

2-22页面静态化与总结

2-22页面静态化与总结	1
1. 课堂案例	2
课堂讲解	2
一、上节回顾	2
两张表的关联操作	2
数据的分组	2
二、学习目标	2
掌握静态化技术	2
三、教学过程描述	2
关于web页面	2
PHP页面静态化应用场景	2
PHP静态化	2
四、小结	6
2. 课堂练习	6
3. 课后练习	6
4. 资料扩展	6
伪静态化	6
rewrite	6

1. 课堂案例

课堂讲解

一、上节回顾

两张表的关联操作

数据的分组

二、学习目标

掌握静态化技术

三、教学过程描述

关于web页面

动态页面

动态页面一般是通过PHP、jsp、asp来实现的

PHP文件执行阶段：

语法分析--->编译--->运行--->展示出来

静态页面

纯静态的html文件

不经过语法分析和编译, 直接运行--->展示出来

PHP页面静态化应用场景

内容不经常改变的页面, 比如新闻首页、新闻详情页、商城首页、商品详情页

PHP静态化

是把PHP生成的动态页面保存成静态的html文件, 用户访问该静态页面, 而不是用户每一次访问都重新生成一张相同的网页, 优点就是减小服务器开销

1、全部纯静态化

整个文件是纯静态的html文件

1-1、全部纯静态化应用到的知识点

1、file_put_contents()

2、file_get_contents()

3、buffer方面的ob系函数

ob_start — 打开输出控制缓冲

ob_get_clean — 得到当前缓冲区的内容并删除当前输出缓冲。

ob_get_contents — 返回输出缓冲区的内容

ob_clean — 清空(擦掉)输出缓冲区

ob_end_clean — 清空(擦除)缓冲区并关闭输出缓冲

2、局部静态化

加载整个文件是纯静态的html文件, 加载完毕后, 通过ajax局部更新页面的内容

2-1、局部静态化应用到的知识点

1、file_put_contents()

2、file_get_contents()

3、buffer方面的ob系函数

ob_start — 打开输出控制缓冲

ob_get_clean — 得到当前缓冲区的内容并删除当前输出缓冲。

ob_get_contents — 返回输出缓冲区的内容

ob_clean — 清空(擦掉)输出缓冲区

ob_end_clean — 清空(擦除)缓冲区并关闭输出缓冲

4、ajax技术

```
<script type="text/javascript">
```

```
$.ajax({  
    type: 'POST',  
    url: "",  
    dataType: 'json',
```

```

        data:{data},
        success: function(data){
            // 成功以后代码走这里
        },
        error:function(data) {
            // 失败以后代码走这里
        },
    });
</script>
<script type="text/javascript">
    $.post('url',{data},function(data){
        //方法执行成功后, 代码才会走到这里
    });
</script>
<script type="text/javascript">
    $.get('url',{data},function(data){
        //方法执行成功后, 代码才会走到这里
    });
</script>

```

3、伪静态化

URL是一个PHP模式, 通过正则或者服务器的配置, 让URL看上去是静态的HTML地址

例如:

http://rulai.com/index.php?m=admin&c=news&a=news_detail&id=50

通过伪静态:

<http://rulai.com/index/50.html>

3-1、php正在表达式的方式实现伪静态

例如静态url:index.php/news/1.html

通过正在表达式去分析伪静态URL

http://rulai.com/index.php?m=admin&a=news&c=news_detail&id=1

<http://rulai.com/index.php/1.html>

```
print_r($_SERVER['PATH_INFO'])
```

备注:nginx服务器默认情况下不支持path_info模式, 需要配置

```
l=>id=1
```

静态地址--->其实获取的是动态的内容

3-2、通过WEB服务器rewrite配置来实现伪静态

apache下rewrite配置

1、虚拟域名配置

在服务器开启以下配置:

开启: LoadModule rewrite module modules/mod_rewrite.so

开启: Include conf/extra/httpd_vhosts.conf

2、httpd_vhosts.conf配置未见配置相关信息

下面配置放到域名下面

```
RewriteEngine on
```

```
//当访问的地址存在该文件或者目录的话则访问存在的
```

```
RewriteCond %{DOCUMENT_ROOT}%{REQUEST_FILENAME}!-d
```

```
RewriteCond %{DOCUMENT_ROOT}%{REQUEST_FILENAME}!-f
```

```
//正则表达式
```

```
RewriteRule ^/detail/([0-9]*).html$/detail.php?id=$1
```

nginx下rewrite配置

```
rewrite ^/detail/(\d+).html$/detail.php?id=$1 last;
```

```
nginx > cd /etc/nginx/conf.d
```

```
nginx > ll
```

```
nginx > vim static.singwa.com.conf
```

在这个中间插入正则表达式

```
location /{
```

```

if(!-e $request_filename) {
    //正则表达式写在这里
    rewrite ^/detail/(\d+).html$/detail.php?id=$1 last;

    break;
}
}

```

四、小结

- 1、在项目中其实联表功能是很少使用的, 因为你的表如果设计的不好就会导致很多问题的发生
- 2、在数据大的时候联表查询反而没分开查询的快

2. 课堂练习

掌握静态化技术

3. 课后练习

完善前台新闻详情的静态化

4. 资料扩展

伪静态化

美观;便于搜索引擎收录

rewrite

Apache服务器rewrite配置

在httpd.conf文件中, 找到

```

#注释：去掉前边的" # "开启rewrite服务，重启服务器生效
#LoadModule rewrite_module modules/mod_rewrite.so

#注释：http-vhosts.conf文件是虚拟域名配置的文件，开启改文件可以配置虚拟域名，一般默认是开启的
#Include conf/extra/httpd-vhosts.conf

```