目录

第一部分　　HTML 和 CSS

DAY01　Web 基础

1.XHTML

HTML[HyperText Markup Language]:超文本标记语言：HTML 1.0；HTML 4.0

XHTML[eXtensibile HyperText Markup Language]:扩展超文本标记语言：XHTML 1.0

HTML 5 & CSS 3

2.标记分类

1》单标记

<标记名称 />

2》双标记

<标记名称>内容</标记名称>

注意：

标记名称一定要小些

标记一定顺序嵌套

行内元素：在一行显示，span

块元素：自己占一行，默认占父元素的 100%,div

3.属性(对象的特征描述)

<标记名称 属性名称="值" 属性名称="值" .../>

<标记名称 属性名称="值"...>内容</标记名称>

注意：

A、属性不是必须的

B、属性不区分先后顺序

C、属性值一定写在引号之间

D、多个属性以空格划分

E、属性名称和值都要是小写　　如：<hr color=”red” width=“20px”/>

4.XHTML 语法结构

```
<!DOCTYPE DTD>
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
        <title>Insert title here</title>
    </head>
    <body>
        网页主体
    </body>
```

```
</html>
```

DTD[Document Type Defination]:文档类型定义

告诉我们文档的根元素是谁，html 是文档的根元素，根元素可以有哪些

子元素，子元素又可以有哪些属性

Strict(严格)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Transitional(过渡)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Frameset(框架集)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

html:双标记，整个文档的根元素

xmlns：命名空间，解决命名的冲突的

head：头部标记

title:网页标题

<meta http-equiv='content-type' content='text/html;charset=utf-8'/>:

告诉浏览器以什么编码方式解析什么类型的文档

注意：所有的编码方式统一 UTF-8

body：网页主体，所写任何内容要写在 body 中

Web 页面的扩展名.html 或者.htm

5.W3C 标准

http://validator.w3.org 导入、手写、粘贴 进行检查

6.文本标记

加粗：内容 内容

倾斜：内容内容

下划线：<u>内容</u> <ins>内容</ins>

删除线：<s>内容</s> <strike>内容</strike> ~~内容~~

上标：^{内容}

下标：_{内容}

代码：<code>内容</code>

字体：内容

行内元素：内容

块元素： 内容

注意：块元素默认占父元素的 100%，自己占一行，配合 CSS 来使用

7.换行和段落

换行：

段落： 内容

8.HTML Entities(HTML 实体)

大于号>：>

小于号<：<

单引号'：'(IE6 不支持) '

双引号"："

连接符&：&

不间断的空格：

版权：©

注册商标：®

9.标题

<h1 align="left|center|right">内容</h1>

<h2>内容</h2>

<h3>内容</h3>

<h4>内容</h4>

<h5>内容</h5>

<h6>内容</h6>

10.水平线

<hr color="颜色" width="宽度" size="高度" align="center|left|right"/>

11.HTML 中注释： <!-- 注释内容...-->

CSS 中注释： /*注释内容.....*/

页面显示效果看不到，在源代码中可以看到

12.图片

提示文本的输入格式及 title 用法

必有属性：

src="目标文档的 URL"

alt="提示文本" （如：正在加载中）

注意：当 alt 值不为空的时候，通过 title 公共属性来代替

可选属性：

width:宽度

height:高度

border:边框

jpg|jpeg，gif，png

13.路径

绝对路径：标准 URL 形式

http://g.hiphotos.baidu.com/image/pic/item/b03533fa828ba61e2e4dd5554234970a304e592c.jpg

相对路径：从当前文档出发

./:当前目录

../:上级目录 ../../:上上级目录

根相对路径：以/开始的路径 (php 所对应服务器根目录)

14.公共属性(除了 html,link,base,head)

id:

class:

style:

title:注释，提示文本

15.列表

无序列表

```
<ul  type="disc|circle|square|none">
    <li>内容</li>
    <li>内容</li>
    ...
</ul>
```

有序列表

```
<ol  type="1|a|A|i|I"  start='起始点'>
    <li type=''>内容</li>
    <li>内容</li>
    <li>内容</li>
    ...
</ol>
```

定义列表

```
<dl>
        <dt>下定义的对象</dt>
    <dd>定义</dd>
    <dd>定义</dd>
    <dd>定义</dd>

        <dt>下定义的对象</dt>
    <dd>定义</dd>
    <dd>定义</dd>
    <dd>定义</dd>
    ...
</dl>
```

作业：做一个人员统计表

3 行 3 列的表格　表头=人员统计

15.表格

```
<table border='边框' bordercolor='边框的颜色' width='数值|百分比' bgcolor='颜色'
background='背景图像' cellpadding='内边距' cellspacing='外边距'>
    <caption>表头</caption>
    <tr>
        <th|td rowspan='合并行'>内容</th|td>
        <th|td colspan='合并列'>内容</th|td>
```

```
        <th|td>内容</th|td>
    </tr>
    ...
</table>
```

table 的属性：

align='left|center|right'

border='边框'

bordercolor='边框颜色'

width='宽度'

height='高度'

bgcolor='背景颜色'

background='背景图像，目标文档的 URL'

cellpadding='内边距，内容到边框的距离'

cellspacing='外边距，单元格到单元格的距离'

tr/td/th 属性：

rowspan='合并行'　　注：选择要合并的行数，然后删去被合并的单元格即可

colspan='合并列'

valign='top|middel|bottom'

16.Zend Studio 常用快捷键

Ctrl+D:删除当前行

Ctrl+Z:撤销操作

Ctrl+Y:Ctrl+Z 的逆过程

DAY02　超链接　表单

1.超链接

1》可以通过链接实现页面跳转

2》链接元素

3》文本、图片都可以作为链接元素

4》target='窗口的形式'

_self:默认值，在当前窗口中打开

_blank:新窗口中打开

在指定的框架集的窗口中打开

_top:在顶层窗口打开

_parent:在父窗口打开

或者 链接页 name=" main" 设 target= "main",在主窗口打开

作业：frameset 框架集中链接显示在主主窗口

5》href 的形式

链接到网页或者是浏览器支持的格式

网页：

静态网页：.html .htm

动态网页：和数据库有交互 .php

浏览器支持的格式：

.jpg .jpeg .png .gif

链接到下载资源:

可以通过专门的压缩工具或者可以通过程序实现压缩

直接写压缩之后的文件名称

发送邮件:

内容

锚点：

A、创建锚点 ：

B、使用锚点：

锚点和链接在同一文档下 ：内容

如果链接和锚点不在同一文档下：

内容

通过 id 在标记上定义一个唯一名称，#id 名称即可找到

核心属性：

id:唯一标识符

class

style

title

空链接

内容

内容

点击链接原地不动

内容

链接到 Js 代码

弹出框 alert()

内容

内容

弹出警示对话框 confirm()

内容

弹出输入框 prompt()

内容

2.表单

1》表单的作用：收集客户端信息，发送到服务器端

2》表单的标记

<form action='目标文档的 URL' method='get|post'>

</form>

enctype='x-www-application-urlencode'默认值

enctype=’ multipart/form-data’ 上传文件必须为此

method='get|post'

get:数据附着在地址栏之后

post：数据附着在 HTTP 的头信息中

action='目标文档的 URL'

3》表单中的控件

单行文本框:text

<input type='text' name='有意义的名称' value='默认值' maxlength='最大字符数'/>

可以通过 placeholder 代替 value

密码框：password

<input type='password' name='名称' placeholder='值' maxlength='最大长度'/>

单选框：radio

<input type='radio' name='名称' value='提交值' checked='checked'/>

注意：一组类型的单选框名称相同

复选框：checkbox

<input type='checkbox' name='名称[]' value='提交值' checked='checked'/>

注意：

一组类型的复选框命名成数组形式

数组：

名称[]

名称[数字]

名称[字符]

浏览框：file

<input type='file' name='名称'/>

注意：

如果表单有浏览框控件

method='post'

enctype='multipart/form-data'

告诉表单在发送数据之前如何对数据编码

<form action='目标文档的 URL' method='post' enctype='multipart/form-data'>

多行文本框：textarea

<textarea name='名称' rows='行高' cols='列宽' placeholder='默认值' readonly='readonly'></textarea>

readonly='readonly'代表只读

下拉框

单选下拉框

```
<select name='名称'>
    <option value='提交值' selected='selected'>内容</option>
    <option value='提交值'>内容</option>
    <option value='提交值'>内容</option>
    ...
</select>
```

分组

```
<select name='名称'>
    <optgroup label='分组名称'>
        <option value='值'>xx</option>
        ...
    </optgroup>

    <optgroup label='分组名称'>
        <option value='值'>xx</option>
        ...
    </optgroup>
    ...
    <option value='值'>xxx</option>
</select>
```

多选下拉框

```
<select name='名称[]' multiple='multiple' size='长度'>
    <optgroup label='分组名称'>
        <option value='提交值'>xxx</option>
        ...
    </optgroup>
```

</select>

multiple='multiple'代表多选

名称要命名成数组形式

隐藏域：hidden

<input type='hidden' name='名称' value='值' disabled='disabled'/>

按钮：

提交按钮

<input type='submit' name='名称' value='值' disabled='disabled'/>

图片当做提交按钮

<input type='image' src='目标文档的 URL' disabled='disabled'/>

重置按钮

<input type='reset' name='名称' value='值' disabled='disabled'/>

自定义按钮

<input type='button' name='名称' value='值' disabled='disabled'/>

通过 js 控制自定义按钮的行为

disabled='disabled'代表禁用

HTML5 新增：

datetime:日期+时间

datetime-local:本地日期时间

time:时间

month:月份

week:一年的内第多少周

email:检测邮箱的合法性

url:检测 URL 的合法性

number:检测数字的合法性

search:搜索

range:范围

tel:电话

color:颜色

min:最小值

max:最大值

step:步长

required='required'必须的

autofocus:自动获得焦点

autocomplete:自动填写功能

DAY03 框架集 CSS

1.框架集

1》框架集的语法结构

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<html xmlns='http://www.w3.org/1999/xhtml'>
    <head>
        <title>网页标题</title>
        <meta http-equiv='content-type' content='text/html;charset=utf-8'/>
    </head>
    <frameset rows='100,*' frameborder='yes(默认值)|no' border='边框大小'>
        <frame src='目标文档的 URL' noresize='noresize' scrolling='yes|no|auto(默认值)'/>
        <frameset cols='150,*'>
            <frame src='目标文档的 URL'/>
            <frame src='目标文档的 URL'/>
        </framset>
    </frameset>
    <noframes>
        <body>
```

```
            <h1>Sorry 您的浏览器不支持框架集</h1>
        </body>
    </noframes>

</html>
```

2.iframe 嵌套网页

```
<iframe src='目标文档的 URL'></iframe>
```

3.音频

```
<audio src='目标文档的 URL' controls autoplay loop muted >
<source src='资源的地址' type='MIME 类型'/>
</audio>
```

controls:控制面板

autoplay:自动播放

loop:循环播放

muted:静音输出

4.视频

```
<video src='目标文档的 URL' controls autoplay loop muted posted='图片地址' width='宽度'
height='高度'>
    <source src='资源的地址' type='video/mp4'/>
</video>
```

5.meta

设置编码

<meta http-equiv='content-type' content='text/html;charset=utf-8'/>

告诉浏览器以什么编码方式解析什么类型的文档

实现刷新

<meta http-equiv='refresh' content='秒数'/>

实现重定向

<meta http-equiv='refresh' content='秒数;url=目标文档的 URL'/>

关键字

<meta name='keywords' content='关键词,关键词,...' />

描述

<meta name='description' content='描述的内容'/>

作者

<meta name='author' content='king'/>

6.CSS

1》CSS 的作用：修饰页面

2》CSS[Cascading Style Sheet]:层叠样式表

3》CSS 的使用方式

书写于标记内部 style

通过<style>标记书写于 head 标记中

通过@import 目标文档的 URL

通过 link 链接外部样式表

CSS 文件的扩展名为.css

<link href='目标文档的 URL' type='text/css' rel='stylesheet' media='ALL' charset='utf-8' />

书写于 head 中

4》CSS 的语法结构

```
选择器{
    样式;
    样式;
    样式;
    ...
}
```

样式=属性名称:属性值;

```
h1{
    color:red;
}
```

5》选择器

统配选择器：*

自动匹配文档中的所有元素

```
*{
```

样式;

...

}

标记选择器：tag

tag{

样式;

...

}

ID 选择器：

#id 名称{

样式;

...

}

CSS 会用到 id

js/Jquery 会用到 id

类选择器：

.类名称{

样式;

...

}

标记名称.类名称：标记具有 class 这个类名称的元素才会有这个样式

类选择器支持词汇列表的形式 class='name1 name2 ...'

群组选择器：

选择器,选择器,...{

样式;

...

}

缩写的形式

后代选择器：

选择器 1 选择器 2{

样式;

...

}

选择器 1 和选择器 2 至少为父子关系

对选择器 2 有样式

子代选择器：

选择器 1>选择器 2{

样式;

...

}

选择器 1 和选择器 2 只能为父子关系

兄弟选择器：

选择器 1+选择器 2{

样式;

...

}

注意：对紧随选择器 1 之后的这个兄弟起作用

伪类选择器：

:link->设置链接未被访问时的样式

:visited->设置链接访问过的样式

:hover->设置鼠标悬浮于元素上方的时候添加的样式

:active->元素被激活时的样式

:link{

样式;

}

选择器:link{

样式;

...

}

link 和 visited 给一个样式

hover 和 active 给一个样式

注：必须遵循 LVHA 原则

:focus->具有输入域的元素获得焦点的时候的样式

:first-child->向元素的第一个子元素添加样式

看看是不是父亲的第一个孩子，如果是的话会有效果

:first-letter->向元素的一个字母添加样式

:first-line->向元素的第一行添加样式

公共属性：

id：唯一标识，名称不能重复

class：一类的

style:样式

title:

4》常用样式

修饰字体：

color:red;

颜色值：预定义的颜色名称

十六进制的颜色值

rgb(x1,x2,x3) x 的值为 0~255

rgb(%x1,$x2,$x3) 0~100%

rgba(x,x,x,0~1) 设置透明颜色 CSS3 新增

font-size:修饰字体大小 20px

font-weight:bold;使字体加粗

font-family:修改字体形态

font-family:"幼圆","微软雅黑","new Nork"

text-decoration:none|underline(下划线)|overline(上划线)|line-through(删除线)

可以结合起来使用

5》CSS 中的注释： /*注释内容*/

DAY04 选择器 浮动

1.选择器

1》伪元素

:before->在当前元素的内容之前添加内容

:after->在当前元素的内容之后添加内容

