

# Mapping Object

## 1b - MappingObject的类代码加载到前端

```
MappingObject类型- {key: "(MappingObject)|||Foo|||{value:123}"}

RenderWrap.load(Task.load("RenderWrap::MappingObject"))
RenderWrap.load(Task.load("DemoFoo::Foo"))
```

## 5 - jsrb\_undata(data) 函数

```
obj = (Object.const_get class_name).new
# 等价于 obj = Foo.new, 动态语言特性

obj.from_data(obj_data)
```

后端

前端

DB



```
Class Foo < MappingObject
  mapping_accessor :value

  def calc_value()
    return value**2
  end
end

foo=Foo.new()
foo.value = load_data_from_db()
RenderWrap["foo"] = foo
RenderWrap.data
```

HTML



```
<pre id='data-transfer'
style='display:none'>
  xxxxxxxxxxxxxxxxxxxxxx
</pre>

<script>
  class MappingObject
  end

  class Foo < MappingObject
  end
</script>

<script>
  $data = jsrb_undata($document.at_css('#data-
transfer').text)
</script>
```



```
$data["foo"].calc_value
```

# Render Wrap

- RenderWrap: 在html/jsrb前后包上代码

```
RenderWrap.html = "<h1>hello world<h1>"  
RenderWrap.jsrb = "puts 'this is console log' "
```

