

帮助方法:

1. 行内helper: 最终输出只是字符串的helper

声明helper: `hbs.registerHelper('addOne', function (num){ return xx;})`

使用helper: `{{addOne arg1}}`

常见使用场景: 数据格式化

2. 块级helper: 最终输出是html的helper

范例: 扩展代码extend

3. N多帮助方法库<https://github.com/helpers/handlebars-helpers>

```
const helpers = require('handlebars-helpers');//helpers库
helpers.comparison({handlebars: hbs.handlebars})
```

部分视图 - partial (每次编写新的partial, 需要重启)

1. 注册partial目录

```
hbs.registerPartials(path.join(__dirname, '../views/partials'))
```

2. 使用partial

```
{{> nav}}
```

3. 动态partial

```
<!--动态partial: whichPartial是一个helper-->
{{> (whichPartial navName)}}
```

4. 块级部分视图 partial-block

```
{{#> oox}}
  出现错误时, 您会看到这句话
{{/oox}}

<!--使用块级partial传递内容-->
{{#> win}}
  <div>这是窗口里面的内容。。。。。。。。。。。。。。。。。 </div>
{{/win}}

// 声明布局partial
<div class="top">
  {{> top}}
</div>

// 使用布局partial
```

```
{{#> layout}}
  {{#*inline 'top'}}
    这是top
  {{/inline}}
  {{#*inline 'foo'}}
    这是foo
  {{/inline}}
  {{#*inline 'bar'}}
    这是bar
  {{/inline}}
{{/layout}}
```

5. 作业:

1. 请使用动态视图的方法实现导航栏根据用户登录状态渲染
2. 能否将win组件的title作为参数传递并动态显示
3. 复习今日内容
4. 预习mysql workbench使用
5. 预习sql语法
6. 预习node-mysql使用方法, [官网](#)