

67. 缓存功能

学习要点：

1. 缓存功能

本节课我们来学习一下系统提供的缓存功能，实现缓存的写入和读取。

一. 缓存功能

1. 系统内置了很多类型的缓存，除了 **File**，其它均需要结合相关产品；
2. 我们这里主要演示 **File** 文本缓存，其它的需要学习相关产品；
3. 在 **app.php** 中，配置文件 **cache.php** 进行缓存配置，可以用 **::init()**；

```
// 驱动方式
'type' => 'File',
// 缓存保存目录(默认 runtime/cache)
'path' => '',
// 缓存前缀
'prefix' => '',
// 缓存有效期 0 表示永久缓存
'expire' => 0,
```

4. **::set()**方法，可以设置一个缓存，然后再 **runtime/cache** 查看生成结果；
`Cache::set('user', 'Mr.Lee');`

5. **::inc()**和**::dec()**实现缓存数据的自增和自减操作；

```
Cache::inc('num');
Cache::inc('price', 3);

Cache::dec('num');
Cache::dec('price', 3);
```

6. **::has()**方法，判断缓存是否存在，返回布尔值；

```
Cache::has('user');
```

7. **::get()**方法，从缓存中获取到相应的数据，无数据返回 **null**；

```
Cache::get('user');
```

8. **::delete()**方法，可以删除指定的缓存文件；

```
Cache::rm('user');
```

9. **::pull()**方法，先获取缓存值，然后再删除掉这个缓存，无数据返回 **null**；

```
Cache::pull('user');
```

10. `::clear()`方法，可以清除所有缓存；

```
Cache::clear();
```

11. `::tag()`标签，可以将多个缓存归类到标签中，方便统一管理，比如清除；

```
Cache::tag('tag')->set('user', 'Mr.Lee');
```

```
Cache::tag('tag')->set('age', 20);
```

```
Cache::set('user', 'Mr.Lee');
```

```
Cache::set('age', 20);
```

```
Cache::tag('tag', ['user', 'age']);
```

```
Cache::clear('tag');
```

12. 助手函数的使用：`cache()`；

```
//设置缓存
```

```
cache('user', 'Mr.Lee', 3600);
```

```
//输出缓存
```

```
echo cache('user');
```

```
//删除指定缓存
```

```
cache('user', null);
```