* 文件系统
  + 文件类型
    - Windows:
      * Dir
      * File
      * unknown
    - Linux:
      * 7种
    - 对于类型的检测：
      * Is\_xxx()
      * 返回bool
  + 路径：
    - 相对路径
      * ./:表示当前目录，只能玩一次。
      * ../：表示上级目录，可以多次使用
    - 绝对路径
      * 网络绝对路径：url
      * 本地绝对路径：E:\\xxx\xx\xx
      * 几乎所有关于路径信息的操作，返回的结果都是本地绝对路径
  + 目录
    - 目录句柄操作
      * Opendir()
      * Readdir()
      * Closedir()
      * Rewinddir()
    - 目录信息获取
      * Getcwd()
      * Basename()
      * Dirname()
      * Pathinfo()
      * Realpath()
    - 目录新建&删除
      * Mkdir()
      * Rmdir()
    - 目录检测：
      * File\_exists()
      * Is\_dir()
    - 磁盘空间统计
      * Disk\_free\_space()
      * Disk\_total\_space()
      * 不论给的是文件还是目录，返回的都是所在磁盘的空间大小
      * 返回的是字节数，如果想要直观显示，需要自行转换
  + 文件
    - 文件大小统计
      * Filesize()
    - 文件的删除
      * Unlink()
    - 时间相关的
      * Filectime():创建时间
      * Fileatime()：访问时间
      * Filemtime()：修改时间
      * 注意：返回的都是时间戳
      * 访问时间和创建时间在windows下一样的

