

Vue3 Introduction



同花顺 T2 职级课 – Vue3 专题

分享: 沈佳棋

部门: B2C开发基础架构组





目录

1. 重构后的 Vue3 带来了什么?
2. Vue3 核心知识点讲解
3. Composition API
4. 调试: Vue DevTools
5. Vue3 生态下, 我们能做些什么?
6. Vue3 学习资料推荐
7. 作业

重构后的 Vue3 带来了什么？

新特性:

- 使用 proxy 代理, 修复 Vue2 响应式弊端
- **Composition API**
- TypeScript 支持
- patchFag 静态标记
- 优化跨端问题
- 新组件 `Fragment`、`Teleport`、`Suspense`
- 新增 `v-memo` 缓存 html 模板
- 支持 TreeShaking
- 自定义渲染器
- 构建工具 Vite
- ...

框架性能提升:

- 资源大小减少 `41%`
- 初始渲染速度加快 `55%`
- 更新速度加快 `133%`
- 内存占用减少 `54%`

Vue3 核心知识点讲解

新文档:

<https://staging-cn.vuejs.org/>

Vue 3 已经成为当前默认版本！[了解更多](#)

Vue.js

Q Search

⌕ K

文档 ∨APIPlayground生态系统 ∨语言 ∨关于 ∨赞助者...

渐进式
JavaScript 框架

一款用于构建 Web 界面，易学易用，性能出色且功能丰富的框架。

▶ Why Vue

快速开始 →

安装

特别赞助

HBuilder

为 Vue 打造的先进 IDE

易学易用

基于标准 HTML、CSS 和 JavaScript 构建，拥有直观的 API 和世界一流的文档。

性能出色

经过编译器优化、完全响应式的渲染系统，几乎不需要手动优化。

功能丰富

拥有丰富的、可渐进式集成的生态系统，可以根据规模在仅使用一个库和使用整套框架间切换自如。

Vue 3 已经成为当前默认版本！[了解更多](#)

Vue.js

Q Search

⌕ K

文档 ∨APIPlayground生态系统 ∨语言 ∨关于 ∨赞助者...

组合式 API

setup()

Basic Usage

Accessing Props

Setup Context

Usage with Render Functions

响应式: 核心

ref()

computed()

reactive()

readonly()

watchEffect()

watchPostEffect()

watchSyncEffect()

watch()

响应式: 工具

isRef()

unref()

toRef()

toRefs()

isProxy()

isReactive()

isReadonly()

响应式: 进阶

shallowRef()

triggerRef()

customRef()

shallowReactive()

shallowReadonly()

toRaw()

markRaw()

effectScope()

getCurrentScope()

onScopeDispose()

生命周期钩子

onMounted()

onUpdated()

onUnmounted()

onBeforeMount()

onBeforeUpdate()

onBeforeUnmount()

onErrorCaptured()

onRenderTracked()

onRenderTriggered()

onActivated()

onDeactivated()

onServerPrefetch()

依赖注入

provide()

inject()

Vue3 核心知识点:

- 内置新增组件 ``Fragment``、``Teleport``、``Suspense``
- ``watch`` 与 ``watchEffect`` 的区别
- `<script setup>` 语法 (3.2+)
- ``css`` 动态变量注入 (3.2+)
- ``Composition API``
- ...

[Try in playground](#)

Composition API 

Compostion API 是 Vue3 中引入的编写组件的新方式

☀ Light

OPTIONS API

```
export default {
  data() {
    return {
      media: matchMedia(
        '(prefers-color-scheme: dark)'
      ),
      dark: false,
    },
  },
  methods: {
    toggleDark() { this.dark = !this.dark },
    update() { this.dark = this.media.matches },
  },
  created() {
    this.media.addEventListener('change', this.update)
    this.update()
  },
  destroyed() {
    this.media.removeEventListener('change', this.update)
  }
}
```

COMPOSITION API

```
import { ref, onUnmounted } from 'vue'
export default {
  setup() {
    const media = matchMedia(
      '(prefers-color-scheme: dark)'
    )
    const dark = ref(media.matches)

    const toggleDark = () => dark.value = !dark.value
    const update = () => dark.value = media.matches

    update()
    media.addEventListener('change', update)
    onUnmounted(() => {
      media.removeEventListener('change', update)
    })

    return { dark, toggleDark }
  }
}
```

`Options API` VS `Composition API`

对象式 API 存在的问题

- 不利于复用
- 潜在命名冲突（mixin）
- 上下文丢失
- 有限的类型支持
- 按 API 类型组织

组合式 API 提供的能力

- 极易复用 (原生 JS 函数)
- 可灵活组合 (生命周期钩子可多次使用)
- 提供更好的上下文支持
- 更好的 TypeScript 类型支持
- 按功能/逻辑组织（思维方式变更）
- 可独立于 Vue 组件使用（`@vue/reactivity`）

But...