```
选择器:before{
    content:'内容';
}
选择器:after{
    content:url(目标文档的 URL);
}
```

2》属性选择器

```
[attr=value]{
    样式;
```

...

}

包含这个值

[attr*=value]{

样式;

...

}

以值开始

[attr^=value]{

样式;

...

}

以值结束

[attr$=value]{

}

3》UI 元素选择器

:disabled->元素禁用状态时的样式

:enable->元素启用时的状态

:checked->选中时的样式

::selection->选中文字时的样式

2.选择器的优先级问题

1》使用 CSS 的时候

书写于标记内部>内部样式>外部样式

注意：如果内部样式和外部样式冲突的话需要根据他们两个的书写顺序

2》选择器的优先级

每个选择器都有权值，权值越大，越先执行

书写于标记内部->1000, 1,0,0,0

ID 选择器->100

类选择器->10

标记选择器->1

伪元素->1

伪类->10

!important 代表很重要，不能被覆盖 优先级最高

3》边框样式

border:1px solid red;

border-width:设置宽度

border-style:边框的样式

none:没有边框，默认值

solid:实线

dotted:点

dashed:虚线

double:双线

border-color:设置颜色

设置每一个边框的样式

border-top:width style color;

border-top-width:1px;

border-top-style:solid;

border-top-color:red;

border-bottom:底部边框

border-left:

border-right:

4》轮廓样式

outline:width style color

outline-width:轮廓的宽度

outline-style:

outline-color:

5》布局属性

:行内元素,行内元素不能直接设置宽和高

：块元素

width:宽度

height:高度

margin:外边距

margin:value;上下左右

margin:value value;上下 左右

margin:value value value;上 左右 下

margin:value value value value;上 右 下 左

margin-top:

margin-bottom:

margin-left:

margin-right:

注意：垂直外边距重合的时候显示的是大的

可以写负值

调节盒子到盒子的距离

行内可以调节外边距

外边距重合的时候，显示的是外边距的和

padding:内边距

padding:value;

padding:value value;

padding:value value value;

padding:value value value value;

padding-top:

padding-bottom:

padding-left:

padding-right:

注意：内边距不可以写负值和 auto

6》浮动

float 实现浮动：

float:left|right

当没有设置块元素的宽和高的时候，

如果让他浮动，会尽量收缩到包含内容为止

碰到包含框或者包含框的内外边距就结束，如果放不下的时候会把它挤下去

浮动只能是水平方向 left 或者 right

clear:清除浮动

clear:left|right|both

浮动元素造成的影响：

会对下级元素造成影响，clear=both，即可清除对下级元素的影响

如果包含框中的子元素都浮动的话，父元素的还没有设置高度，

这时候父元素的高度会被清空为 0

只用对父元素应用样式 class='clear'

overflow:hidden|auto,就可以解决父元素高度清空为 0 的情况

如果对行内元素做浮动，可以设置行内元素的宽度和高度

DAY05　　属性相关

1.display:如何显示元素

display:none 隐藏

block 块元素

inline 行内元素

inline-block:以块的形式显示为行内元素，可以设定宽和高

visibility:visible(显示)|hidden(隐藏)

display 隐藏的元素不保留原来文档的文档流中的位置

visibility 隐藏的元素保留文档流中的位置

overflow:元素超出部分如何处理

visible(默认值，撑破盒子)

hidden(超出部分隐藏)

auto(根据内容是否放的下自动添加滚动条)

scroll(带有滚动条)

overflow-x:

overflow-y:

2.文本相关属性

1》white-space：

normal：默认值，自动换行

nowrap:强制在同一行内显示所有文本

2》text-overflow：clip,不显示

ellipsis：显示...省略标记

3》word-wrap:设置单词是否折行

normal:撑破边界

break-word:拆开单词

4》word-spacing:调节字之间的间隔

5》letter-spacing:调节字母之间的间隔

6》text-indent:文本缩进

可以是正数，也可以是负数

7》text-align:设置文本的对其方式

left|center|right

3.字体相关

1》color:设置字体颜色

2》font-size:字体大小

3》font-weight:normal(默认值)|bold(加粗 700)

100~900

4》font-style:设置字体是否倾斜

normal(默认值)

italic(倾斜)

5》font-variant:显示小型的大写字体

normal(默认值)

small-caps(小型的大写字体)

6》line-height:行高,25px/1.5

高度和行高一致即可

7》font-family:设置字体形态

font :

[font-style][font-variant][font-weight]字体大小 [/line-height]字体形态

font:italic small-caps bold 20px/40px Arial;

font: 20px Arial;

8》text-transform

capitalize:将每个单词的第一个字母转换成大写，其余无转换发生

uppercase:转换成大写

lowercase:转换成小写

9》阴影的效果

text-shadow:x y 模糊 颜色[,...];

4.背景图像

background:color image repeat attachment position;

background-color:背景颜色

background-image:url(目标文档的 URL);

background-repeat:repeat no-repeat repeat-x repeat-y

background-attachement:scroll fixed

background-position:背景图像的位置

水平方向 left center right

垂直方向 top center bottom

left=left center

DAY06 定位

1.列表样式

list-style: type image position

list-style-type:

list-style-image:

list-style-position:

2.透明度

opacity:0~1

filter:alpha(opacity=0~100);针对于 IE 浏览器

3.定位

position:static(默认值,静态的)

fixed(固定定位):相对于浏览器的窗口

absolute(绝对定位):

当祖先元素有相对定位的时候，这时候如果设置绝对定位是相对于祖先元素的位置来定位

如果祖先没有定位，则相对于浏览器窗口定位

relative(相对定位):相对于文档流中自身的位置来定位

top:

bottom:

left:

right:

通过 z-index:调节层的显示顺序,只能对于定位元素可以设置

可以正可以负 默认是 0

z-index 的值越大越先显示

第二部分 PHP 基础

DAY01 PHP 基础

1.PHP

PHP[HyperText Preprocessor]:超文本预处理器

LAMP：Linux+Apache(Web 服务器)+MySQL(数据库)+PHP

2.搭建环境

WAMP：Windows+Apache+MySQL+PHP

1》安装 Apache 服务器

conf 目录：apache 配置文件所在目录，httpd.conf 是 Apache 的配置文件

配置文件中需要注意的：

a.Listen 80,默认端口

b.DocumentRoot "F:/phpdev/apache2.2/htdocs",

Apache 的默认主目录

c.Directory "F:/phpdev/apache2.2/htdocs,浏览目录

d.DirectoryIndex index.html,默认主页

2》访问 Apache 服务器

http://localhost

http://127.0.0.1

3》修改 Apache 的默认主目录

修改 Apache 的配置文件 httpd.conf

DocumentRoot "F:/phpdev/apache2.2/htdocs",

Apache 的默认主目录

Directory "F:/phpdev/apache2.2/htdocs,浏览目录

重启服务器即可生效

4》Apache 的工作原理

5》安装 PHP

解压安装包，找到安装目录下的 php.ini-development 文件，

将其重命名成 php.ini,作为 PHP 的配置文件

6》将 apache 和 PHP 绑定到一起

配置 apache 的配置文件,httpd.conf

#加载 PHP5 模块

LoadModule php5_module "F:\phpdev\php5.4\php5apache2_2.dll"

#加载 PHP 的配置文件的位置

PHPIniDir F:\phpdev\php5.4

#告诉 apache 服务器什么样类型的文件交给 PHP 引擎来处理

AddType application/x-httpd-php .php

学生机的主目录：/home/www

3.PHP 基础

1》PHP 文件扩展名.php

2》PHP 的语法结构

a.标准风格，XML 风格(开发中推荐使用的风格)

<?php

代码段;

?>

注意：PHP 的书写规范

每一句代码以英文的分号结束

如果文档中不只有 PHP 代码，开始和结束标记一定要成对出现

可以在任何位置，出现多少次都可以

b.短风格

```
<?
    代码段;
?>
```

注意：使用短风格需要配置 PHP 的配置文件 php.ini,short_open_tag=On,

之后重启服务器即可

c.ASP 风格

```
<%
    代码段;
%>
```

需要开启 asp_tags=On，重启服务器

d.长风格

```
<script language='php'>
    代码段;
</script>
```

4.常用内容

a.echo 语句结构

echo 输出一个或者多个字符串

echo 1;

echo 1.2;

echo 'this is a test';

b.解决中文乱码

header("content-type:text/html;charset=utf-8");

header 写在任何输出之前

5.PHP 文档组成

PHP 代码

HTML/CSS

Js/Jquery

6.PHP 中的错误

a.Parse error(解析错误): syntax error(语法错误),

unexpected '<' in F:\test\psd1501\php\day01\demo\test_3.php

on line 5

b.Notice(注意): Undefined variable: sdkljflksdjfl in

F:\test\psd1501\php\day01\demo\variable_10.php on line 36

配置 PHP 的配置文件 php.ini

搜索 error_reporting = E_ALL & ~E_NOTICE

重启服务器

7.变量

a.变量在程序执行期间可以变化的量，保存数据

b.声明变量

$变量名称;

声明变量的同时赋值

$变量名称=值;

一次声明多个变量赋相同的初始值

$变量名称=$变量名称...=值;

注意：

变量名称一定以字母或者下划线开始，后面可以跟上数字字母和下划线

不要包含特殊字符，* ? /

变量名称最好含义明确

变量名称最好遵循驼峰标记法和下划线法

驼峰标记法：

大驼峰：$UserName,$FirstName

小驼峰：$userName,$firstName

下划线法：

$_username

变量名称区分大小写

$a

$A

php 是弱类型语言，变量可以不声明直接使用

如果变量名称重复，后面的值会覆盖之前的值

c.可变变量

等量代换的原则

8.PHP 中的注释

给当前行添加注释或者取消注释

Ctrl+/或者是 Ctrl+Shift+c

//单行注释-C++风格

#单行注释- Shell 风格

多行注释

/*

注释内容

*/

9.PHP 中的数据类型

一共支持 8 种主要数据类型和 4 种伪类型

a.标量类型(只能存储单一数据 scalar type)

整型(int|integer)：十进制的数字、八进制、十六进制，就是

整数，可以是正整数和负整数

带符号-21 亿~21 亿

无符号：0~42 亿

超出整型会产生溢出的现象

浮点型(float|double|real)：小数或者是科学计数法

的写法带有 e 或者 E

2e3=2x10 的三次方

2E-3=2x10 的负三次方

不要比较浮点数的大小，涉及到精度

布尔类型(bool|boolean)：真和假,true 或者 TRUE，flase 或者 FALSE

字符串型(string)：

定界符

''

""

单引号和双引号的区别:

1》单引号不解析变量，双引号解析变量

2》单引号的执行速度快

3》单引号直解析\'和\\,而双引号解析所有的转义符

当定界符中的内容和定界符冲突的时候，这时候就需要使用转义符

\'->'

\"->"

\$->$

\\->\

源代码中有效果

\n->换行

\r->回车

\t->水平制表符，相当于 tab

大文本或者大量字符串的时候

hereDoc->" （相当于双引号，这个 HEREDOC 不用写）

<<<名称

内容

名称;

注意：结束名称之前不能有任何输出

nowDoc->'

<<<'名称'

内容

名称;

{}的作用：

把变量当做一个整体

$username

{$username}

${username}

对字符串中的指定字符做增删改查

根据下标找到对应的字符，一次只能添加跟新或者查找或删除一个字符

字符串的下标是从 0 开始

b.复合类型:数组　　对象

c.特殊类型：资源　　空（NULL）{

未声明的变量；

声明变量的同时赋值为 NULL；

经过 unset()注销过的变量；

}

d.4 种为类型：number　　数值

mixed　　混合型

callback　　回调函数

void　　没有返回值

10.常用函数

var_dump($var,$var...):打印变量的详细信息

DAY02　　PHP 数据类型

1.PHP 中支持的数据类型

a.8 种主要数据类型

标量类型(Scalar Type)

整型

浮点

布尔

字符串

复合类型

数组(array)

对象(object)

特殊类型

资源(resource)

空 NULL:

未声明的变量

声明变量的同时赋值为 NULL|null

经过 unset 注销过的变量

b.4 种伪类型

number:数值

mixed:混合的

callback:回调函数

void：没有返回值

2.常量

系统常量：

PHP_VERSION:PHP 的版本

PHP_OS:操作系统

常量一经定义，在程序执行期间不能改变，保存值

1》定义常量

define($name,$value):定义常量

如果希望常量名称不区分大小写，可以在定义的时候给参数

define($name,$value[,$flag=false]):

define('TEST','this is a test',true):不区分大小写

注意：

常量名称一般使用大写

常量的值只能为标量类型

const name=值;

2》使用常量

直接写常量名称即可

constant($name):得到指定常量名称的值,如果存在则返回常量的值

返回 NULL

3》检测常量名称是否被定义

defined($name):检测常量名称是否存在,如果存在返回的 TRUE，不存在

返回的是 FALSE

4》魔术常量

__LINE__:得到当前的行号

__FILE__:得到当前脚本绝对路径包含文件名称

__DIR__:得到当前文件所在的绝对路径

__FUNCTION__:得到当前的函数名称

__CLASS__:得到当前的类名

__METHOD__:得到当前的方法名称

__TRAIT__:得到当前 TRAIT 名称

__NAMESPACE__:得到当前命名空间的名称

5》get_defined_constants():返回当前所有可用常量，

返回的数组，包含系统常量和自定义常量

print_r()打印数组

3.预定义变量

$_GET:HTTP 的 GET 变量,接收以?形式传参的值

$_POST:HTTP 的 POST 变量，接收表单以 post 形式发送数据的值

接收的原则：根据名称找到对应的值

$_REQUEST:$_GET+$_POST+$_COOKIE

$_ENV:环境变量

$_SERVER:服务器变量

$_COOKIE:HTTP COOKIE

$_SESSION:会话变量

$_FILES:文件上传变量

$GLOBALS:超全局变量，包含以上所有的值

4.PHP 类型转换

1》自动转换(隐式转换)

其他类型转换成数值型：

true->1

false->0

null->0

'3king'->3

'true'->0

其他类型转换成字符串型

数字->数值本身

true->1

false->空字符串

null->空字符串

数组->Array

对象不能直接转换成字符串，可以通过魔术方法__toString()

资源->Resource id #数字

其他类型转换成布尔类型

找出转换布尔类型假的有

0->false

0.0->false

''或者""或者是'0'或者"0"->false

null|NULL->false

array()空数组->false

2》强制转换(显示转换)

a.临时转换

整型：(int)|(integer)

浮点类型：(float)|(double)|(real)

布尔类型：(bool)|(boolean)

字符串类型：(string)

空：(unset)

数组：(array)

对象：(object)

通过函数的形式

intval($var):返回转换整型之后的值

floatval($var)|doubleval($var):返回转换成浮点类型之后的值

boolval($var):转换成布尔类型，要求 PHP 版本大于 5.5.0

strval($var):转换成字符串型

b.永久转换

settype($var,$type):将变量设置成什么类型,设置成功返回为 true

失败返回为假

gettype($var):得到变量的类型,返回的变量的类型，返回的是字符串

$var=123;

type 的可能值为：

“boolean” （或为“bool”，从 PHP 4.2.0 起）

“integer” （或为“int”，从 PHP 4.2.0 起）

“float” （只在 PHP 4.2.0 之后可以使用，对于旧版本中使用的“double”现已停用）

"string"

"array"

"object"

“null” （从 PHP 4.2.0 起）

变量函数库来检测变量的类型

is_*

is_int($var)|is_integer($var)|is_long($var):检测变量是否为整型

is_float($var)|is_double($var)|is_real($var):

is_string($var):

is_bool($var):

is_scalar($var):检测是否为标量类型

is_array($var):

is_object($var):

is_resource($var):

is_null($var):

is_numeric($var):检测是否为数值型或者是字符串形式的数值

$var=123;

$var=12.3;

$var='123';

$var='3.3';

$var='3king';

DAY03　　运算符

1.运算符

1》算术运算符

+ - * / %

++ --

递增递减运算符

前缀形式：先加减 1，在执行

后缀形式：先执行，在加减 1

$var=1;

++$var;

$var--;

数值型支持自增自减运算符

布尔类型不支持自增自减运算符

字符串只支持自增不支持自减

空只支持自增不支持自减

ord($char):得到指定字符的 ASCII 值

chr($ascii):根据 ASCII 得到对应的字符

a~z-97~122

A~Z-65~90

2》字符连接运算符

通过.连接字符串、或者连接变量

3》赋值运算符

= += -= *= /= %= .=

