

06.toRef 和 toRefs

学习要点：

1.toRef 和 toRefs

本节课我们来了解一下 Vue3.x 中的 ref 两个周边 API 的用法；

一. toRef 和 toRefs

1. toRef 可以将源响应式对象上的 property 创建一个 ref 对象；

```
const obj = reactive({
  name : 'Mr.Lee',
  age  : 100
})

// ref 和 toRef 都是给一个静态数据实行响应式，转换成 ref 对象
// ref 是复制操作，和之前的对象断了关联；
// toRef 是引用操作，当修改数据时，原数据也会被修改；
const nameRef = ref(obj.name)
const nameToRef = toRef(obj, 'name')

console.log(nameRef)
console.log(nameToRef)

nameRef.value = 'Mr.Wang'
nameToRef.value = 'Mr.Wang'

console.log(obj.name)
```

PS: 手册上使用的 prop 传递，因为我们还没有学 setup(prop)，我们换个例子；

```
const obj = reactive({
  name : 'Mr.Lee',
  age  : 100
})

const info = {
  // 响应式操作，使用 ref()，则无法获取原数据的修改信息
  name : toRef(obj, 'name')
}

setTimeout(() => {
  // 由于使用了 toRef，原数据修改后，这里会更新到最新数据
  console.log(info.name.value)
}, 1000)
```

```
// 修改原数据，关联被 toRef 的数据  
obj.name = 'Mr.Wang'
```

2. `toRefs` 会将原来响应式对象转换为普通对象，并将内部所有 `property` 都 `ref`;
3. 比如：我们要把一个函数里响应式对象里的 `property` 解构出来时还保持响应式;

```
const userF = () => {  
  const state = reactive({  
    foo : 1,  
    bar : 2  
  })  
  
  return toRefs(state)  
}
```

```
const { foo, bar } = userF()
```

```
console.log(foo)
```

```
console.log(bar)
```