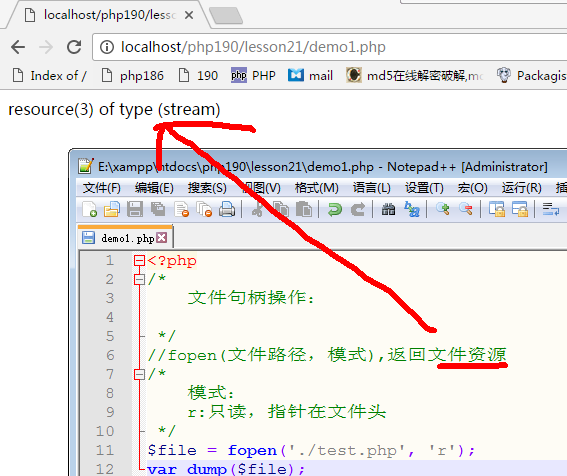
# 目录系统

## 文件

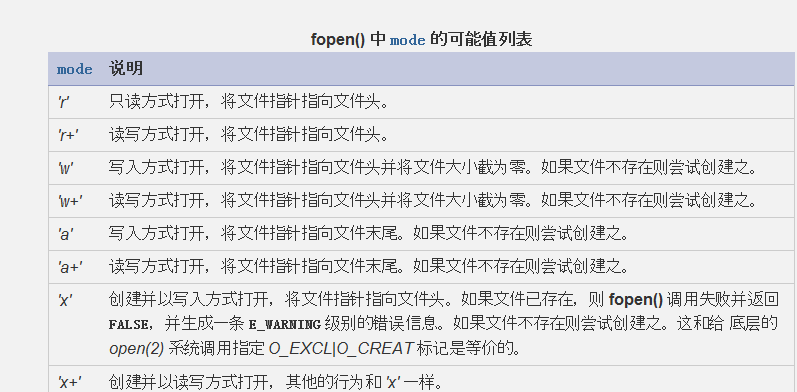
### 文件的句柄操作

#### Fopen(‘文件路径’, 模式)

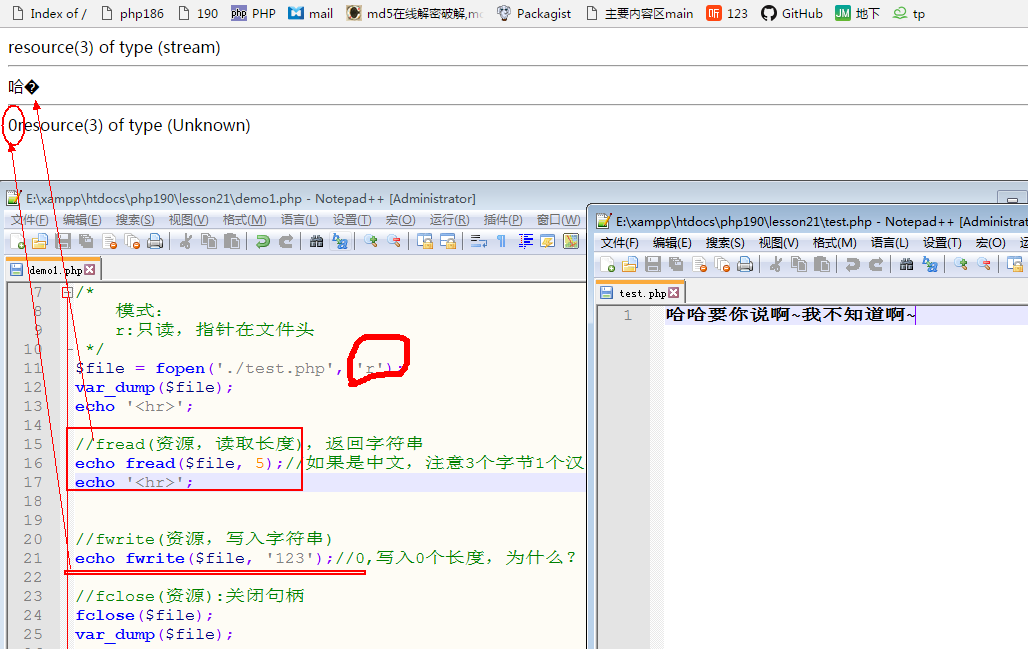




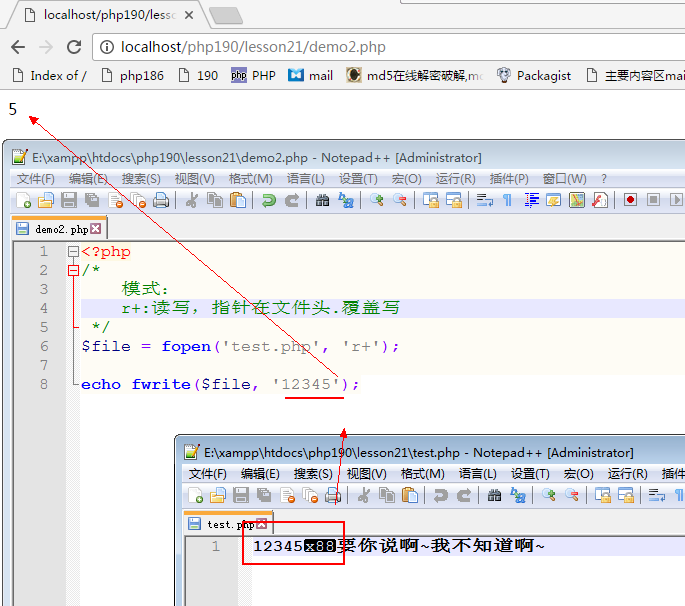
* 常见的模式



* **R:**
