



typescript学习心得分享

by 邓佳杰
企业办公事业处



纲要

01. 价值所在

02. 配置文件

03. 声明空间

04. 语法有哪些区别

05. 声明文件

06. 最佳实践之一

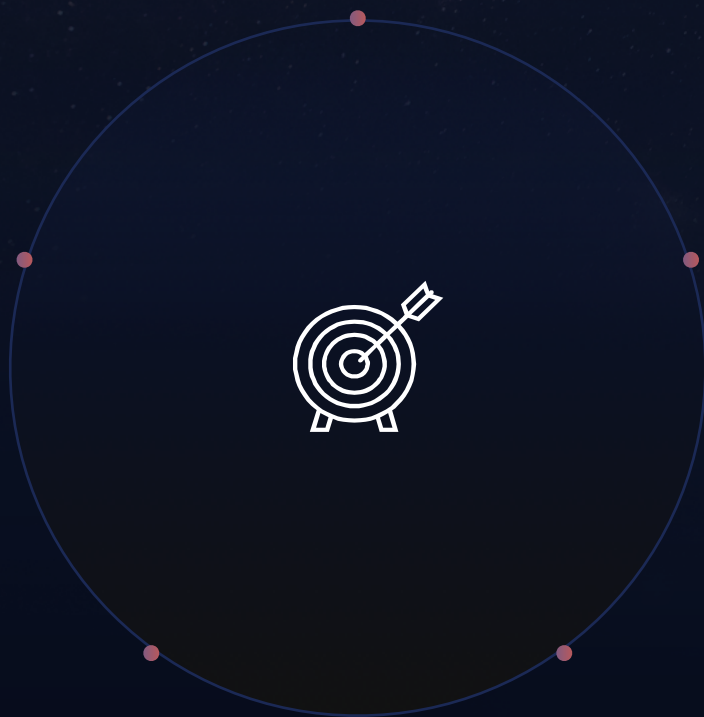


01

价值所在

The Value

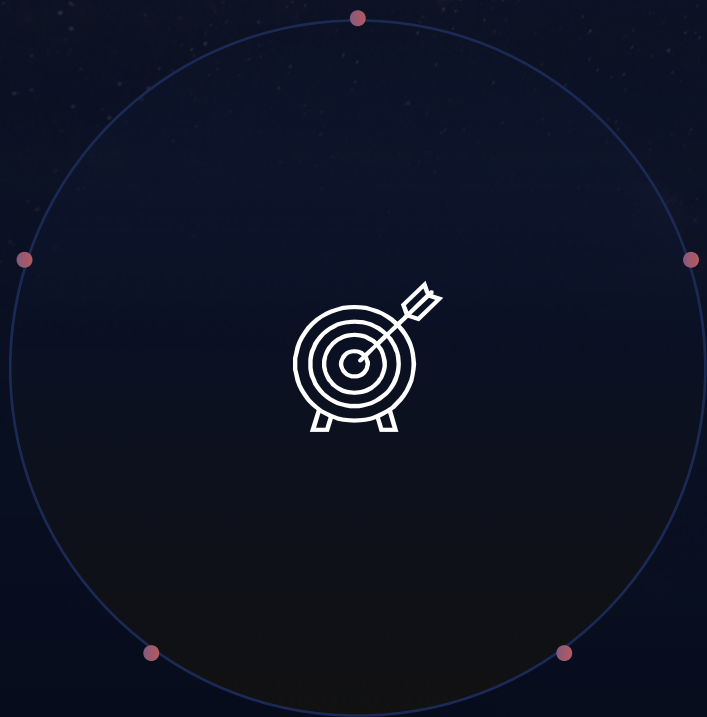
从问题场景出发 寻找答案



问题1：语法太灵活导致开发大型 Web 项目困难

性能不能满足一些场景的需要

业界大佬们的尝试



微软: typescript. 加入类型检查, 提高代码健壮性

谷歌: Dart。为浏览器引入新的虚拟机去直接运行 Dart 程序以提升性能

火狐的: asm.js。取 JS 的子集, JS 引擎针对 asm.js 做性能优化

WebAssembly

ts优缺点

优点

1. 类型检查
2. 语言扩展
3. 帮助团队重塑“类型思维”