```
$var=1;
$var+=2;//$var=$var+2;
echo $var;//3
$str='hello';
$str.='world;//$str=$str.'world';
echo $str;//hello world
```

4》比较运算符

比较的结果只能是 true|false

> >= < <=

== != <>

=== !==

==:比较两个表达式是否为 true，比较两个表达式的值是否相等

===:比较值和类型都要相同，这才代表相同

5》逻辑运算符

逻辑运算符的结果也是 true|false

逻辑与:&& and,并且的意思，必须是两个表达式都为 true，结果才为 true

true && true=true

true && false=false

false && true=false

false && false=false

注意：如果第一个表达式为 false，造成短路，整个结果就为 false

逻辑或:|| or，或者的意思，至少有一个表达式为 true，结果就为 true

true || true=true

true || false=true

false or true=true

false or false=false

注意：如果第一个表达式为 true，第二个表达式就被短路了，结果就为 true

逻辑非:通过! ,代表取反

!true=false

!false=true

逻辑异或:通过 xor,如果两个表达式不同为 true 或者 false，结果为 true

否则的话的是 false

true xor true=false

false xor false=false

true xor false=true

false xor true=true

6》错误抑制符:可以抑制错误的输出，通过@，加在会产生错误的表达式之前

7》执行运算符：``可以执行外壳命令，相当于 shell_exec($cmd)

8》三元运算符:

一元运算符：!true

二元运算符：3+1

三元运算符：

exp1?exp2:exp3;

if(exp1){

exp2;

}else{

exp3;

}

9》运算符的优先级

echo (1+3-2)*4/5;

可以通过()改变优先级

运算符的优先级有短路先考虑短路

10》常用函数

mt_rand():得到随机数

strlen():得到字符串长度

strpos():得到字符在字符串中第一次出现的位置

stripos():不分大小写得到字符在字符串中第一次出现的位置

strrpos():得到字符在字符串中最后一次出现的位置

strripos():不分大小写得到字符在字符串中最后一次出现的位置

strcmp():比较字符串

strcasecmp():不分大小写比较字符串

ord():得到字符对应的 ASCII 码值

chr():得到 ASCII 码值对应的字符

DAY04　　流程控制

1.流程控制

1》条件

if 语句的形式

a.if(exp)

执行一句;

不建议使用

b.if(exp){

exp 为真的代码段;

}

c.if(exp){

exp 为 true 的代码段;

}else{

exp 为 false 的代码段;

}

d.if(exp1){

}elseif(exp2){

}...

if(exp1){

exp1 为 true 的代码段

}elseif(exp2){

exp2 为 true 的代码段

}...else{

以上条件都为 false，会执行到 else 中的代码段;

}

== 只比较值是否相等

```
switch(exp){
    case  值 1:
        执行代码段;
        //break;
    case  值 2:
        执行的代码段;
        break;
    ...
    default:
        当以上 case 都没有匹配上的时候，会
        执行到 default 中的代码段
}
```

break:碰到 break，switch 语句就结束了

常用函数：

exit()|die():终止执行函数

数学函数：

max():求最大值

min():求最小值

ceil():进一步取整

floor():取整

round():四舍五入

abs():绝对值

sqrt():开方

pow($number,$exp):幂运算

mt_rand():产生更好的随机数

rand():产生个随机数

字符串函数

strstr()|strchr():寻找第一个字符出现的位置 strpos:寻找字符在字符串中第一次出现的位置

stristr():不分大小写的寻找

strrchr():从右往左寻找

substr($string,start,length):返回字符串的一部分

start：开始位置 可正可负，负数是从右往左

length:长度 可正可负，正数是长度，负数是终止于从右往左的位置

str_replace():字符替换函数

str_ireplace():不论大小写替换

也可通过替换本函数转换字符串中的特殊字符为实体字符，达到过滤字符的目的

strip_tags():过滤标记<>函数

strip_tags($string[,$allowTags]):可以通过$allowTags 指定那些标记不被过滤掉，如 strip_tags($string,’<h1>’)；

htmlspecialchars():对字符串中的< > ‘ “ &进行处理（默认不处理’）

ENT_COMPAT: 只处理双引号，不处理单引号，如 echo hrmlspecialchars ($string,ENT_COMPAT)；

ENT_QUOTES：处理单引号和双引号；

ENT_NOQUOTES：不处理单引号和双引号

日期时间：

date_default_timezone_set():设置默认时区

PRC

Asia:shanghai

Asia:chongqing

date_default_timezone_get():获得默认时区

date($format[,$time=time()]):得到服务器的日期时间

Y:四位的年份

m:两位的月份

d:两位的日

H:两位小时

i:两位的分钟

s:两位的秒

w：一周内的第多少天，返回值 0~6,0 代表星期日

格式：date("Y-m-d".'w'."H:i:s") 年-月-日 星期 时-分-秒

time():代表当前的时间戳，1900 年到现在所经历的秒数

扩展（上有）： 过滤用户输入：

1、通过 str_replace（ ）函数转换字符串中的特殊字符为实体，过滤掉 html 标记，如< “；

2、通过 Strip_tags($string)函数， 过滤标记<>,不处理其他；

Strip_tags($string，[$allowTags]) 留下指定标记，如 strip_tags($string,’ <h1>’）；

3、htmlspecialchars($string); 对字符串中的< > ‘ “ &进行处理（默认不处理’）；

ENT_COMPAT:只处理双引号，不处理单引号，

如 echo hrmlspecialchars ($string,ENT_COMPAT)；

ENT_QUOTES：处理单引号和双引号；

ENT_NOQUOTES：不处理单引号和双引号；

DAY05　　流程控制

1.流程控制

1》条件语句

A、 if(exp){

执行代码段;

if(exp){

}else{

}

}else{

执行代码段;

if(exp){

}elseif(exp){

}elseif(exp){

..

}

}

B、if(exp){

exp 为 true 的代码段;

}else{

exp 为 false 的代码段;

}

C、if(exp):

代码段;

else:

代码段;

endif;

D、if(exp1){

}elseif(exp2){

}elseif(exp3){

}else{

```
}
```

E、if(exp1):

    代码段;

elseif(exp2):

    代码段;

elseif(exp3):

    ...

else:

    ...

endif;

2.实现页面跳转的方式

```
<script type='text/javascript'>
    location.href="目标文档的 URL";
</script>
```

几秒钟后跳转指定页面

```
<meta http-equiv='refresh' content='秒数;url=目标文档的 URL'/>
```

重定向

```
header('location:目标文档的 URL');
```

3.循环

for 循环

```
for(exp1;exp2;exp3){
    循环体;(代码段)
}
```

break:通过 break 结束 for 循环

continue:跳过当次循环，进入下次循环

碰到 for 循环，会无条件执行 exp1，一般在 exp1 赋初始值

exp2 控制是否能执行循环体的条件,exp2 为 true 的话，会执行循环体

exp3 做一些操作++ --

while 循环

do...while 循环

常用函数：

字符串函数库：

strtolower():返回小写字符串

strtoupper():返回大写字符串

ucfirst():字符串首字母大写

lcfirst(): 字符串首字母小写

ucwords():单词首字母大写

trim():默认去掉字符串两端空格

trim($string[,$charlist]):去掉字符串两端指定字符串

ltrim():去掉字符串左端空格

rtrim():去掉字符串右端空格

md5():对密码进行 32 位加密　　单向，不可逆

sha1():对密码进行 40 位加密　　单向

isset():检查变量是否设置值 ,只要变量有值并且值不为 NULL 的话 ,函数返回 true ,否则返回为 false；

$var=" " 为真；$var=NULL 为假；

empty():检测变量是否为空，空为真，否则假；

认为空的情况，转换布尔类型假的情况+false；

0 0.0 '' "" '0' "0" null array( ) false

DAY06　　循环

1.循环

1》　for 循环：精确控制循环次数

for 语句如果有多个判断条件，最后一个决定跳出循环；

continue； 跳出本次循环，不执行下面语句，进入下次循环；

break； 跳出全部循环；

break 可以结束 while、do....while 和 for 循环；

2》while

```
while(exp){
循环体;                                    //没有$i++会陷入死循环；
}
```

do...while

```
do{
    无条件的执行一次循环体
}while(exp);
```

2.数组：数据的集合

索引数组：下标是数字的就是索引数组

关联数组：下标是字符串的就是关联数组

1》声明数组

array():空数组

索引数组：

array(值,值,...):下标连续的索引数组，数组的下标从 0 开始

array(键名=>键值,键名=>键值..)：声明手动指定下标的索引数组

关联数组：

array(键名=>键值,...):声明关联数组

注：1、数组、资源、对象不能做键名；

2、键名重复时后值覆盖前值；

3、echo ‘<pre>’； 以源代码的形式输出；

动态创建数组：

$数组名称[]=值;//下标连续的索引数组，从 0 开始

$数组名称[数字]=值;//手动指定数组的下标

$数组名称[字符串]=值;//关联数组

快速创建数组：

range($min,$max[,$step=1]):快速创建索引数组,下标连续的索引数组

compact():快速创建关联数组

2》数组的使用

通过键名找键值

$数组名称[键名]

3》二维数组或者多维数组的声明与使用

4》遍历数组

键名+键值的形式

```
foreach($数组名称  as $key=>$val){
    循环体;
}
```

键值：

```
foreach($数组名称  as $value){
    循环体;
}
```

```
foreach($数组名称  as $key=>$val):
    循环体;
endforeach;
```

```
foreach($数组名称  as $val):
    循环体;
endforeach;
```

DAY07　　文件上传

1.文件上传

1》客户端：

表单页面：浏览框

注意：

method='post'

enctype='multipart/form-data'

2》在服务器端：

$_FILES:文件上传变量，上传文件的信息都保存在这个变量中

所有的预定义变量都是数组