  + **只读，指针在文件头**

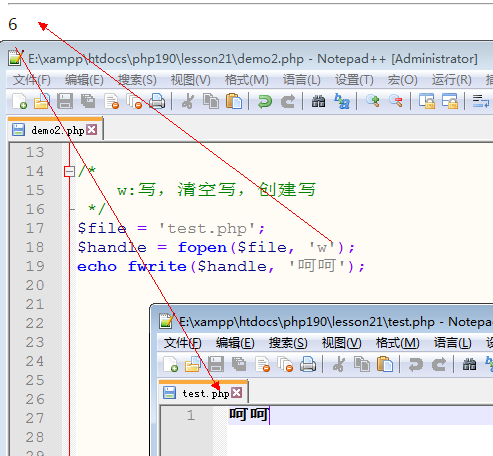


* **R+:**
  + **读写，指针在文件头，覆盖写**

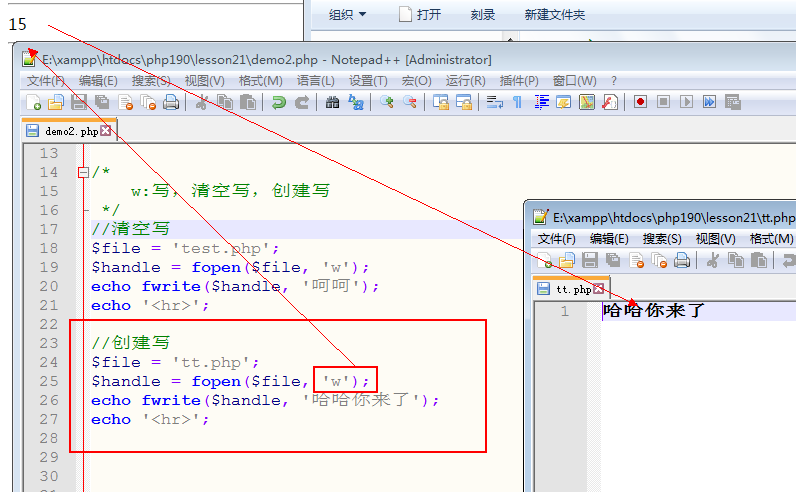


* **w:**
  + **写，清空写，创建写**

示例1：清空写



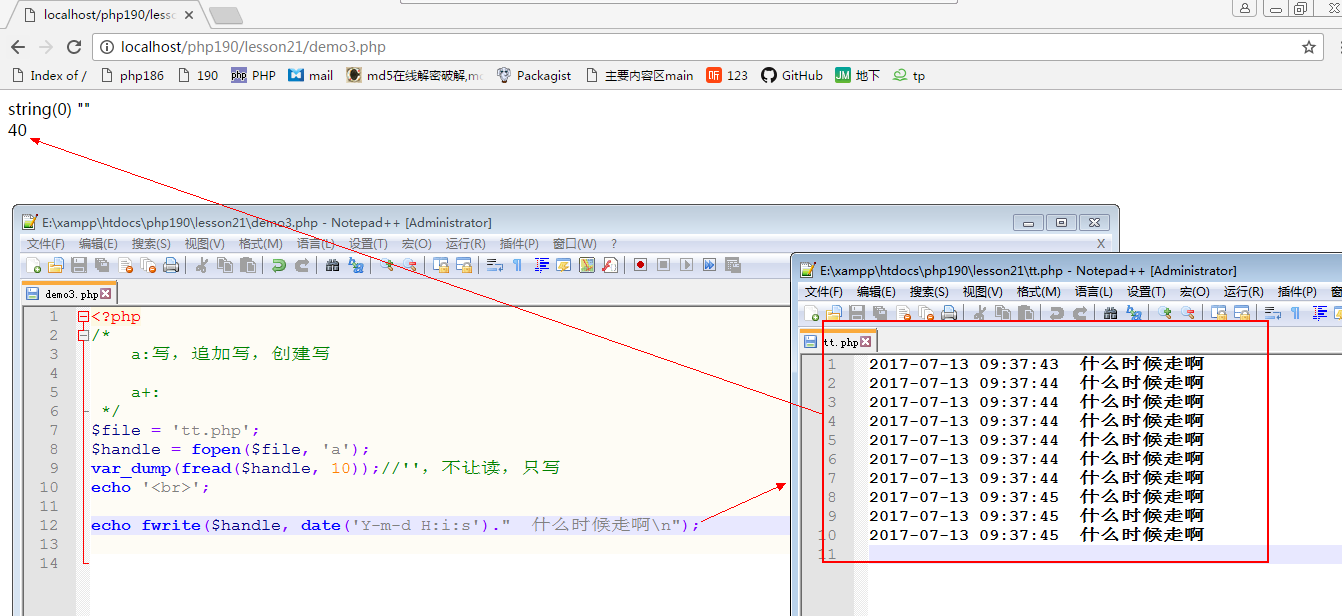
示例2：创建写