缺点

1. 会增加一定开发成本



02

配置文件

一、tsconfig.json

是什么：

指定了用来编译这个项目的根文件和编译选项（ts编译的配置文件），不带任何输入文件的情况下调用tsc，编译器会从当前目录开始去查找tsconfig.json文件，逐级向上搜索父目录。tsconfig.json可以是空文件，空文件时用默认配置。

一、tsconfig.json

主要配置项

1. “files”、“include”、“exclude”: files指定的文件总会被包含在内, include指定的可被exclude排除。
2. compilerOptions
 - outDir: 输出目录。指定的目录下的文件永远会被编译器排除,除非用files指定.
 - sourceMap: 编译成js时是否生成map文件
 - noImplicitAny: 是否允许隐式any
 - baseUrl: 解析非相对模块名的基准目录
 - target: 目标代码类型
 - lib: 编译过程中需要引入的库文件的列表
 - strict: true, 启用所有严格类型检查选项
 - jsx: 指定jsx代码的生成, 三种模式(preserve、react、react-native)
 - types: 需要包含的类型声明文件名列表,一般不用指定



03

声明空间

一、声明空间

类型声明空间：包含用来当做类型注解的内容，不能当做变量来用

```
class Foo {}  
interface Bar {}  
type Bas = {};
```

变量声明空间：包含可用作变量的内容

```
class Foo {}  
const someVar = Foo;  
const someOtherVar = 123;
```



04

语法有哪些区别

一、数据类型

基础类型

1. 元组 Tuple: 一个已知元素数量和类型的数组，各元素的类型不必相同
2. 枚举(enum)
3. 任意值 (any): 关闭类型检查
4. 空值(void): 表示没有任何类型，一个函数没有一个返回值
5. never: 永远不存在的值的类型。（1. 从来不会有返回值的函数 2. 总是抛错的函数）

一、数据类型

高级类型

1. 联合类型
2. 交叉类型
3. 索引类型 (keyof 索引类型查询操作符)
4. 映射类型

二、类

1. 静态方法、静态属性 (es)
2. 三种访问修饰符(业务中一般不需要使用)
 1. public: 在任何地方都可以访问到
 2. private: 不能在声明它的类的外部访问, 不能再子类中访问
 3. protected: 修饰的属性或方法受保护, 在子类中允许被访问
3. 抽象类 (abstract 加class前, abstract 加在方法前)
 1. 不允许被实例化
 2. 抽象类的抽象方法必须被子类实现

三、接口

1. 对类的一部分行为进行抽象 (implements)
2. 对对象的形状进行描述 (可选属性、任意属性 (添加字符串索引签名)、只读属性)
3. 接口可继承另一个接口, 可同时继承多个接口, 接口可继承类
4. 描述函数, 描述类

四、函数

1. 可选参数

2. 函数重载

五、泛型 (generics)

1. 指在定义函数、接口或类的时候，不预先指定具体的类型，而在使用的时候再指定类型的一种特性（就是在使用的时候可以传入一个类型参数）
2. 泛型函数、泛型接口、泛型类
3. 定义泛型时可一次定义多个类型参数一般用T、U、V表示，多个的时候也可以更语义化但通常以T作为前缀
4. 泛型约束
5. 常见的工具泛型: Readonly、Partial、



05

声明文件

一、声明文件（.d.ts）

1. 为全局变量定义类型

2. declare var（声明全局变量）、declare function（声明全局方法）、declare class（声明全局类）、declare enum（声明全局枚举类型）、declare namespace、declare module（声明（含有子属性的）全局对象）、interface 和type 声明全局类型

3. 全局模块 and 文件模块：如果在你的 TypeScript 文件的根级别位置含有 import 或者 export，它会在这个文件中创建一个本地的作用域



06

最佳实践之一

看一下umi + antd + ts 的写法

第一步：创建一个文件夹，并进入

第二步： `npm create umi` 或者 `yarn create umi`

第三步： 根据操作选择 `antd + ts`



THANK YOU & QA