使用 Composition API 的思想误区:

没有了这些 Option 的限制, ``data``、``computed``、``method`` 我可以随便写在函数的某个位置了 ❌

解释: Composition API 最重要的目的是帮助我们做功能拆分 && 功能组合, 而不是说一个组件 setup 函数里面密密麻麻写了 2000 行代码

Composition API 的 "缺点":

非常考验开发对功能的抽象能力, 抽离方式不当的话反而会适得其反, 降低代码的可维护性。

所以 Vue3 一方面给前端开发带来了更高的灵活性, 也对开发人员在代码设计层面提出了更高的要求。

Note: 新技术出来的时候也不要一味拉踩老技术, Option API 与 Composition API 体现的仅仅是编程抽象思想上的不同, 因此 Vue3 也兼容了 Option API 的写法。

Composition API 示例:

- `useDark`
- **`useStorage`**
- **`useEventListener`**
- **`useFetch`**

useStorage



name: 'Banana'

color: 'Yellow'

size: 'Medium'

count: 0

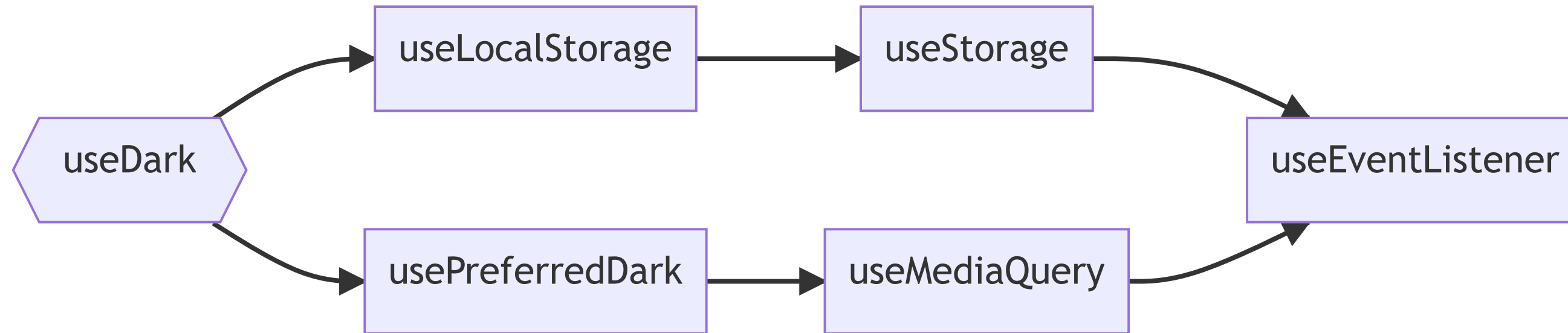
```
import { useStorage } from '@vueuse/core'
import { stringify } from '@vueuse/docs-utils'

const state = useStorage('vue-use-local-storage', {
  name: 'Banana',
  color: 'Yellow',
  size: 'Medium',
  count: 0,
})

const text = stringify(state.value)
```

上面代码中的 `state` 和 `text` 均为响应式数据，仅需如上注册一次，我们就无须再考虑监听数据变化再进行 storage 存取操作的事情。

逻辑组合



- 其中每一个函数都可以独立使用
- 专注点分离
- 建立 输入 → 输出 的连结
- 输出会自动根据输入的改变而改变

useEventListener

```
import { useEventListener } from '@vueuse/core'

const element = ref<HTMLDivElement>()
useEventListener(element, 'keydown', (e) => { console.log(e.key) })
```

```
import { onUnmounted } from 'vue'

export function useEventListener(target: EventTarget, name: string, fn: any) {
  target.addEventListener(name, fn)

  onUnmounted(() => {
    target.removeEventListener(name, fn) // <--
  })
}
```

自动清除副作用，仅需封装一次，使用时无须考虑事件导致的内存泄露问题

useFetch

使用 `useFetch`，我们甚至可以将异步请求转换为 "同步"

正常的异步请求

```
const data = await fetch('https://api.github.com/').then(r => r.json())

// use data
```

使用 `USEFETCH`

```
const { data } = useFetch('https://api.github.com/').json()

const user_url = computed(() => data.value?.user_url)
```

先建立数据间的“连结”，然后再等待异步请求返回将数据填充。概念和 React 中的 SWR (stale-while-revalidate) 类似。

Any More?



