

Hook 在不同开发框架中的重要性

...

...

...

REACT.memo 缓存了一个函数组件，接受普通值作为参数，实现类组件 PureComponent 的功能；

而 REACT.callback 钩子缓存了一个函数，函数是引用类型变量，作为属性时

REACT.memo 缓存不了，因为每次重新渲染都会产生一个新的函数；

REACT.memo 缓存一个函数的返回值，接受依赖项。

补充：Proxy 对象

使用方式：Proxy (target, handler)

Proxy 对象是对 target 对象的一种包装器，可以劫持一些对 target 对象的属性的操作，由 handler 定义其中对劫持属性的操作，如果不赋值 handler，则 Proxy 对象可以理解为一层透明保护膜，常见的捕获器如下图：

内部方法	Handler 方法	何时触发
[[Get]]	get	读取属性
[[Set]]	set	写入属性
[[HasProperty]]	has	in 操作符
[[Delete]]	deleteProperty	delete 操作符
[[Call]]	apply	函数调用
[[Construct]]	construct	new 操作符
[[GetPrototypeOf]]	getPrototypeOf	Object.getPrototypeOf
[[SetPrototypeOf]]	setPrototypeOf	Object.setPrototypeOf
[[IsExtensible]]	isExtensible	Object.isExtensible
[[PreventExtensions]]	preventExtensions	Object.preventExtensions
[[DefineOwnProperty]]	defineProperty	Object.defineProperty, Object.defineProperties
[[GetOwnProperty]]	getOwnPropertyDescriptor	Object.getOwnPropertyDescriptor, for...in, Object.keys/values/entries
[[OwnPropertyKeys]]	ownKeys	Object.getPrototypeOfNames, Object.getPrototypeOfSymbols, for...in, Object.keys/values/entries

注意 ⚠：Proxy 代理和函数包装器的区别在于，Proxy 对任何东西的劫持都会将转发到目标对象，代理上的所有操作都能被转发到原始函数，如果存在劫持会劫持操作而已。

讨论：ES6 模块与 CommonJS 模块之间的差别

三个主要差异：

1. CommondJS 模块输出的是一个值的拷贝，而 ES6 模块输出的是值的引用；
2. CommandJS 模块是运行时加载，而 ES6 模块是编译时输出接口；
3. CommandJS 模块的 require() 时同步加载的，而 ES6 的 import 模块是异步加载的，又一个独立的模块依赖的解析阶段；

主要差异原因：

1. CommondJS出现较早，由nodeJS实现的模块化方案，实现引入拷贝，将引入对象作为一个对象缓存下来module.exports，并且是同步加载的；
2. ES6后续出现，提出了新的模块化方案，动态加载，异步加载，需要时去指定模块读取，不同观念的设计方案，ES6 模块不是对象，它的对外接口只是一种静态定义，在代码静态解析阶段就会生成。

(补充一点：npx命令来自于npm，可以调用内部已经安装的模块)
(npx命令也可以下载到一个临时目录，使用以后再删除create-app)

rollup.js 学习 一个打包 ES6 模块文件的工具