```
Array
(
    [myFile] => Array
        (
            [name] => 3.jpg
            [type] => image/jpeg
            [tmp_name] => C:\WINDOWS\Temp\php5DA.tmp
            [error] => 0
            [size] => 218581
        )

)
```

$_FILES['浏览框名称']['name']:上传文件的名称

$_FILES['浏览框名称']['type']:文件的 MIME 类型

$_FILES['浏览框名称']['tmp_name']:临时文件名称（移动包括操作的都是服务器端的临时文件）

$_FILES['浏览框名称']['error']:上传文件的错误号

$_FILES['浏览框名称']['size']:上传文件的大小，单位是字节

注意：

如果接不到上传文件的信息，检查表单那两个属性是否正确

Warning: POST Content-Length of 34811295 bytes exceeds the

limit of 8388608 bytes in Unknown on line 0

3》上传文件的错误号

(*)0 或者 UPLOAD_ERR_OK:文件上传到服务器上成功

(*)1 或者 UPLOAD_ERR_INI_SIZE:超过了 PHP 配置文件中 upload_max_filesize 选项的值

(*)2 或者 UPLOAD_ERR_FORM_SIZE:超过了表单中设置的 MAX_FILE_SIZE 选项的值

3 或者 UPLOAD_ERR_PARTIAL:代表文件部分被上传

(*)4 或者 UPLOAD_ERR_NO_FILE:代表没有文件被上传

6 或者 UPLOAD_ERR_NO_TMP_DIR:没有找到临时目录

7 或者 UPLOAD_ERR_CANT_WRITE:无法写入文件

8 或者 UPLOAD_ERR_EXTENSION:由于 PHP 的扩展程序中断了文件上传

4》PHP 配置文件中和文件上传相关的选项

file_uploads = On，是否支持通过 HTTP POST 形式上传文件

upload_tmp_dir =上传文件保存的临时目录

max_file_uploads = 20，一次最多上传多少个文件

想上传大文件需要配置的选项

upload_max_filesize = 2M，默认 2M 允许上传文件的最大值

post_max_size = 8M，表单以 POST 方式发送数据的最大值

5》在客户端做限制

限制上传文件的大小：

<input type='hidden' name='MAX_FILE_SIZE' value='允许上传文件的大小，单位为字节'/>

限制上传文件的类型

<input type='file' name='myFile' accept='规定上传文件的MIME类型,多个类型之间以逗号分隔

'/>

在服务器做限制

检测下上传文件的大小

检测上传文件的扩展名

检测文件是否是通过 HTTP POST 方式上传上来的

6》常用的 MIME 类型

css text/css

doc application/msword

exe application/octet-stream

htm text/html

html text/html

hts text/html

pdf application/pdf

rar application/octet-stream

zip application/zip

txt text/plain

jpeg|jpg image/jpeg

gif image/gif

png image/png

7》 常用函数

Explode（‘ .’,$filename） 以指定分割符拆分字符串，回数组

Shuffle（$arr） 打乱数组

Str_shuffle 打乱字符串

Implode （ ）/join（ ） 连接

key($arr):当前指针所在位置元素的键名

current($arr):返回当前指针所在位置元素的键值

next($arr):将指针向下移动一位，并且返回当前指针所在位置元素的键值

prev($arr):将指针向上移动一位，并且返回当前指针所在位置元素的键值

reset($arr):重置数组指针，将指针移动到数组的开始，并且返回当前指针所在位置元素的键值

end($arr):将指针移动到最后一位，并且返回当前指针所在位置元素的键值

explode($delimiter,$string):以指定的分隔符拆分字符串，返回拆分的数组

implode($delimiter,$array)|join($delimiter,$array): 以指定分隔符连接数组中的键值,连接成字符串返回

array_reverse($arr);//翻转数组

strrev($str);//字符串的翻转

array_rand($arr): 随机取出数组的下标

Array_rand($arr[,$num]) : 返回随机下标的数组

Array_merge($arr1,$arr2) 合并数组

Array_pop( ) 弹出数组的最后一个元素

array_search($search,$arr):如果存在返回其对应的键名，不存在返回 false

array_push($arr,$val,...):在数组末尾添加元素,返回添加之后元素的个数

array_pop($arr):弹出数组末尾的元素,返回弹出之后的值

array_unshift():在数组开始加入元素

array_shift():弹出数组开始的元素

array_sum($arr):求数组中键值的和

array_unique($array):去掉数组中重复的值

End($arr)函数 将数组的内部指针移动到最后一个元素

In_array($i,$arr) 判断某个值是否在数组中

Microtime() 微秒数，时间戳 time( );

Uniqid(Microtime()) 产生唯一字符串

DAY08　　PHP 中的函数

1.PHP 中的函数：

完成某一功能的特定的代码段

1》系统函数

2》自定义函数

function 函数名称([参数...]){

　　函数体;(代码段)

　　return 返回值;

}

注意：

a》函数名称以字母或者下划线开始，后面跟上数字+字母+下划线

b》不能包含特殊字符

c》函数名称最好含义明确，最好以动词开始

d》最好遵循驼峰标记法或者是下划线法

e》函数名称是不区分大小写,在使用的时候最好遵循大小写 strtolower StRtOlOWER

f》函数名称不支持重名，不支持重载，不能和自定义函数名称重名，也不能和系统函数名称重复

g》默认函数返回值为 NULL，如果想有返回值，可以通过 return 加上返回值

3》函数执行原理

函数不调用不执行，当自定义函数之后首先先载入到内存中，当我调用函数的时候

找到这个函数，开始执行函数体，执行完毕之后再将控制权移交到调用函数的位置上。

如果函数碰到 return，立刻结束

4》调用函数

函数名称([参数...]);

5》函数参数

函数的参数不是必须的

形参和实参：

形参是定义函数时指定的参数

必选参数：就是在调用的时候必须传的参数

可选参数：调用的时候可以传也可以不传，如果不传参的话会使用默认值

注意：如果参数中既有可选参数又有必选参数

一定必选参数在前面，可选参数在后面

实参是调用函数的时候你实际传过来的参数

常用函数

function_exists($funcName):检测函数名称是否存在，如果存在返回 true，否则返回 false

get_defined_functions():得到当前可用系统函数和自定义函数

返回的数组中包含了 2 部分，一部分是内置函数，一部分是自定义函数

DAY09 变量的作用域

1.变量的作用域

1》局部变量：函数体内声明的变量是局部变量，在函数执行之后变量就被释放了

动态局部变量：在函数执行之后释放掉了

静态局部变量：通过 static 关键字在函数体内定义变量的是静态变量，

当第一次调用函数的时候，相当于初始化，当函数执行完毕之后静态变量

并没有被释放，而是保存到静态内存中，当再次调用的时候，首先从

静态内存中取出这个变量的值，之后再来执行。

2》全局变量：在函数体外声明的变量

在函数体内使用全局变量，可以通过两种方式：

通过 global 关键字定义的变量，是全局的变量

$GLOBALS:超全局变量

2.函数的传值和传引用的区别

传值在函数体内对变量做更改不会影响这个变量本身

传引用在函数体内对变量做更改会影响这个变量本身

3.特殊形式的函数

1》变量函数：就是函数名称赋值给变量，$变量名称([参数...]);

2》递归函数：函数体自己调用自己本身

遍历目录

无限极分类

3》可变参数的函数

func_num_args():得到传入函数参数的个数

func_get_arg($index):得到指定参数的值，参数的起始点是从 0 开始

func_get_args():返回的是数组

4.包含文件

require/require_once

include/include_once

第三部分 MySQL 数据库

DAY01 数据库基础

1.数据库：存储数据的仓库

LAMP:MySQL ，MySQL5.5 的版本

2.安装与配置

1》MySQL 的配置文件

在安装目录下，my.ini 是配置文件

[client]

port=3306

[mysql]

default-character-set=utf8

[mysqld]

port=3306

basedir="F:/mysql5.5/"

数据保存的位置：datadir="C:/Documents and Settings/All Users/Application Data/MySQL/MySQL Server 5.5/Data/"

服务器端的默认编码方式：character-set-server=utf8

默认的存储引擎：default-storage-engine=INNODB

3.登陆/退出 MySQL

1》登陆:

mysql -u 用户名 -p 密码

mysql -u 用户名 -p

Enter Password:****

mysql -h 服务器名称 -u 用户名 -p

mysql -h127.0.0.1 -uroot -proot

mysql -hlocalhost -uroot -proot

mysql -hlocalhost -uroot -proot -P3306

登陆的同时打开指定数据库

mysql -hlocalhost -uroot -proot -P3306 -Ddb_name

得到 mySQL 版本

mysql -uroot -proot -V

2》退出

exit

quit

\q

Ctrl+c

3》如何设置命令提示符

a.登陆的同时设置命令提示符

mysql -hlocalhost -uroot -proot -P3306 --prompt=命令提示符

b.登陆之后设置命令提示符

prompt 命令提示符

c.常用的命令提示符

\D:完整的英文的日期时间

\d:当前打开的数据库

\h:当前服务器名称

\u:当前登陆的用户名称

2.SQL 结构化查询语言

1》DDL：数据定义语言

创建数据库、创建表、修改表结构等

CREATE / ALTER /DROP

2》DML：数据操纵语言

对数据的增删改

INSERT / UPDATE /DELETE

3》DQL：数据查询语言

查询数据

SELECT

注意：

SQL 语句以指定的分隔符结尾

SQL 语句不区分大小写，像关键字、保留字、函数大写，

和名称有关的小写

SQL 语句支持折行操作,完整的单词或者引号不要折行写

\c 当前命令不执行

可以通过键盘上下键调用之前写的命令

开启输出日志：\T 目标文件的位置

\t

mysql 中的注释

-- 注释内容

注释内容

注意：如果名称和 MySQL 保留字、关键字冲突的话，通过反引号`名称`

3.常用命令

SELECT VERSION();//得到当前版本

SELECT NOW();//当前的日期时间

SELECT USER();//得到当前用户

SELECT DATABASE()|SCHEMA();//得到当前打开的数据库名称

查看 MySQL 手册

help CREATE

\h CREATE DATABASE

? CREATE DATABASE

4.数据库相关操作

1》创建数据库

CREATE {DATABASE|SCHEMA} db_name;

-- 如果不存在则创建

CREATE {DATABASE|SCHEMA} [IF NOT EXISTS] db_name;

-- 创建数据库的同时指定编码方式

CREATE {DATABASE|SCHEMA} [IF NOT EXISTS] db_name

[DEFAULT] CHARACTER SET [=] '编码方式';

2》得到当前服务器已有数据库

SHOW DATABASES|SCHEMAS;

3》查看指定数据库的详细信息

SHOW CREATE DATABASE|SCHEMA db_name;

4》修改指定数据库的编码方式

ALTER DATABASE|SCHEMA db_name

[DEFAULT] CHARACTER SET [=] '编码方式';

5》打开指定的数据库

USE db_name;

6》得到当前打开的数据库

SELECT DATABASE()|SCHEMA();

7》删除指定数据库

DROP DATABASE db_name;

DROP DATABASE|SCHEMA [IF EXISTS] db_name;

得到上一步操作产生的警告

SHOW WARNINGS;

通过 DELIMITER 命令分隔符

5.数据表相关操作

数据表是数据库基础组成单位，数据都是存在于数据表中

数据表由行和列组成，列叫做字段，一个数据表至少有一列

行也可以叫做数据，可以没有数据，空表

1》创建数据表

CREATE TABLE [IF NOT EXISTS] tbl_name(

字段名称 数据类型 [完整性约束条件]

)ENGINE=存储引擎名称 DEFAULT CHARSET=编码方式;

2》查看当前数据库下的数据表

SHOW TABLES;

3》查看指定数据表的表结构

DESC tbl_name;

DESCRIBE tbl_name;

SHOW COLUMNS FROM tbl_name;

4》查看指定数据表的详细信息

SHOW CREATE TABLE tbl_name;

5》删除指定的数据表

DROP TABLE [IF EXISTS] tbl_name;

6.MySQL 中支持的数据类型

a.数值型：

1>整型：

TINYINT 带符号-128~127 无符号 0~255

SMALLINT

MEDIUMINT

INT

BIGINT

2>浮点类型

FLOAT(M,D):M 代表总长度，D 小数点后几位

FLOAT(6,2):9999.99

浮点数有误差，精度和平台有关

DECIMAL(M,D):M 总长度，D 小数点后几位

内部以字符串形式保存，不涉及误差

b.字符串类型

CHAR(M):定长字符串

VARCHAR(M):变长字符串

TINYTEXT:

TEXT:

MEDIUMTEXT:

LONGTEXT:

ENUM(value1,...):枚举类型，只能从列出来的值中选择一个，最多保存 65535 个值

c.日期时间型

YEAR 保存年份 1 1901~2155

注意：和日期时间有关的，基本都是整型 int 保存时间戳，方便计算

Server characterset: utf8

Db characterset: utf8

Client characterset: utf8

Conn. characterset: utf8

注意：如果有中文的时候需要先临时转换客户端的编码方式

只针对当前连接有效

SET NAMES GBK;

查看状态

\s

status

通过 COMMENT 给字段添加注释

7.数据的操作 DML

1》插入记录 INSERT

INSERT [INTO] tbl_name VALUES(值,...);

2》查询表中的记录 SELECT

SELECT * FROM tbl_name;

SELECT * FROM tbl_name\G;

8.完整性约束条件

CREATE TABLE [IF NOT EXISTS] tbl_name(

字段名称 字段类型 [字段属性 UNSIGNED|ZEROFILL][完整性约束条件]

