02

OPEN ORIENTED

凹 凸 实 验 室

Symbol

1

介绍

Symbol 是什么

02

第 7 种基础数据类型

Symbol 怎么创建

02

- Symbol([desc])
- Symbol.for([desc])

全局 Symbol 怎么获取

Symbol.keyFor(desc)

Symbol 作为 Object Keys

- 不可被 Object.keys、Object.getOwnPropertyNames 、for…in 遍历
- 可以被 Object.getOwnPropertySymbols、Reflect.ownKeys 遍历
- 能被 Object.assign 赋值

2

玩法

作为枚举数据


```
const logLevel = {
  DEBUG: 'debug',
  INFO: 'info',
  WARN: 'warn'
}
```

```
function log (level, msg) {
  switch (level) {
    case logLevel.DEBUG:
      return console.debug(msg)
    case logLevel.INFO:
      return console.info(msg)
    case logLevel.WARN:
      return console.warn(msg)
    default:
      return
  }
}
```

作为枚举数据


```
const logLevel = {
  DEBUG: Symbol(‘debug’),
  INFO: Symbol('info'),
  WARN: Symbol('warn')
}
```

```
function log (level, msg) {
  switch (level) {
    case logLevel.DEBUG:
      return console.debug(msg)
    case logLevel.INFO:
      return console.info(msg)
    case logLevel.WARN:
      return console.warn(msg)
    default:
      return
  }
}
```

作为弱私有属性

并不是真正意义上的私有

为提供的 API 提供勾子


```
function foo (o) {
  if (typeof o[foo.Symbols.INSPECT] === 'function') {
    // 传入的对象实现了钩子，就按钩子逻辑走
  } else {
    // 按正常逻辑走
  }
}
```

内置 Symbol

- Symbol.hasInstance
- Symbol.iterator
- Symbol.isConcatSpreadable
- Symbol.unscopables
- Symbol.match
- Symbol.replace
- Symbol.search
- Symbol.split
- Symbol.species
- Symbol.toPrimitive
- Symbol.toStringTag

Symbol.iterator

for of

- obj[Symbol.iterator] = function* () {
  yield xxx
  }

- class A () {
  *[Symbol.iterator] () {
  yield xxx
  }
  }

Metaprogramming

- Symbol
- Reflect
- Proxy

THANKS

FOR YOUR WATCHING