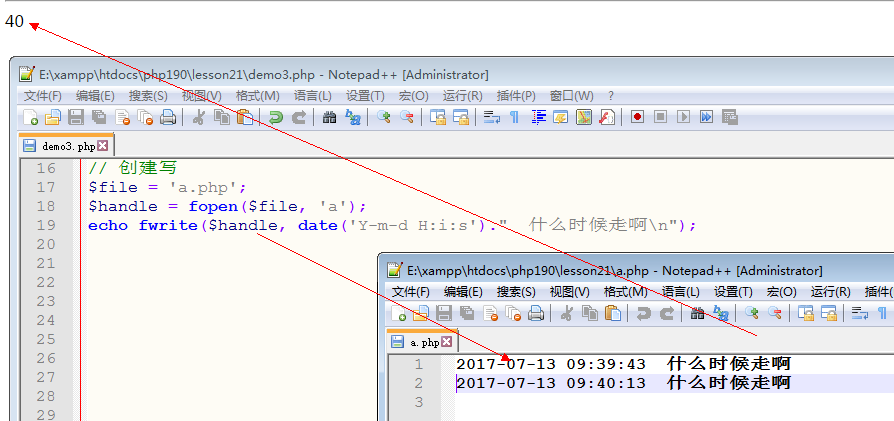
* **W+:**
  + **读写，清空写，创建写**

示例：自己完成

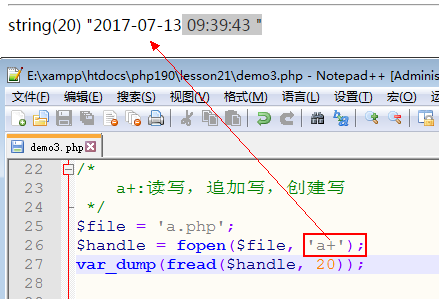
* **a:写，追加写，创建写**
  + 示例：追加写



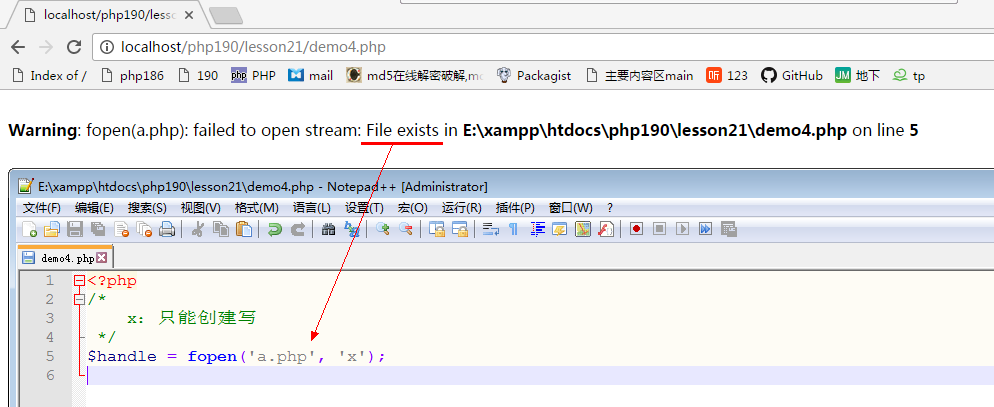
* + 示例：创建写

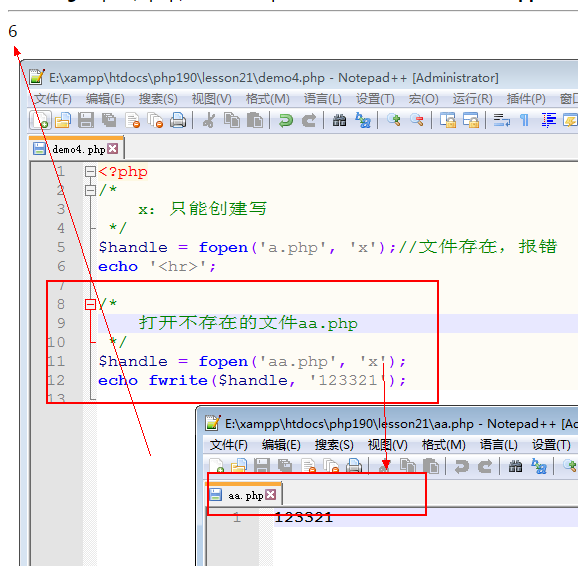