VueUse

调试： Vue DevTools

调试组件

<DailyLessson>727 × 2845.18

下载APP | 图标

课程训练营每日一课大厂案例

每日一课案例精选

10 分钟解决一个技术难题

点击查看

小编推荐

最热

11:10

2022高效学习法：如何成为知识的主人？

怎么选择适合自己的内容去学习，怎么更好、更高效地掌握学习内容是我们要关注的问题

#个人成长

个人成长 · 5665次播放

上新

10:22

风轻云淡——如何用好对象存储（上）

云上存储已经是目前主流的存储之一，几乎所有的服务都要用到对象存储

#运维/测试

139次播放

上新

08:12

风轻云淡——如何用好对象存储（下）

使用了OSS对象处理后，我们的图片处理工作可以不在业务服务器上进行

71次播放

全部视频

最热 最新

SQL 语句使用了索引，却还是慢查询？

林晓斌 腾讯云

6.1w11:53

会员免费

如何搭建一套高性能、高可用的 HBase 架构？

ElementsConsoleSourcesNetworkPerformanceMemoryApplicationRecorderLighthouseVueReduxsingle-spa Inspector

InspectorTimelineComponentsRoutesVuex

Find apps...

App 12.6.13App 22.6.13App 32.6.13App 42.6.13

Find components...

<Root>

<App> 30.4 ms

<HomeColumn> 28.3 ms

<HomeMobile> 28.3 ms

<TopNav> 27.7 ms

<HeaderTimeMobile> 27.4 ms

<Tab>

<PremiumVip> 26.2 ms

<DailyLessson> 93.3 ms

<Home> inactive 28.5 ms

<LearnPathList> inactive

<Dropdown> inactive

<Dropdown> inactive

<Dropdown> inactive

<Dropdown> inactive

<ColumnItem> key=0> inactive

<ColumnItem> key=1> inactive

<ColumnItem> key=2> inactive

<ColumnItem> key=3> inactive

<ColumnItem> key=4> inactive

<ColumnItem> key=5> inactive

<ColumnItem> key=6> inactive

<ColumnItem> key=7> inactive

<ColumnItem> key=8> inactive

<ColumnItem> key=9> inactive

<ColumnItem> key=10> inactive 14 ms

<ColumnItem> key=11> inactive 12.2 ms

<ColumnItem> key=12> inactive 11.2 ms

<ColumnItem> key=13> inactive

<ColumnItem> key=14> inactive

<ColumnItem> key=15> inactive

<ColumnItem> key=16> inactive

<ColumnItem> key=17> inactive

<ColumnItem> key=18> inactive

<ColumnItem> key=19> inactive

<GlobalLayer>

<TransitionGroup>

<DailyLessson> Filter state...

data

banner: Object

block_id: 88

block_name: "daily_lesson_banner"

block_title: "轮播图"

block_type: 1

list: Array[3]

nav_id: 0

page: Object

product_type: "b1"

currSwitchIndex: 0

isInApp: false

isShowCat: true

meeting: Object

block_id: 92

block_name: "cm_dl_con_1"

block_title: "技术大会"

block_type: 20

list: Array[4]

nav_id: 0

page: Object

product_type: "b20"

oldvip: Object

end_time: -1

status: -1

type: "d"

recommend: Object

block_id: 89

block_name: "editor_recommend"

block_title: "小编推荐"

block_type: 30

list: Array[3]

nav_id: 0

page: Object

product_type: "b30"

sortOrder: "hot"

switchOptions: Array[2]

0: Object

1: Object

topic: Object

block_id: 91

block_name: "topic_top"

block_title: "精选专题"

block_type: 31

list: Array[5]

nav_id: 0

page: Object

product_type: "b31"

调试路由

极客时间

课程训练营每日一课大厂案例

如何跨越技术鸿沟?

查看详情

最热

11:10

2022高效学习法：如何成为知识的主人？

怎么选择适合自己的内容去学习，怎么更好、更高效地掌握学习内容是我们要关注的问题

#个人成长

个人成长 · 5665次播放

上新

10:22

风轻云淡——如何用好对象存储（上）

云上存储已经是目前主流的存储之一，几乎所有的服务都要用到对象存储

#运维/测试

139次播放

上新

08:12

风轻云淡——如何用好对象存储（下）

使用了OSS对象处理后，我们的图片处理工作可以不在业务服务器上进行

71次播放

全部视频

最热 最新

6.1w11:53

为什么 SQL 语句使用了索引，但却还是慢查询？

林晓斌 腾讯云

会员免费

如何搭建 教学技能 高可用架构 MySQL 架构

ElementsConsoleSourcesNetworkPerformanceMemoryApplicationRecorderLighthouseVueReduxsingle-spa Inspector

InspectorTimelineComponentsRoutesVuex

Find apps...Search routes/column/:id(\d+)Filter state...