);

UNSIGNED:被 UNSIGNED 标识的字段，字段值从 0 开始，只能是数值型

ZEROFILL :字段被标识 ZEROFILL,会自动的添加 UNSIGNED,达不到显示长度的时候

以 0 在前面补齐至指定长度

DAY02 数据操作语言

1.字符串类型

CHAR>VARCHAR>TEXT...

2.数据操作语言

INSERT [INTO] tbl_name(字段名称,...) VALUES(值,...)

3.完整性约束条件

CREATE TABLE [IF NOT EXISTS] tbl_name(

字段名称 字段类型 [字段属性 UNSIGNED] [完整性约束条件][NOT NULL][DEFAULT 默认值]

[[PRIMARY] KEY] [AUTO_INCREMENT] [UNIQUE [KEY]]

);

NOT NULL:被标识 NOT NULL 的字段，在插入记录的时候值不能是 NULL，必须要赋值

DEFAULT:某个字段有 DEFAULT 默认值的时候，如果插入记录的时候不给值，会使用

默认值，如果给值就使用赋的值，一般和 NOT NULL 结合起来使用

PRIMARY KEY:主键约束，被标识成主键的字段值禁止重复，自动的禁止为 NULL，

一个表中只可以有一个主键，用来当作唯一标识符。一般主键添加到无意义的字段上，

习惯拿编号当做主键

AUTO_INCREMENT:应用到数值型，而且必须配合唯一来使用，习惯和主键搭配使用

默认起始点是从 1 开始

直接给值

NULL 和 DEFAULT 让其自增，已有编号的最大值+1，AUTO_INCREMENT 的值

SHOW CREATE TABLE tbl_name;

UNIQUE KEY:一个表中只能有一个主键，但可以有多个字段被标识成唯一

被标识成唯一的字段，值不能重复，NULL 是一个特殊的值，他可以重复

4.表结构相关命令

1》动态添加字段 ADD

ALTER TABLE tbl_name

ADD 字段名称 字段类型 [字段属性] [完整性约束条件] [FIRST|AFTER 字段名称];

一次添加多个

ALTER TABLE tbl_name

ADD ...,

ADD ...,

...;

2》删除指定字段 DROP

ALTER TABLE tbl_name

DROP 字段名称;

3》修改字段类型 约束 MODIFY

ALTER TABLE tbl_name

MODIFY 字段名称 字段类型 [字段属性] [完整性约束条件] [FIRST|AFTER 字段名称];

4》修改字段名称 CHANGE

ALTER TABLE tbl_name

CHANGE 原字段名称 新名称 字段类型 [完整性约束条件] [FIRST|AFTER 字段名称];

5》字段添加默认值 ALTER 字段 SET DEFAULT '默认值'

ALTER TABLE tbl_name

ALTER 字段 SET DEFAULT 默认值;

6》删除字段默认值 ALTER 字段 DROP DEFAULT

ALTER TABLE tbl_name

ALTER 字段 DROP DEFAULT;

7》添加主键 ADD PRIMARY KEY(字段名称)

ALTER TABLE tbl_name

ADD PRIMARY KEY(字段名称);

8》删除主键 DROP PRIMARY KEY

ALTER TABLE tbl_name

DROP PRIMARY KEY;

9》添加唯一 ADD UNIQUE KEY(字段名称)

ALTER TABLE tbl_name

ADD UNIQUE KEY|INDEX(字段名称);

添加唯一索引指定索引名称

ALTER TABLE tbl_name

ADD [CONSTRAINT index_name] UNIQUE KEY(字段名称);

如果不加约束名称，默认是以字段名称当做索引名称

10》删除唯一 DROP INDEX index_name

ALTER TABLE tbl_name

DROP INDEX index_name;

11》重名名数据表

ALTER TABLE tbl_name RENAME [TO|AS] new_tblName;

RENAME TABLE tbl_name TO new_tblName;

12》修改自增长的值

ALTER TABLE tbl_name AUTO_INCREMENT =值;

5.数据相关操作(DML):对数据增删改

添加记录：INSERT

INSERT [INTO] tbl_name VALUE|VALUES(值,...);

按照建表时字段顺序一一对应

INSERT [INTO] tbl_name(字段名称,...) VALUES|VALUE(值,...);

一次添加多条记录：

INSERT [INTO] tbl_name[(字段名称...)] VALUES(值,...),(值,...),...;

INSERT SET ...添加记录

INSERT [INTO] tbl_name SET 字段名称=值,字段名称=值,...;

修改记录：UPDATE

UPDATE tbl_name SET 字段名称=值,字段名称=值,...[WHERE 条件];

注意：如果没有添加 WHERE 条件，表中的所有记录都会被更改

删除记录：DELETE

DELETE FROM tbl_name [WHERE 条件];

彻底清空数据表：

TRUNCATE [TABLE] tbl_name;

DAY03 数据查询语言

1.DQL:数据查询语言 SELECT

1》基本查询

SELECT * FROM tbl_name;

*代表的是所有字段，*号效率低

a.查询指定字段的值

SELECT 字段名称,字段名称... FROM tbl_name;

b.给字段起别名

SELECT 字段名称 [AS] 别名,字段名称 [AS] 别名,... FROM tbl_name;

c.给表起别名

SELECT 字段名称 [AS] 别名,... FROM tbl_name [AS] 别名;

d.表名.字段名

告诉字段来自于哪张表

SELECT tbl_name.字段名称,... FROM tbl_name;

e.库名.表名的形式

告诉这个表来自于那张数据库，省去打开指定数据库的操作

2.SELECT 语句

SELECT expr,... FROM tbl_name

[WHERE 条件]

[GROUP BY 分组查询 [HAVING 对分组结果进行二次筛选]]

[ORDER BY 字段名称]

[LIMIT 限制结果集的显示条数]

;

1》WHERE 条件：筛选出符合条件的记录,对记录仅进行第一次筛选

可以作为 WHERE 条件的有：

a.比较运算符

= > >= < <= != <>

<=>:可以检测 NULL 值

b.检测 NULL

IS [NOT] NULL

c.指定范围

[NOT] BETWEEN min... AND max：

d.固定范围

[NOT] IN:IN(值 1,值 2,...)

e.逻辑运算符

and:并且的意思

or:或者的意思

f.实现模糊查询

[NOT] LIKE

%:代表任意字符

_:代表一个任意字符

2》分组查询

GROUP BY 字段名称：把值相同的放到组里

分组查询经常配合 GROUP_CONCAT(字段名称)得到分组详情

配合聚合函数来使用

COUNT():统计记录数

COUNT(*):统计总记录数，* NULL 也统计进去

COUNT(id)：如果有 NULL 值不算一条记录

MAX(字段):求最大值

MIN(..):求最小值

AVG():求平均值

SUM():求总和

HAVING 子句 对分组结果进行二次筛选

3》对结果集进行排序

ORDER BY [ASC 升序] DESC 降序

按照多个字段排序

ORDER BY 字段名称 [ASC 升序],字段名称 DESC

随机打乱查询结果

SELECT 字段名称,... FROM tbl_name ORDER BY RAND();

4》通过 LIMIT 限制结果集的显示条数

LIMIT 是分页的核心

LIMIT 值;->显示前几条记录

LIMIT 偏移量,显示几条记录

SELECT 字段名称,... FROM tbl_name

[WHERE 条件]

[GROUP BY 字段名称 [HAVING 子句对分组结果进行二次筛选]]

[ORDER BY 字段名称 ASC|DESC]

[LIMIT 值[,值]];

ORDER BY 和 LIMIT 可以应用到 UPDATE 和 DELETE

注意：LIMIT 应用于 UPDATE 和 DELETE 只能写 LIMTI 值;

DAY04 数据的增删改查

1.DML:数据的增删改

添加记录：INSERT

INSERT [INTO] tbl_name[(字段名称,...)] VALUES(值,....);

INSERT [INTO] tbl_name[(字段名称,...)] VALUES(值,...),(值,...)...;

INSERT [INTO] SET 字段名称=值,...;

更新记录：UPDATE

UPDATE tbl_name SET 字段名称=值,...

[WHERE 条件] [ORDER BY 字段名称,..] [LIMIT 值];

删除记录：DELETE

DELETE FROM tbl_name

[WHERE 条件] [ORDER BY 字段名称] [LIMIT 值];

ALTER TABLE tbl_name AUTO_INCREMENT=1;

彻底清空数据表：TRUNCATE

TRUNCATE [TABLE] tbl_name;

2.DQL:数据的查询 SELECT

SELECT * FROM tbl_name;

SELECT 字段名称,... FROM tbl_name;

SELECT 字段名称 [AS] 别名名称,... FROM tbl_name;

SELECT 字段名称 [AS] 别名名称,... FROM tbl_name [AS] 别名名称;

SELECT tbl_name.字段名称,... FROM tbl_name;

SELECT alias_name.字段名称,... FROM tbl_name [AS] alias_name;

SELECT 字段名称,.. FROM db_name.tbl_name;

SELECT 字段名称,... FROM tbl_name

[WHERE 条件]

[GROUP BY 字段名称 [HAVING 子句进行二次筛选]]

[ORDER BY 字段名称,.. [ASC]|DESC]

[LIMIT 限制显示条数];

1》WHERE 条件

赋值运算符= <=>,<=>检测 NULL

比较运算符> >= < <= != <>

指定范围：[NOT] BETWEEN min AND max

固定范围：[NOT] IN(值,...)

逻辑运算符：AND OR

实现模糊查询：[NOT] LIKE

%:代表任意多个任意字符

_:代表一个任意字符

2》GROUP BY 分组查询

GROUP_CONCAT(字段名称)得到分组详情

聚合函数：

COUNT(*|字段名称)：

MAX():

MIN():

AVG():

SUM():

HAVING 子句对分组结果进行二次筛选

3》ORDER BY 字段名称 ASC|DESC 排序

ORDER BY RAND( );

4》LIMIT 限制结果集显示条数

实现分页的核心

LIMIT 值; 显示前几条记录

LIMIT 偏移量,显示几条记录;

3.连接查询

1》内连接查询：查询的都是符合两个表中连接条件的记录

[INNER|CROSS] JOIN tbl_name [AS] alias_name

ON 连接条件

[INNER|CROSS] JOIN tbl_name [AS] alias_name

ON 连接条件

...

2》外连接查询

左外联

右外联

第四部分　　PHP 核心数据库

DAY01　MySQL 函数库

1.MySQL 函数库

1》验证 MySQL 函数库是否可以使用

a.phpinfo();

b.extension_loaded($name):检测扩展是否已经开启

c.function_exists($funcName):检测函数名称是否存在

2.MySQL 函数库操作数据库的步骤

1》连接 MySQL--- mysql -hlocalhost -uroot -proot

mysql_connect($host,$username,$password):连接 MySQL 服务器，

连接成功返回的 MySQL 连接标识符(MySQL Link)(是资源)，连接失败返回的是 FALSE

2》设置客户端的编码方式--- SET NAMES GBK;

mysql_set_charset($charset):设置客户端的编码方式，UTF8

mysql_query("SET NAMES UTF8");

3》打开指定的数据库 --- USE dbName;

mysql_select_db($dbName):打开成功返回 true，失败返回 false

4》执行 SQL 查询(语句)-- 写 SQL 语句执行下就可以

mysql_query($sql):执行 SQL 语句

返回值分为两种情况：

a.针对于包含 SELECT/SHOW/DESC/DESCRIBE/EXPLAIN

，执行成功返回的结果集(MySQL Result),执行失败返回 false

b.针对于其他 SQL，执行成功返回 true，失败返回 false

5》如果产生结果集，需要释放结果集

mysql_free_result($result):释放结果集

6》关闭连接 -- exit

mysql_close():关闭连接

3.常用函数

mysql_error():得到上一步操作产生的错误信息

mysql_errno():得到上一步操作产生的错误编号

mysql_insert_id():得到上一步插入操作产生的自增长值，

而且表中必须有字段是自增的，如果没有返回的是 0

mysql_affected_rows():得到上一步操作产生的受影响的记录的条数

和结果集有关的函数：

mysql_fetch_array($result[,$type=MYSQL_BOTH]):取出结果集中的一条记录作为关联+索引的数组返回

直到结果集的末尾，返回 false

$type 值：

MYSQL_BOTH:关联+索引

MYSQL_ASSOC:关联

MYSQL_NUM:索引

mysql_fetch_assoc($result):关联部分

mysql_fetch_row($result):索引部分

得到结果集中记录的条数：

mysql_num_rows($result):取出结果集中记录的条数

mysql_affected_rows():得到上一步操作产生的受影响的记录的条数

$sql="SELECT COUNT(*) AS totalRows FROM tbl_name";

DAY02 通过 PHP 程序写入 Mysql 数据库

王志强老师 qq:634962146 tel:17710268205

mysql 函数库*

连接 mysql

设置交互字符集

选择数据库

执行 SQL 语句

insert

获取主键 id 值

select---结果集资源

获取结果集中的记录，返回的是数组

获取结果集中记录的条数

update

获取影响记录的条数

delete

关闭数据库

文件总结：

login.html---登录表单

login.php---登录表单数据的处理程序

index.php---后台首页

connect.php----数据库连接的公共文件

public

menu.html----左侧管理项页面

welcome.php----后台的欢迎界面

news

add.php---文章添加的表单页面

doAction.php----表单数据处理页面

lister.php----文章列表页面

后台登陆（会员登录）

1、表单

2、数据库表

id username password

create table admin(

id int(5) primary key auto_increment,

username char(20) not null,

password char(32) not null

)

3、有一条记录

insert into admin(username,password) values('admin','e10adc3949ba59abbe56e057f20f883e');

4、写用户名及密码的判断程序

login.php

后台首页：