* **a+:读写，追加写，创建写**
  + 示例：读取



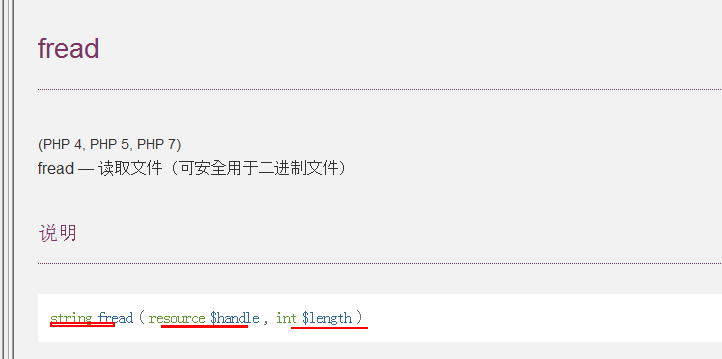
* **X:只能创建写，如果文件存在，报错**



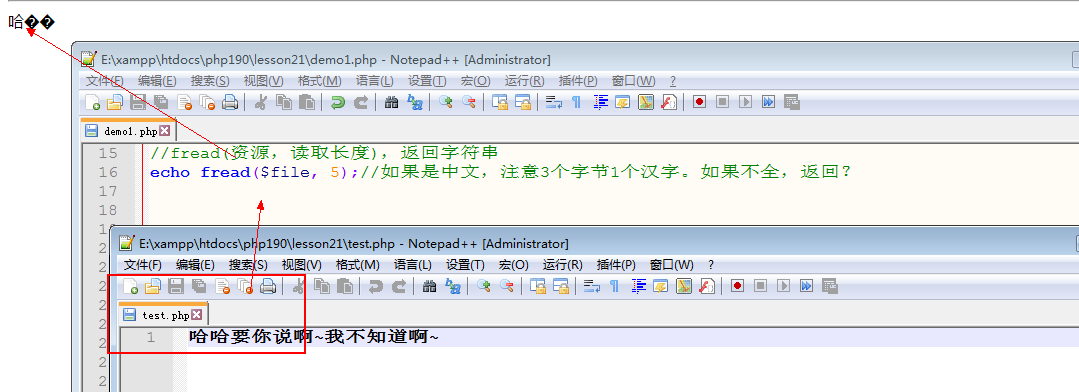


* **X+:只能创建读写，如果文件存在，报错**

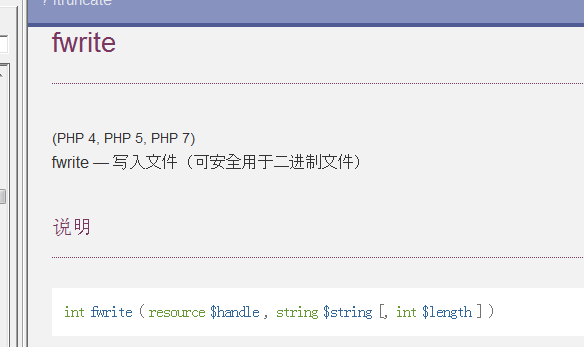
#### Fread(文件句柄, 读取长度)



