

# theme('hook', \$arguments)

- ← [ 载入主题注册表 ]
- ← [ 检查挂钩的 \* 通配符 ]

## 如果挂钩是函数

- (f.a) `themeName_hook($arguments)` |-----> `template.php`
- (f.b) `engineName_hook($arguments)` |-----> `.engine`
- (f.c) `theme_hook($arguments)` |-----> `.module/.inc`  
[ 默认执行为函数 ]

## 如果挂钩是模板

- ← [ 把参数 (\$arguments) 转换为 变量 ]
- ← [ 检查 渲染函数, 默认为 `theme_render_template` ]
- ← [ 检查通过文件扩展名指定的函数并且执行, 默认为 `".tpl.php"` ]

### 预处理函数

[ 设定 变量 和 建议处理 ]

每个预处理函数之间, 变量都是通过引用来传递。因此, 变量一般都要包装成变量数组, 作为一个集合向下传递。

- ← (t.a) `template_preprocess` |-----> `theme.inc`
- ← (t.a) `template_preprocess_hook` |-----> `.module/.inc`
- ← (t.b) `moduleName_preprocess` |-----> `.module`
- ← (t.b) `moduleName_preprocess_hook` |-----> `.module`
- ← (t.c) `engineName_engine_preprocess` |-----> `.engine`
- ← (t.c) `engineName_engine_preprocess_hook` |-----> `.engine`
- ← (t.d) `engineName_preprocess` |-----> `template.php`
- ← (t.d) `engineName_preprocess_hook` |-----> `template.php`
- ← `themeName_preprocess` |-----> `template.php`
- ← `themeName_preprocess_hook` |-----> `template.php`

- ← [ 收集 处理建议 ]
- ← `drupal_discover_template`  
[ find implemented suggestion ]

`theme_render_template`

[ assemble .tpl.php file ]

suggestion

default hook

```
suggestion.tpl.php
<html>
  <?php print $variable ?>
</html>
```

```
hook.tpl.php
<html>
  <?php print $variable ?>
</html>
```