```
<frameset cols="100,*">
    <frame src="public/menu.html"/>
    <frame name="main" src="public/welcome.php"/>
</frameset>
```

文章后台管理

添加文章

1、文章表

a、这里放什么数据，文章数据。

id 标题 发布时间 内容 作者 点击数 分类

b、创建表

create table news(

id int(8) primary key auto_increment,

title varchar(50) not null,

content text not null,

pubtime int(10),

author char(10),

clicknum int(5) default 0,

type enum('国内','国际','军事','体育')

)

2、表单页面

'' null

3、接收表单数据，写到文章表

insert news() values()

删除文章

http://localhost/psd1501/phpcore/day02/demo/news/delete.php?id=2

接收 id 号

拼 sql，delete

执行

修改文章

1、update.php，修改的表单页面

2、接收 id，把记录的原数据显示在对应的表单元素中

3、给表单加一个隐藏域，来记录被修改文章的 id

4、写表单处理程序，doAction.php

文章列表，分页显示

分页算法：

1、确定每页的条数 $pageSize=5;

2、接收页码值 $page=isset($_GET['page']) ? $_GET['page'] : 1;

3、判断页码值的合法性

31、页码值不能小于 1

```
if($page<1){
    $page=1;
}
```

4、总条数

```
$sql="select count(*) as num from news";
$re=mysql_query($sql);
$arr=mysql_fetch_assoc($re);
$num=$arr['num'];
```

5、得到总页数

```
$pageNum=ceil($num/$pageSize);
```

32、页码值不能大于总页数

```
if($page>$pageNum){
    $page=$pageNum;
}
```

6、算出开始位置

```
$start=($page-1)*$pageSize;
```

7、拼 sql 语句，获取当前页的数据

```
$sql="select * from news limit $start,$pageSize";
```

练习：

```
create database cms;
use cms;
set names gbk;
create table student(
    id int(8) primary key auto_increment,
    name char(10) not null,
    age tinyint(2)
)
insert into student(name,age) values('he1','22');
insert into student(name,age) values('he2','21');
insert into student(name,age) values('he3','20');
insert into student(name,age) values('he4','19');
insert into student(name,age) values('he5','18');
```

```
insert into student(name,age) values('he6','22');

insert into student(name,age) values('he7','21');

insert into student(name,age) values('he8','20');

insert into student(name,age) values('he9','19');

insert into student(name,age) values('he10','18');

insert into student(name,age) values('he11','22');

insert into student(name,age) values('he12','21');

insert into student(name,age) values('he13','20');

insert into student(name,age) values('he14','19');

insert into student(name,age) values('he15','18');
```

分页呈现学生数据，每页 2 条

studentList.php

会话变量 **

gd 库*

文件及目录操作*

oop 面向对象编程 ***

js

jquery

项目

DAY 03　分页与会话变量

后台文章模块

添加

列表

修改

列表页上有一个修改的连接

|

|

id 值

|

修改页面，获取 id 值，根据 id 值，

把记录的原来内容显示在表单中

|

|

数据处理 doAction.php，借助于 update 表名 set 字段='值'

.... where 条件

删除

列表页的分页效果

分页算法：

1、设置每页的条数 $pageSize=5;

2、接收当前的页码值

$page=isset($_GET['page'])?$_GET['page']:1;

3、判断页码值得合法性

31、页码值不小于 1，小于 1，就等于 1

```
if($page<1){
    $page=1;
}
```

4、获取总条数

```
$sql="select count(*) as num from news";
$re=mysql_query($sql);
$arr=mysql_fetch_assoc($re);
$num=$arr['num'];
```

5、获取总页数

```
$pageNum=ceil($num/$pageSize);
```

6、判断页码值，是否大于总页数，大于则显示最后一页

```
if($page>$pageNum){
    $page=$pageNum;
}
```

7、算出开始位置

```
$start=($page-1)*$pageSize;
```

8、拼 sql 语句

$sql="select * from news limit $start,$pageSize";

9、执行

$re=mysql_query($sql);

10、呈现分页效果

会话变量：跨页面访问

cookie 变量

创建

setcookie(name,value,过期时间,作用目录,作用域名,是否为安全连接 0|1);

过期时间：

time()+时间段　　单位秒

声明周期为 30 分钟

time()+1800

0---关闭浏览器，cookie 失效

作用目录

默认：./

/　----　网站所有程序都能访问这个 cookie

作用域名：

localhost

访问：

$_COOKIE[name]

为什么 cookie 可以跨页面访问？？？

工作原理：

1、cookie 值在客户端保存

2、当由这个客服端再次发送请求时，

浏览器自动判断是否把 cookie 值跟上，

如果条件符合，则和请求一起到服务器上，

从而、服务器程序能够再次得到这个值。

解决安全性：

加密

背包算法：0-a 1-b 2-w 3-c 4-n 5-m 6-t 7-i 8-j 9-o

只适合于数字加密，一个数字对应一个字母

$arr=array('a','b','w','c','n','m','t','i','j','o');

123--bwc

加密函数

逐位获取数字，找到数字对应的字母，拼一个新的字符串

解密函数

不用 cookie----session

赋值

覆盖一次

setcookie(name,value,失效时间,path)

删除

setcookie(name,value,time()-1,path)

总结 cookie 的特点：

1、跨页面访问

2、只能存放字符串

3、数据存放在客户端

session 会话变量：

创建

session_start();

$_SESSION['名称']=值;

访问

session_start();

$_SESSION['名称']

赋值

session_start();

$_SESSION['名称']=值;

删除

session_start();

session_destroy();

练习：使用 session，实现后台访问控制。

sess_fne61hv289jttishsiebu2fgr2

fne61hv289jttishsiebu2fgr2---cookie 中的值，名字 PHPSESSID

原理：

客户端和服务器，通过来回传递文件名称，来实现多个页面共享一个 session 文件

缺点：

性能问题。

解决：

内存缓存

session 中，有些配置项：php.ini

session.name = PHPSESSID　指定 cookie 名称

session.cookie_lifetime = 0　指定 cookie 的生命周期

session.cookie_path = /　cookie 的作用目录

session.save_path = "F:/sess"　指定 seesion 文件目录

例子：记录网民的行为

练习：后台访问控制。

作业：解密函数

DAY 04 GD 库

复习：

cookie

创建

setcookie(name,value,生命周期,作用目录);

使用

$_COOKIE['name'] var_dump($_COOKIE);

赋值

setcookie(name,value,生命周期,作用目录);

删除

setcookie(name,value,time()-1,作用目录);

session

创建

session_start();

$_SESSION['name']=值;

使用

session_start();

$_SESSION['name']

赋值

session_start();

$_SESSION['name']=值;

删除

session_start();

session_destroy();

作业：解密

算法：

1、创建一个数组

2、逐位获取字母，找到字母对应数字，拼一个新的字符串

gd 库函数

修改配置文件：

开启 gd

作用：处理图片

应该有哪些函数，对应的实现什么功能。

创建空白图片，产生一个画布

resource imagecreatetruecolor(w,h)

分配颜色

imagecolorallocate($i,r,g,b)

r--0-255

g--0-255

b--0-255

填充颜色

imagefill($i,x,y,颜色)

写英文字母

imagechar($image, $font, $x, $y, $c, $color)

说明：x,y 和字的左上角重叠

imagecharup()

imagestring()

imagestringup()

写汉字

imagefttext($image, $size, $angle, $x, $y, $color, $fontfile, $text);

说明：x,y 和字的左下角重叠

验证码效果：

1、创建一个空白画布

2、分配白色

3、填充背景

4、产生字符库

5、获取 4 位随机字符串

6、把 4 个字符写在图片上，且每次颜色，倾斜角度，位置不同。

7、输出图片

验证码的作用：

防止自动发帖机，自动注册机，发送数据。

画点

imagesetpixel($image,x,y,color)

画线

imageline($image,x1,y1,x2,y2,color)

画椭圆

imageellipse($image, $cx, $cy, $width, $height, $color)

imagefilledellipse($image, $cx, $cy, $width, $height, $color);

画圆弧

imagefilledarc($image,$cx,$cy,$w,$h,$s,$e,$color,$style);

style:

1---起点 终点 圆心三点组成的三角形

2---圆弧

3---起点到终点的直线

4---扇形

练习：画正弦曲线。

画矩形

获取已知图片的大小

array getimagesize($path)

0---width

1---height

mime---类型　image/png　image/gif image/jpeg image/pjpeg

从已知图片创建一个画布

resource imagecreatefromgif(path)

resource imagecreatefromjpeg(path)

resource imagecreatefrompng(path)

练习：

接收图片路径，生成一个新的图片，在页面上输出。

接收图片路径，生成一个新的图片，保存一个新图片，名字及类型和原来一样，目录不同。

图片拷贝

生成或保存图片

imagepng(图片资源[,图片路径及名称])

imagejpeg(图片资源[,图片路径及名称])

imagegif(图片资源[,图片路径及名称])

header("content-type:image/PNG")

DAY05　图片的拷贝

创建画布

分配颜色

填充颜色

写英文

写中文

画点

画线

画圆

矩形

圆弧

图片保存或输出

imagepng()

imagejpeg()

imagegif()

图片的拷贝

imagecopy()

把 800-600.jpg----->03.png 中

```
$sPath="images/800-600.jpg";

$dPath="images/03.png";

$s=imagecreatefromjpeg($sPath);

$d=iamgecreatefrompng($dPath);

imagecopyre($d,$s,0,0,100,200,100,100);

header("content-type:image/PNG");
```

imagepng($d);

把 logo.png---->800-800.jpg，右下角

1、获取小图的宽高 $sw $sh

2、获取大图的宽高 $dw $dh

3、产生目标图片资源 $d

4、产生源图片资源 $s

5、算出目标图片上位置，$dx $dy

6、拷贝

imagecopy($d,$s,$dx,$dy,0,0,$sw,$sh)

7、保存目标图片，类型一样，名称一样，保存路径

//拷贝同时把原图片缩小

imagecopyresized(目标图片资源,源图片资源,

目标图片上 x 坐标,目标图片上 y 坐标,

源图片开始拷贝 x 坐标,原图片开始拷贝 y 坐标,

到目标图片上宽度,到目标图片上的高度,

拷贝原图片的宽度,拷贝原图片的高度)

例子：

把图片 800-800.jpg，放到 200*200 的空白图片中

$sPath="images/800-800.jpg";

$d=imagecreatetruecolor(200,200);

```
$s=imagecreatefromjpeg($sPath);

imagecopyresized($d,$s,0,0,0,0,200,200,800,800);

header("content-type:image/PNG");

imagepng($d);
```

例子：800-600.jpg，放到 200*200 的空白图片中

图片缩放

总结 gd 库函数：

创建空白图片

```
resource imagecreatetruecolor(w,h)
```

分配颜色

```
imagecolorallocate($image,r,g,b);
```

填充颜色

```
imagefill($image,x,y,$color)
```

写英文

```
imagechar($image,$fontsize,x,y,char,color)

imagestring($image,$fontsize,x,y,string,color)

imagecharup($image,$fontsize,x,y,char,color)

imagestringup($image,$fontsize,x,y,string,color)
```

写汉字

```
imagefttext($image,size,angle,x,y,color,fontfile,text);
```

画点

setpixel($image,x,y,color);

画线

imageline($image,x1,y1,x2,y2,color);

画椭圆

imageellipse($image,cx,cy,w,h,color)

imagefilledellipse($image,cx,cy,w,h,color);

画圆弧

imagefilledarc($image,cx,cy,w,h,start,end,color,style)

画矩形

imagerectangle($image,x1,y1,x2,y2,color);

imagefilledrectangle($image,x1,y1,x2,y2,color);

由已知图片，创建画布

imagecreatefromjpeg($path)

imagecreatefromgif()

imagecreatefrompng()

图片的拷贝

imagecopy($d,$s,$dx,$dy,$sx,$sy,$sW,$sH)

imagecopyresized($d,$s,$dx,$dy,$sx,$sy,$dW,$dH,$sW,$sH)

图片输出或保存

imagepng($image[,$path])

imagejpeg($image[,$path])

imagegif($image[,$path])

文件及目录操作

文本文件操作

打开文件，同时创建文件

resource fopen(文件路径,打开方式)

打开方式：

r----read　文件必须存在

w----write　文件不存在，自动创建,自动清空文件

a----append　文件不存在，自动创建

写文件

int fwrite($f,写的内容);//utf-8 gbk

读文件

string fread($f,length)

每次读一行

string fgets($f)

每次读一行，把 html 标签过滤掉

string fgetss($f)

判断文件是否存在

bool is_file(path)

获取文件的大小

int filesize(path)

删除文件

unlink(path)

关闭文件

bool fclose($f);

目录的操作

判断目录是否存在

is_dir(path)

创建

bool mkdir(path,0665,true)

7=1+2+4

1---执行

2---写

4---读

打开

resource opendir(path)

读

string readdir($d)

关闭

closedir($d)

删除空目录

rmdir(path)

目录的递归遍历？？？？？

作业：

验证码

水印

缩略图

DAY06　文件操作函数　正则表达式

文件操作函数

bool is_file(path)

int filesize(path)

resource fopen(path,r|w|a)

string fread($f,length)

string fgets($f)

string fgetss($f)

fclose($f)

int fwrite($f,content)

string file_get_contents(path|url)//获取文件的内容

int file_put_contents(path,content)//写文件

目录操作函数

bool is_dir(path)

bool mkdir(path,0777,true)

opendir(path)

readdir(resource)

closedir(resource)

array scandir(path) 获取某目录下的所有内容

rmdir(path)

目录的递归遍历？？？？？

遇到目录自己调用自己

遍历两级目录：

先打开 a，遍历 a

遇到目录继续打开，遍历