示例：

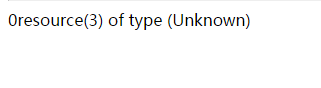


#### Fwrite(文件句柄，写入的字符串，[可选的写入长度])



#### Fclose(文件句柄)

* 关闭文件句柄，和closedir()一样

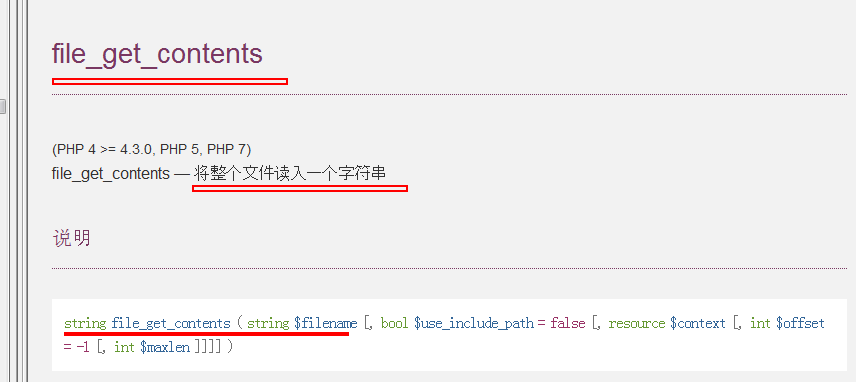


#### Rewind(文件句柄)

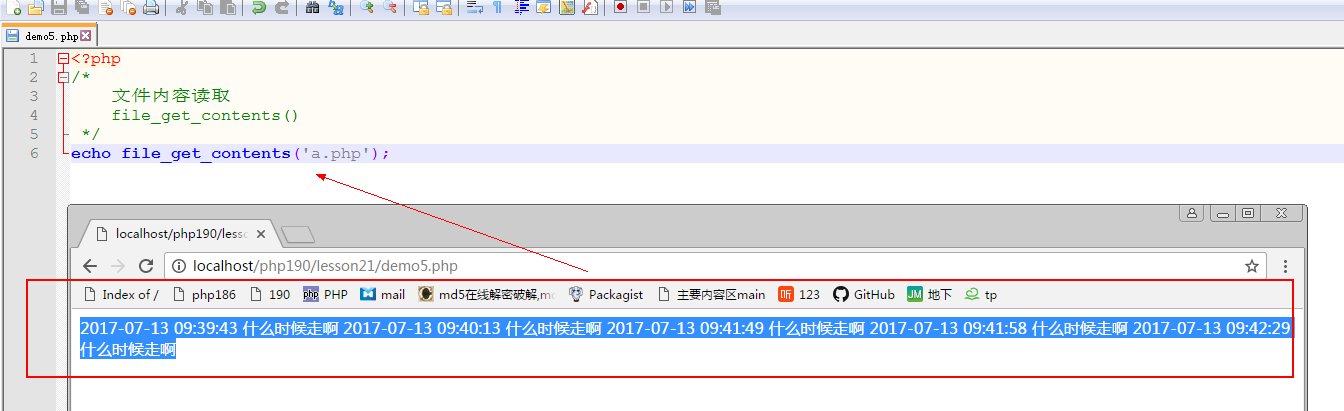
* 重置文件指针到开头，和rewinddir（），reset()一样

### 文件内容读取

#### File\_get\_contents(‘文件路径’)



示例：

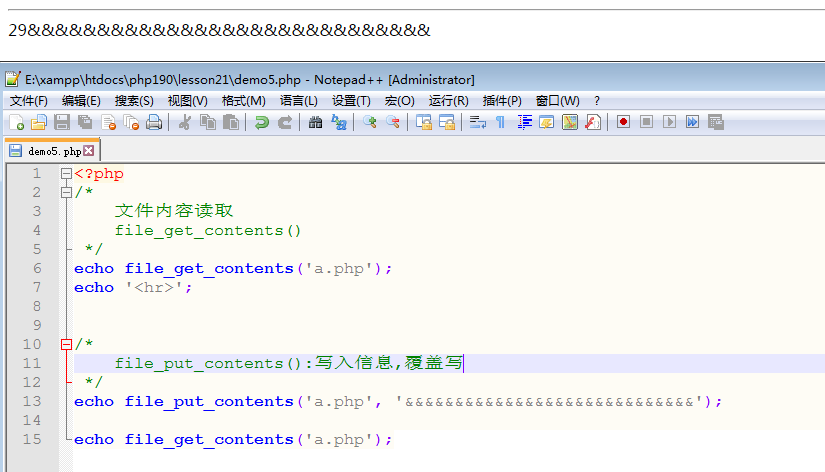