App 1	2.6.13	/ Index active
App 2	2.6.13	/library/:id LibraryDetail
App 3	2.6.13	/category Category
App 4	2.6.13	/resource Resource
		/download Download
		/download/extensions DownloadExtensions
		/search Search
		/column/intro ColumnIntroForPreview
		/live-column-intro/:id LiveColumnIntro
		/live/all LiveAll
		/column/comment/:id ColumnComment
		/column/:id(\d+) ColumnIntro
		/column/article ColumnArticle
		/column/article/preview ColumnArticlePreview
		/column/article/author-preview ColumnArticleAuthorPrev
		/column/article/:id(\d+) ColumnArticleDetail
		/column/article/comment Comment
		/article ArticlePreview
		/article/:id(\d+) Article
		/column/article/:id(\w+)/share Share
		/column/intro/:id(\d+) ColumnIntroDetail
		/recharge/login RechargeLogin
		/recharge/confirm RechargeConfirm
		/activity/recharge-2019 RechargeFor2019
		/activity/recharge/expand RechargeExpand
		/list/goupon/:id ListGoupon
		/activity/dl-generalize DailyLessonGeneralize
		/course/intro/:id CourseIntro
		/course/detail/:id-:vid CourseDetail
		/activity/promo Promo
		/activity/4th-2021 FourthBenefit2021
		/activity/11-2021 Double112021
		/cash/account CashAccount
		/cash/detail CashDetail
		/wxlite/groupbuy/:id(\d+) WxliteGroupBuy
		/wxlite/paid/:code WxliteGroupBuyPaid
		/wxlite/course/aliyunvideo/:cid-:aid WxliteCourseAliy
		/note/all NoteAll
		/note/all/v2 NoteAllV2
		/note/detail/:id NoteDetail
		/comment/all CommentAll
		/comment/detail/:id(\d+) CommentDetail
		/comment/nice/:id(\d+) NiceComment
		/comment/nice-module/:id(\d+) NiceCommentModule
		/discuss/detail/:id(\d+) DiscussDetail
		/discuss/write DiscussWrite
		/activity/wxlite/groupbuylist WxliteGroupBuyList
		/reply Reply

调试 Vuex

极客时间

课程训练营每日一课大厂案例

10 分钟解决一个技术难题

点击查看

最热



11:10

2022高效学习法：如何成为知识的主人？

怎么选择适合自己的内容去学习，怎么更好、更高效地掌握学习内容是我们要关注的问题

#个人成长

个人成长 · 5665次播放

上新



10:22

风轻云淡——如何用好对象存储（上）

云上存储已经是目前主流的存储之一，几乎所有的服务都要用到对象存储

#运维/测试

139次播放

上新



08:12

风轻云淡——如何用好对象存储（下）

使用了OSS对象处理后，我们的图片处理工作可以不在业务服务器上进行

71次播放

全部视频

最热 最新



为什么 SQL 语句使用了索引，但却还是慢查询？

林晓斌 腾讯云

6.1w 11:53

会员免费

如何搭建 教学技能 高可用架构 14 SQL 架构

ElementsConsoleSourcesNetworkPerformanceMemoryApplicationRecorderLighthouseVueReduxsingle-spa Inspector

Inspector

Timeline

Components

Routes

Vuex

Find apps...

Filter stores...

Root

Filter state...

App 12.6.13

App 22.6.13

App 32.6.13

App 42.6.13

Root

state

nowTime: 1648397645000

loginUser: Object

universityUser: false

pvip: Object

end_time: 1678982398

status: 1

type: "p"

hasPvipFirstPromo: false

pvipPriceInfo: Object

monthEveryDayPrice: "1.87"

monthFirstPrice: 6

monthMarketPrice: 99

monthSalePrice: 58

yearEveryDayPrice: "1.36"

yearEveryDayPriceOneFloat: "1.3"

yearFirstPrice: 0

yearMarketPrice: 699

yearSalePrice: 49800

globalLayers: Array[0]