```
$d=opendir("a");
while($dir=readdir($d)){
    if($dir!='.' && $dir!=".."){
        echo $dir,"<br/>";
    }
}
```

正则表达式

作用：去字符串中，找我们想要的内容。

$str="dfsadf7987809dsafd8790dfsa709df87s";

原理：使用了一个有规则的式子，用这个式子去字符串中比对。

正则表达式的写法：

由字符簇，限定符，定位符等组成的式子。

字符簇：字符集合，规定了找什么。

写法：[0123456789]

[0-9]

[a-z]

[A-Z]

[abc]

\d----[0-9]

\w----[a-zA-Z0-9_]

.-----任意字符 ，换行除外

限定符：确定了找的个数。作用：找几个。

{m,n} 找 m 到 n 个

\d{1,5}

{m,} 找 m 个以上

\d{1,}

{0,n} 找 n 个以内

{m} 找 m 个

? {0,1}

+ {1,}

* {0,}

正则表达式的写法：

"/字符簇限定符定位符/"

写出几个正则表达式：

1、找 3 位数字 demo-1.php

"/[0,9]{3}/"

int preg_match_all(正则表达式,字符串,$arr)

preg_match()

2、找 1 到 5 个数字

"/\d{1,5}/"

3、找出字符中小写英文字母

"/[a-z]+/"

4、找出字符串中符合要求”两个字母后有三个数字“的字符串。 demo-3.php

$str="dsafdds79dfsaf78908dfasfd8908080";

"/[a-zA-Z]{2}\d{3}/"

5、找出字符串中的正整数 demo-3.php

123

+123

"/\+?\d+/"

或 |

匹配 x 或 y 或 z

"[xyz]"

"x|y|z"

例子：

找字符串中 abc 或 bcd 或 www

[abc]{3}|[bcd]{3}|[w]{3}

例子：

获取图片名称中扩展名

\.jpg|\.gif|\.png

子表达式:加了括号的表达式，称作子表达式。

特点：子表达式后可以使用限定符，规定子表达式出现的次数

例子：

找符合要求”两位数字一个 a 或一个 b 或一个 c 三个数字“的字符串

"/\d{2}[abc]\d{3}/"

"/\d{2}a|b|c\d{3}/" demo-4.php

"/(a|b|c){3}/"=====/[abc]{3}/"

邮箱正则表达式：

beifangdelang107@163.com

_abc@163.com

a.b.c@163.com

邮箱名称：2 为以上英文字母数字下划线中线.组成

域名：数字英文字母-组成，1 位以上

后缀：英文字母.,两位以上。

.cn

.com

.com.cn

[0-9a-zA-Z_\-\.]{2,}@[0-9a-zA-Z\-]+(\.[a-zA-Z]{2,}){1,2}

定位符：匹配位置

^----字符串的开始

$----字符串的结尾

应用场景：

1、从字符串开始匹配到结束，匹配一次 ^ $

2、从字符的开始匹配 ^

判断一个 sql 语句是否为 select 语句

if(preg_match_all("/^select/",$sql)){

```
}
```

3、匹配到字符串的结束 $ path.php

判断一个路径，否是一个图片路径。

.png .gif .jpg

```
$path="a/b/c/aa.JPg";
if(preg_match_all("/\.png$|\.gif$|\.jpg$/",$path,$arr)){
    var_dump($arr);
}
```

匹配模式

/正则表达式/i i 的作用是，匹配时，不区分大小写

/正则表达式/s s 的作用是，. 这个字符串包含 换行。

```
$str="
<p>p-1</p>
<h1>tttt-1</h1>
<div>div-1</div>
";
preg_match_all("/<h1>.*</h1>/",$str,$arr);
```

.*的使用

特点：有贪婪性

.*? 可以使.*去除贪婪性

练习：

获取页面中 style 标签中样式语句，

把这些样式语句拼在一起，然后在文件 style.css 中。

```
$arr=array('a','b','c');

foreach($arr as $v){

    $str.=$v;

}
```

正则表达式相关函数：

int preg_match_all(正则表达式,字符串,$arr)

int preg_match(正则表达式,字符串,$arr)

array preg_split(正则表达式,被拆分的字符串)

例子：根据数值把这个字符串"ou45325uou54321jhl4j23aaa"，拆分

```
$str="ou45325uou54321jhl4j23aaa";

$arr=preg_split("/\d+/",$str);

var_dump($arr);
```

string preg_replace(正则表达式,替换成什么,谁被操作了)

例子：把字符串$str，中的数值替换成"W"

```
echo preg_replace("/\d+/","W",$str);
```

反向引用：第一个参数中子表达式获取的内容，给第二个参数用。

反向引用变量：第一参数中的子表达式，匹配到的内容会自动放在变量

$1 $2 $3 ..... 中，$1,.... 被称作反向引用变量。

例子：$d="03/25/2015";//---->2015-03-25　$3-$1-$2

echo preg_replace("/(\d{2})\/(\d{2})\/(\d{4})/","$3-$1-$2",$d);

练习：把字符串"abcdef",使用反向引用翻转 成”fedcba“

echo preg_replace("/([a-z])([a-z])([a-z])([a-z])([a-z])([a-z])/","$6$5$4$3$2$1","abcdef");

常用的正则表达式：

邮箱正则表达式

"/^\w+@[a-z0-9\-]+(\.[a-z]+)+$/i"

座机号

4 位-7 位

3-8 位

/^(\d{4}[\-]\d{7}|\d{3}[\-]\d{8})$/

手机号

1

3 5 7 8

9 位 数字

/^1[3578]\d{9}$/

qq 号

/^\d{5,13}$/

邮编

/^\d{6}$/

用户名:由英文字母数字下划线组成，且开头不能是数字 4 到 20 位

/^[a-z_]\w{3,19}$/i

密码：由英文字母数字下划线组成@#￥%^&*等组成，6-20 位

/^.{6,20}$/

作业：

目录递归遍历

获取页面中 style 标签中样式语句，把这些样式语句拼在一起，然后在文件 style.css 中。

PHP 核心数据库总结

phpcore

mysql 函数库

resource mysql_connect(host,username,password);

bool mysql_select_db(dbName)

mysql_set_charset("utf8|gbk")

mixed mysql_query($sql);

select 字段列表 from 表名

where 条件

group by 字段名称

having 条件

order by 字段 desc|asc

limit 开始位置,长度

insert into 表名(字段列表) values(值列表)

delete from 表名 where 条件

update 表名 set 字段=值,.... where 条件

mysql_fetch_assoc(结果集资源)

mysql_fetch_array(结果集资源)

mysql_fetch_row(结果集资源)

mysql_affected_rows()

mysql_num_rows();

mysql_insert_id();

mysql_close();

会话变量

特点：跨页面访问

创建

setcookie(name,value,过期时间,作用目录)

使用

$_COOKIE['name']

修改

setcookie(name,value,过期时间,作用目录)

删除

setcookie(name,value,time()-1,作用目录)

session

创建

session_start();

$_SESSION['name']=value;

使用

session_start();

$_SESSION['name']

修改

session_start();

$_SESSION['name']=value;

删除

session_start();

session_destroy();

应用场景：

一个数据在 a 页面产生，b 等页面使用。

验证码

后台访问控制

gd 库函数

创建空画布

$i=imagecreatetruecolor(w,h)

分配颜色

imagecolorallocate($i,r,g,b);

填充颜色

imagefill($i,x,y,color);

写字

imagechar($i,size,x,y,char,color)

imagecharup($i,size,x,y,char,color)

imagestring($i,size,x,y,char,color)

imagestringup($i,size,x,y,char,color)

imagefttext($i,size,angle,x,y,color,fontfile,text);

画点

imagesetpixel($i,x,y,color)

画线

imageline($i,x1,y1,x2,y2,color);

画椭圆

imageellipse($i,cx,cy,w,h,color);

imagefilledellipse($i,cx,cy,w,h,color);

画圆弧

imagefilledarc($i,cx,cy,w,h,s,e,color,style);

画矩形

imagerectangle($i,x1,y1,x2,y2,color)

imagefilledrectangle($i,x1,y1,x2,y2,color)

由已知图片创建画布

imagecreatefromgif(path)

imagecreatefrompng(path)

imagecreatefromjpeg(path)

图片的拷贝

imagecopy($d,$s,$dx,$dy,$sx,$sy,$sw,$sh);

imagecopyresized($d,$s,$dx,$dy,$sx,$sy,$dw,$dh,$sw,$sh);

生成图片

imagepng(resource[,path])

imagegif(resource[,path])

imagejpeg(resource[,path])

案例：

验证码

1、创建画布

2、分配颜色，填充

3、产生字符库

```
$str=join('',array_merge(range(0,9),range('a','z'),range('A','Z')));
```

4、打乱

5、取前四位

6、把四位字符串放到 session 中。

7、把文字逐个写在画布上。

```
for($j=0;$j<strlen($str);$j++){
    //字符
    $char=$str{$j};
    //随机颜色
    $color=imagecolorallocate($i,mt_rand(0,255),mt_rand(0,255),mt_rand(0,255));
    //随机角度
    $angle=mt_rand(-15,15);
    //位置
    $x=5+$j*20;
    $y=25;
    //写字
    imagefttext($i,20,$angle,$x,$y,$color,"",$char);
}
```

8、干扰元素。

9、输出图片。

水印

function water($dPath,$sPath,$position,$toPath){

1、获取原图片信息

2、获取目标图片信息

3、根据类型，产生一个原图片资源

4、根据类型，产生一个目标图片资源，产生保存图片的函数名称

5、根据$position 的值，产生水印的位置 x ,y

6、拷贝

7、获取目标图片的名称

8、保存

}

缩略图

function zoom($sPath,$toW,$toH,$toPath,$prefix="s_"){

1、产生一个空白画布，目标图片

2、获取原图片信息

3、根据类型创建原图片资源，产生保存图片的函数名称

4、确定缩放的比例

5、原图片缩放后的大小

6、算出原图片缩放后，放到目标图片上的位置

7、开始拷贝

8、获取原图片的老名称

9、拼新名称。

10、保存

}

文件函数

文件是否存在

bool is_file(path)

bool file_exists(path)

目录存在，也返回 true

打开文件

resource fopen('a/a.txt','r|w|a');

获取文件大小

filesize(path);

读文件

fread($f,length)

fgets($f)

fgetss($f);

写文件

int fwrite($f,content);

关闭

fclose();

string file_get_contents(path|url)

int file_put_contents(path,content);

目录操作函数

mkdir(path,0777,true)

rmdir(path)

resource opendir(path)

readdir($d)

closedir($d)

array scandir(path)

目录递归遍历：

```
function getDir($path){
    if(is_dir($path)){
        $d=opendir($path);
        static $arr=array();
        while($dir=readdir($d)){
            if($dir!='.' && $dir!='..'){
              $newPath=$path."/".$dir;
              $arr[]=$newPath;
              if(is_dir($newPath)){
                getDir($newPath);
              }
```

```
                }
            }
        }else{
            return false;
        }
        return $arr;
    }
```

正则表达式

第五部分　　面向对象编程 OOP

DAY01　类的封装

面向对象编程：

对象：万事万物皆为对象。

所有的人：类

类的封装

语法格式：

```
class  类名{
    属性
    public|protected|private $属性名称=值;
    方法
    public|protected|private function  方法名称(参数,.....){
        方法体;
        返回值;
    }
}
```

public 公共 ,可见度最大 ，通过对象 ，类的内部，可以被继承

protected 受保护的，可见度稍小，对象不可见，类的内部，可以被继承

private 私有的，可见性最小，对象不可见，类的内部可见，不可以被继承

例子：人类

```
class People{
    private $money=0;
    public function setMoney($m){
```

```
            $this->money=$m;
        }
        public function eat($price){
            if($this->money>=$price){
                $this->money=$this->money-$price;
                return true;
            }else{
                return false;
            }
        }
        public function work($days){
            //每天 100
            return $this->money=$days*100
        }
}
```

对象的实例化

```
$ob=new  类名();
$ob=new  类名;
$ob=new People();
//吃
$re=$ob->eat(0.5);
var_dump($re);
```

例子：声明一个学生类。

属性：

fenshu=0;

方法：

学习($days)

每学一天，赚 10 分

玩($days)

每玩一天，分数减 10

总结：

属性的特点：

1、属性用来存放数据。

2、属性对于所有的方法，都可见。

3、属性和变量是不一样的。

4、属性的调用

方法中：$this->属性名=值;

对象：$ob->属性名=值;

5、属性的可见性 public 不能省略。如果省略必须加上 var

方法的特点：

1、方法的可见性 public 可以省略。

数据库操作类：

属性

数据库连接资源

方法：

连接数据库

特点：

一定在其他方法之前调用

只调用一次

执行 sql 语句

关闭数据库连接

特点：

一定在其他方法之后执行

执行一次

页面：文件列表，显示 news 表中，前 6 条记录

类----数据库操作类

构造方法

特点：

自动执行，类实例化成功后。

只调用一次

传参跟在类名后

返回值没有意义

function __construct(参数,....){

}

function 类名(参数,....){

}

析构方法

特点：

对象销毁之前调用，且调用一次。

析构方法没有参数，返回值没有意义

function __destruct(){

方法体;

}

作业：

实现数据库操作类。

使用数据库操作类，实现文章添加。

DAY02　继承

作业：

实现数据库操作类。

使用数据库操作类，实现文章添加。

复习：

封装类

class 类名{

属性

public|protected|private $属性名=值;

方法

public|protected|private function 方法名称(参数,..){

方法体;

返回值;

}

构造方法

public function __construct(参数,....){

}

析构方法

public function __destruct(){

}

}

方法及属性的调用

类的外部：

$ob=new 类名();

$ob->属性名;

接收返回值 = $ob->方法(参数);

方法中调用属性：

$this->属性名称=值;

$this 中放的是类的当前对象。*****

文章类，对表 news 进行操作

行为：

添加文章

修改文章，根据文章 id

删除文章，根据文章 id

获取一条记录，根据文章 id

获取文章列表，带分页

获取文章的总条数

select count(*) as num from news

select count(*) as num from news where type='国内'

select count(*) as num from news where pubtime>=1441234567

select count(*) as num from news where pubtime>=1441234567 and type='国内'

会员类，对表 user 进行操作

行为：

添加会员

判断用户名及密码是否正确

查询会员，呈现列表

获取记录总数

根据 id，获取会员信息

继承：子类去继承父类（基类）的属性及方法。

语法格式：两个

class 子类 extends 父类{

自己的属性

从父类继承的属性

自己的方法

从父类继承的方法

}

例子：父类的那些属性及方法会被继承。

