

An abstract composition on a dark gray background. It features several diagonal lines: a white line in the upper left, a light gray line in the lower left, and a bright pink line in the lower right. The word 'Substrate' is written in a bold, orange, sans-serif font, positioned between the white and light gray lines. The word 'River' is written in the same font and color, positioned between the light gray and pink lines.

Substrate


River



Basic-002

Storing a Storage Value

<https://substrate.dev/substrate-collectables-workshop/#/1/storing-a-value>



新建一个函数: decl_module!



函数的结构

```
fn foo(origin, bar: Bar, baz: Baz, ...) -> Result{  
}
```

origin: module中的第一个参数一定是

Result: 是support::dispatch中的类型

<https://substrate.dev/docs/en/overview/glossary#origin>



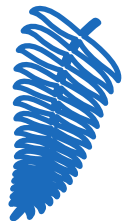
origin

- 1- 由外部帐户签署的Pub calls
- 2- 只允许由治理系统进行的Root calls
- 3- 仅允许块作者和验证器进行Inherent calls



检查Signed Message

- 1- ensure_signed
- 2- ensure_root
- 3- ensure_inherent



use system::ensure_signed;



Result

```
enum Result<T, E> {  
    Ok(T), //成功的返回  
    Err(E), //错误的返回  
}
```



Remember:

- 1- MUST NOT PANIC:在任何情况下(可能除了存储进入不可修复的损坏状态之外), 此函数都不会出现panic。
- 2- NO SIDE-EFFECTS ON ERROR:这个函数必须完全完成并返回Ok(), 或者它必须对存储和返回Err('Some reason')没有副作用。

https://substrate.dev/rustdocs/v1.0/srml_support/dispatch/result/index.html



例子

```
// Add these imports:
//
// use support::{dispatch::Result, StorageValue};
// use system::ensure_signed;
decl_module! {
    pub struct Module<T: Trait> for enum Call where origin: T::Origin
    {

        fn my_function(origin, input_bool: bool) -> Result {
            let _sender = ensure_signed(origin)?;

            <MyBool<T>>::put(input_bool);

            Ok(())
        }
    }
}
```



备注

```
// 视频中没有解释StorageValue和ensure_signed的导入  
//  
// use support::{dispatch::Result, StorageValue};  
// use system::ensure_signed;
```

set_function 在前端显示为setFunction;
做了一个表示的转换



Substrate

River

Thanks