mobile: true

nvcActivated: false

Vue3 生态下，我们能做些什么？

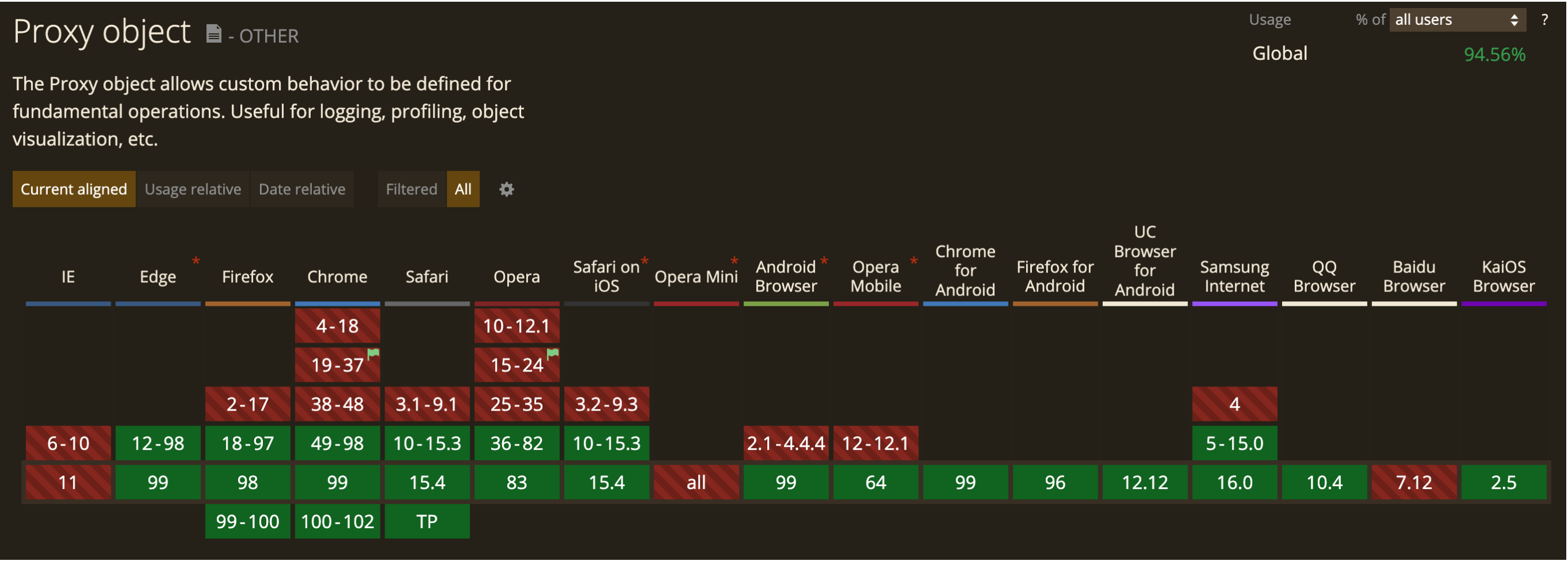
赋能业务开发的工具：

- UI 组件库: PC、移动端（将公司内现有的 Vue2 组件库进行升级）
- 高可复用的 Composition API 库（基于 VueUse）
- 业务型脚手架（在现有开源脚手架的基础上封装业务常用功能，npm 包、vscode 插件均可，内置功能尽可能做到可插拔）
- 业务型物料组件

核心原则：为业务开发提效但不要重复造轮子 !!!

1. 基于开源工具实现业务属性扩展
2. 不同部门的物料、工具要能共享

关于兼容性



- IOS >= 10
- Android >= 5
- IE 不支持

COMING SOON... 提前学习，提前踩坑，提前布局

种一棵树最好的时机是十年前，其次是现在。

Vue3 学习资料推荐

Vue3 learning resource recommend

- 书籍: 《Vue.js 设计与实现》
- 在线课程: 《极客时间-玩转 Vue3 全家桶》
- 新文档: <https://staging-cn.vuejs.org/>
- Awesome Vue: <https://github.com/vuejs/awesome-vue>
- Vue3 源码: <https://github.com/vuejs/core>
- 优质博客: <https://antfu.me/posts>
- 优质 Vue3 组件库: [移动端 Vant](#)、[PC端 NaiveUI](#)
- 优质起手模板: <https://github.com/antfu/vitesse>
- Vue3 生态链: Pinia、Vitest、VueRouter4...



作业

HomeWork

Hxmui-Vue3

挑选下列 `5` 个组件之一，根据交互需求描述及UI稿使用 Vue3 进行开发

- Loading 加载
- Toast 轻提示
- Dialog 对话框
- Cell 单元格
- Layout 布局

作业提交形式：在仓库的主分支中拉取代码，以 `feat-姓名` 的形式创建分支，组件开发完成后提交 pull request 到主分支上。

作业提交时间：xx 月 xx 日前