

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**

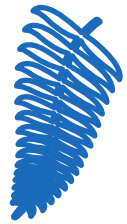
**River**



**Basic-002**

# **Creating a Storage Value**

<https://substrate.dev/substrate-collectables-workshop/#!/1/creating-a-storage-value>



声明Storage Value: decl\_storage!



Rust 原始类型

bool, u8, u32...




Substrate自定义类型

AccountId, BlockNumber, Hash ...

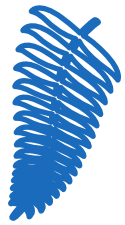


例子



```
decl_storage! {  
  trait Store for Module<T: Trait> as Example {  
    MyU32: u32;  
    MyBool get(my_bool_getter): bool;  
    KittyCount get(kitty_count): u32;  
  }  
}
```

为了存储storage values, 需要引入support::StorageValue



## 使用Storage Value

[https://substrate.dev/rustdocs/v1.0/srml\\_support/storage/trait.StorageValue.html](https://substrate.dev/rustdocs/v1.0/srml_support/storage/trait.StorageValue.html)



```
/// Get the storage key.
fn key() -> &'static [u8];

/// true if the value is defined in storage.
fn exists<S: Storage>(storage: &S) -> bool {
    storage.exists(Self::key())
}

/// Load the value from the provided storage instance.
fn get<S: Storage>(storage: &S) -> Self::Query;

/// Take a value from storage, removing it afterwards.
fn take<S: Storage>(storage: &S) -> Self::Query;

/// Store a value under this key into the provided storage instance.
fn put<S: Storage>(val: &T, storage: &S) {
    storage.put(Self::key(), val)
}

/// Mutate this value
fn mutate<R, F: FnOnce(&mut Self::Query) -> R, S: Storage>(f: F, storage: &S) -> R;

/// Clear the storage value.
fn kill<S: Storage>(storage: &S) {
    storage.kill(Self::key())
}
```



例子



PUT:

```
MyU32::put(1337);
```

GET:

```
let my_bool = MyBool::get();  
let also_my_bool = Self::my_bool_getter();
```



**Substrate**

**River**

**Thanks**