
   1. 内联事件处理器
      1. 事件触发时执行的内联javaScript 语句(与onclick类似)
   2. 方法事件处理器
2. 事件参数
   1. 获取event对象
   2. 传递参数
   3. 传参获取event对象
3. 事件修饰符：简化代码 ：.stop、.prevent、.once、.enter
   1. 阻止默认事件
   2. 阻止事件冒泡
4. 数组变化的侦测
   1. 变更方法：push()、pop()、shift()、unshift()、splice()、sort()、revers()、
   2. 替换一个数组：不对原数据变更，而是返回一个新数据：filter()、concat()、slice()、
5. 计算属性
   1. computed
6. class绑定
   1. 数据绑定操作元素的CSS class列表
   2. 绑定多个对象
7. STYLE绑定
   1. 对象绑定
   2. 数组绑定
8. 侦听器：监听数据的变化
   1. watch
9. 表单输入绑定：v-mode
   1. 复选框
   2. 修饰符：.lazy、.number、.trim\
10. 模板引用
    1. ref
11. 组件组成 -》重要
12. 组件的嵌套关系
13. 组件的注册方式
    1. 全局注册
    2. 局部注册
14. 组件传递数据：props
    1. 动态数据传递
    2. 注意事项：只能父传子
15. 组件传递多种数据类型
16. 组件传递props校验
17. 组件事件
    1. 组件传递数据
       1. 方案：父传子：props 子传父：自定义事件（this.$emit）
    2. 组件事件配合v-model
    3. 组件数据传递：函数
       1. props也可以实现子传父
18. 透传：Attributes
19. 插槽slot
    1. 具名插槽
    2. 插槽数据传递
20. 组件的生命周期
21. 组件的生命周期应用
22. 动态组件
23. 组件保持存活：keep-alive
24. 异步组件