# 阶段一：VUE3学习

学习链接：【前端最新Vue2+Vue3基础入门到实战项目全套教程，自学前端vue就选黑马程序员，一套全通关！】<https://www.bilibili.com/video/BV1HV4y1a7n4?p=144&vd_source=67ef8f54727a0cebc492490009cdb9fa>

# 组合式API - setup选项

## 1. setup选项的写法和执行时机

写法

<script>

export default {

setup(){

},

beforeCreate(){

}

}

</script>

> 在beforeCreate钩子之前执行

## 2. setup中写代码的特点

> 在setup函数中写的数据和方法需要在末尾以对象的方式return，才能给模版使用

<script>

export default {

setup(){

const message = 'this is message'

const logMessage = ()=>{

console.log(message)

}

// 必须return才可以

return {

message,

logMessage

}

}

}

</script>

## 3. <script setup>语法糖

> script标签添加 setup标记，不需要再写导出语句，默认会添加导出语句

<script setup>

const message = 'this is message'

const logMessage = ()=>{

console.log(message)

}

</script>

# 组合式API - reactive和ref函数

## 1. reactive

> 接受对象类型数据的参数传入并返回一个响应式的对象

<script setup>

// 导入

import { reactive } from 'vue'

// 执行函数 传入参数 变量接收

const state = reactive({

msg:'this is msg'

})

const setSate = ()=>{

// 修改数据更新视图

state.msg = 'this is new msg'

}

</script>

<template>

{{ state.msg }}

<button @click="setState">change msg</button>

</template>

2. ref

> 接收简单类型或者对象类型的数据传入并返回一个响应式的对象

<script setup>

// 导入

import { ref } from 'vue'

// 执行函数 传入参数 变量接收

const count = ref(0)

const setCount = ()=>{

// 修改数据更新视图必须加上.value

count.value++

}

</script>

<template>

<button @click="setCount">{{count}}</button>

</template>