#### File\_put\_contents(文件路径，写入内容)

* 返回写入的长度
* **覆盖写**

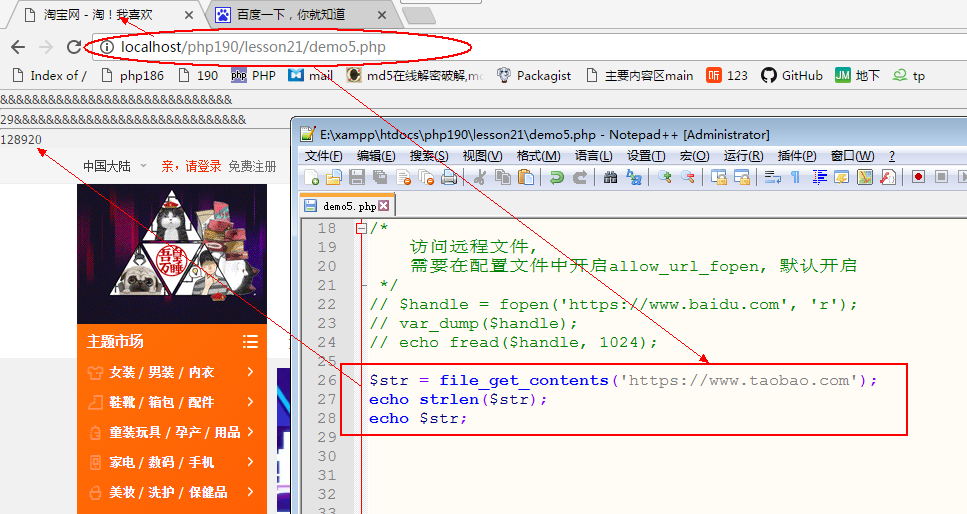


示例：



#### 访问远程的文件

* 注意：需要开启phpini配置中的allow\_url\_fopen，默认开启
* 实际上，访问远程文件就和访问本地文件是一样的道理，只不过一个是相对路径，一个是网络绝对路径。



### 文件中指针的操作

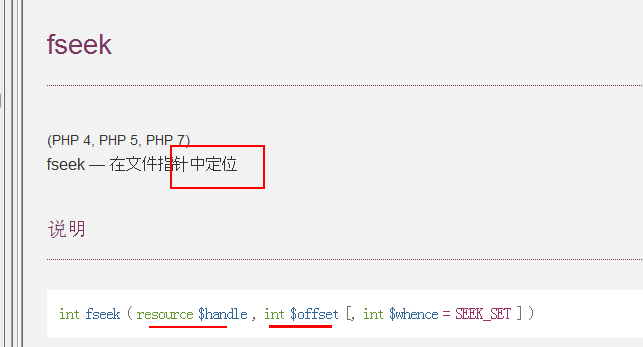
#### Ftell(文件资源)

* 访问当前的指针位置



#### Fseek(文件资源，从文件开头开始算起的偏移量)

* 操作文件指针
* 注意：偏移量始终是按文件开头计算，而不是其他位置



### 锁机制

* Flock(资源，锁常量)