```
class f{
    public $a=1;
    protected $b=2;
    private $c=3;
    function __construct(){
        echo "f_gouzao";
    }
    function fun1(){
        echo "f_fun1";
    }
    function fun4(){
        echo $this->c;
    }
    protected function fun2(){
        echo "f_fun2";
    }
    private function fun3(){
        echo "f_fun3";
    }
}
```

```
class s extends f{



}
```

继承的特点：

1、可见性为 public 的属性及方法可以被继承。

2、构造方法可以被继承，在子类中依然是构造方法。

3、可见性为 protected 的属性及方法会被继承。

4、可见性为 private 的属性及方法不会被继承。

5、私有的属性及方法可以通过从父类继承的方法访问。

覆盖：其实就是子类和父类共用了一个空间。

如果子类中定义一个和父类一样名称的属性或方法，会覆盖。

6、覆盖父类的属性及方法时，可见性应该高于等于原来的属性及方法。

```
class f{
    public $a=1;
    protected $b=2;
    private $c=3;
    function fun1(){
        echo $this->a;
    }
    function fun2(){
```

```
            echo $this->b;
        }
        funciton fun3(){
            echo $this->c;
        }
    }
    class s extends f{
        public $a=4;
        protected $b=5;
        public $c=6;
        function sfun1(){
            echo $this->a;
        }
        function sfun2(){
            echo $this->b;
        }
        funciton sfun3(){
            echo $this->c;
        }
    }
    $ob=new s();
    $ob->fun1();//4
```

$ob->fun2();//5

$ob->fun3();//3

$ob->sfun1();//4

$ob->sfun2();//5

$ob->sfun3();//6

数据模型类：表操作类，实现基类

增加

修改

删除

查询

Model.class.php

DAY 03　静态类

作业：

类的继承：

class 子类名称 extends 父类名称{

　　父类的属性

　　子类自己的属性

　　父类的方法

　　子类的方法

}

特点：

1、父类中可见性为 public protected 的属性和方法可以被继承

2、private 的属性及方法不能被继承。

3、如果子类中属性及方法和继承来的重名，会发生覆盖。

4、覆盖时，可见性只能越来越高。

知识点：

配置文件

域名

网站根目录

数据库连接信息

类的自动加载

类自动加载函数

```
function __autoload($className){
    //由类名拼出类文件名
    //类文件放到固定目录下
    $className.'.class.php';
}
```

类的实例化 new 时，extends .....

魔术方法：

function __construct()

function __destruct()

function __set($name,$value)//当给一个未定义的属性赋值时，自动执行

function __get($name)//去读一个不存在的属性值时，自动执行

function __call($funName,$args)//调用一个未定义的方法时，call 自动调用

function __clone()//对象被克隆

function __sleep()

function __wakeup()

类常量

常量只能被调用，不能被修改。

语法格式：

const 常量名称=值;

类内部使用：

类名|self::常量名称

类的外部：

类名::常量名称

静态类，静态属性，静态方法

静态属性：

public|protected|private static $属性名=值;

特点：属于类，直接通过类名或 self 调用

public|protected|private static function 方法名(参数,){

方法体;

}

特点：属性类

调用静态属性及方法：

类名::$属性

类名::方法();

例子：Tests.class.php

案例：静态类不用实例化对象

验证码类：

显示验证码

静态属性及静态方法的特点：

1、属于类

2、通过类名或 self,直接访问

3、静态方法中，不能使用$this

4、静态类中，可以有非静态方法及属性。

5、非静态方法，可以调用，静态属性或静态方法。

单例模式

作用：在程序运行过程中，能够实现，只有一个对象存在。

做到：

1、不能在类外部，使用 new 实例化对象。

2、能够产生一个，且只有一个。

static function getInstance(){

}

3、对象不能被克隆。

DAY04　类的自动加载

类的自动加载

原理：当由 new extends :: 执行时，函数__autoload()自动执行，

且把类名以参数形式，传给 autoload

实现：

```
function __autoload($className){
    //B.class.php B     Abc.class.php Abc
    require(ROOT."/class/".$className.".class.php");
}
```

静态属性静态方法

1、静态属性及静态方法属于类

2、静态属性和静态方法的调用，不需要对象。

3、普通方法中，可以调用静态属性及方法。

语法：

静态属性

可见性 static $属性名=值;

调用

类名::$属性名

self::$属性名

静态方法：

可见性 static function 方法名称(参数,....){

//不能使用$this

self

}

类常量

语法：

const PI=3.14;

使用：

self|类名::PI;

单例模式

三个要求：

1、不能再类的外部实例化对象

2、有一个静态方法，能够产生一个对象，且只产生一个。

3、这个类产生的对象，不能被克隆

```
class Danli{
    private static $myOb="";
    private function __construct(){
```

```
    }
    static function getInstance(){
        if(!self::$myOb instanceof self){
            $myOb=new self();
            self::$myOb=$myOb;
        }
        return self::$myOb;
    }
    private function __clone(){

    }
}
```

产品模块

产品发布

产品列表（查询）

产品修改

产品删除

步骤：

1、新建表,手机

a、确定字段

```
create table product(
    id int(8) primary key auto_increment,
    name varchar(20) not null,
    price varchar(10),
    userprice varchar(10),
    content text,
    image1 varchar(40),
    image2 varchar(40),
    image3 varchar(40),
    image4 varchar(40),
    color varchar(10),
    brand varchar(8) not null,
    model varchar(20) not null,
    state tinyint(1) default 0,
    ftime date
)
```

2、封装类

继承 Model.class.php

3、实现页面功能。

a、登录

封装 Admin.class.php

b、后台的首页

c、产品添加 管理 删除 修改

管理：

oper.php

使用方法 getProductByList()

遍历数组，呈现列表

图片上传类

方法：

构造方法

指定图片类型，大小，保存的目录

命名

为图片重新起个名字，用到原名称

保存图片

判断类型及大小

产生名字

开始保存

分页类：

1、确定每页的条数

2、接收页码值

3、判断页码值得合法性

4、判断是否小于 1

5、总条数　产品类中的方法 getProductCount(),交给分页类

6、总页数

7、判断页码值是否大于总页数

8、算出开始位置

9、分页类提供一个开始位置值，后获取当前页的数据　产品类

10、显示分页效果

方法：

构造方法($pageSize,$num)

getStart();//返回当前页的起始点

showPage();//显示分页效果

DAY05　抽象类　接口

分页类

1、把所有的页码值，都显示出来

2、每次显示 5 个，和当前页有关系。前后个显示两个。

分析：

当前页 1　前面不够显示 从 1 开始显示，显示 5 个

当前页 2　前面不够显示 从 1 开始显示，显示 5 个

当前页 3　前面够显示，开始值 当前页的页码值-2

当前页 4 前面够显示，开始值 当前页的页码值-2

.....

当前页 49 后面够显示，开始值 当前页的页码值-2

当前页 50 后面够显示，开始值 当前页的页码值-2

当前页 51 后面不够显示,从 总页数-5 52-5+1=48

.....

四种情况

1、总页数小于等于 5 个，无条件都显示

2、总页数大于 5

a、前面不够显示

```
if($this->page-2<=0){
    //从 1---5
}
```

b、后面不够显示

```
else if($this->page+2>$this->pageNum){
    //从 总页数-5+1 开始，显示 5 个
}
```

c、前后都够显示

```
else{
    //从$this->page-2 到 $this->page+2
}
```

产品修改

1、有一个表单页面，先把原数据显示出来

2、提交进行数据处理，图片修改

图片修改

1

2----重新选择--文件保存--删除老图片--image2

抽象方法,抽象类：

没有方法体的方法；

```
abstract class People{
    public|protected abstract function work();
}
```

子类中，去重写：

```
class Phpteacher extends People{
    function work(){
        echo "每天讲解 php 知识，8 个小时";
    }
}
```

特点：

1、不能直接实例化

2、抽象方法的可见性不能为 private

3、抽象方法，在子类中必须重写。

接口：一个团队中，成员与成员间的一个接口。

interface 接口名称{

//抽象方法

function 方法名称(参数 1,....);

}

大拿去写接口，工程师，继承接口封装类，程序猿调用类

练习：给产品类写一个接口文件

设置自动加载目录

set_include_path(path)

set_include_path("F:/test/psd1501/oop/class;F:/test/psd1501/oop/interface")

PATH_SEPARATOR 分隔符 window ; linux :

得到自动加载目录

get_include_path()

缩略图类

方法：

生成缩略图的方法

根据图片的路径，获取宽 高 类型 资源变量 保存图片函数名称

水印类

方法：

获取图片信息

加水印，生成水印图片

DAY06 PDO 类

PDO 类

作用：用来操作数据库

Mysql.class.php

学习：如何使用，如何调用属性，方法。

extension=php_pdo.dll 5.2 以前版本

extension=php_pdo_mysql.dll

PDO 的实例化：

new PDO();

连接数据库

function __construct("mysql:host=localhost;dbname=cms","username","password")

执行 sql 语句

$ob->exec("set names utf8");

作用：设置交互字符集

exec()

作用：用来执行 update insert delete 语句

返回值：影响记录的条数

query()

作用：执行查询语句

prepare()

作用：准备执行

获取最后一个 insert 语句，产生的主键 id 值

$ob->lastInsertId();

PDOStatement

bool setFetchMode(PDO::FETCH_ASSOC|PDO::FETCH_NUM|PDO::FETCH_BOTH)

作用：设置 fetch() fetchAll()方法返回数组的类型

获取记录

fetch()

特点：每次指针下移，当读完最后一条，再次执行返回 false

fetchAll()

特点：获取全部记录，返回二维数组

练习：

呈现文章列表，不带分页。

sql 注入：

通过改变传值，从而改变 sql 语句的作用。

select * from news where id=5 union select * from admin limit 1,1

$PDOStatement=$pOb->prepare($sql)

作用：准备执行，sql 语句

说明：

$sql 是一个带有占位符的 sql 语句。

select * from news where id=?

占位符的写法：

:名称

?

$PDOStatement->bindParam(占位符的名称,对应的数据变量,类型)

说明：

如果占位符是？ ,第一个参数应该为 位置值，从 1 开始。

作用：绑定值

PDO::PARAM_INT //整型

PDO::PARAM_STR //字符串

$PDOStatement->execute()//select * from news where id=5

作用：最终执行 sql 语句

获取数据：

$PDOStatement->setFetchMode()

$PDOStatement->fetch()

$PDOStatement->fetchAll()

sql 语句的那些位置可以使用占位符？？？？

select 字段列表 from 表名

where 字段名 比较运算符 占位符

group by 字段名称

having 字段名 比较运算符 占位符

order by 字段 desc|asc

limit 占位符,占位符

insert into 表名(字段 1,字段 2,.....)

values(占位符,占位符,.....)

update 表名 set 字段=值,....

where 字段名 比较运算符 占位符

delete from 表名 where 字段 比较运算符 占位符

练习：实现删除一篇文章，根据 id，使用 pdo 的预处理功能。

OOP 总结

封装类，调用类

语法格式：

```
class 类名{
    属性
    public|protected|private $属性名=值;
    public|protected|private static $属性名=值;
    const 常量名称=值;
    方法
    可见性 function 方法名(参数,){
        方法体;
        $this->普通属性;
        self::$属性名;
        self::常量名称
        $this->方法();
        self::方法名();
        返回值;
```

}

可见性 static function 方法名(参数,..){

静态方法

静态的属性

}

魔术方法：

__get($varName)

__set($varName,$value)

__call($funName,$args)

__construct(自定义)

__destruct()

__clone()

}

如何封装类：

类的作用---》行为---》封装---》产生属性

Mysql.class.php

类的继承：子类继承父类的非私有属性及方法。

作用：把几个类的公共属性及方法，封装给父类。

语法：

class 子类 extends 父类{

}

Model.class.php

News.class.php

User.class.php

Product.class.php

Admin.class.php

单例模式：Mysql.class.php

1、类的外部不能使用 new 实例化对象

2、有一个静态方法，产生一个对象，且只有一个。

3、对象不能被克隆

Mysql.class.php Water.class.php Page.class.php Thumb.class.php Upload.class.php

|

|

Model.class.php

|

Admin.class.php News.class.php Product.class.php

类的自动加载

set_include_path();

PATH_SEPARATOR

```
function __autoload($className){
    require ROOT."class/".$className.".class.php";
}
```

抽象类

接口

PDO 数据库操作类

PDO

属性：

PDO::PARAM_STR

PDO::PARAM_INT

PDO::FETCH_ASSOC

PDO::FETCH_NUM

PDO::FETCH_BOTH

方法：

PDO->__construct()

PDO->exec()

PDO->query()

PDO->prepare()

PDO->lastInsertId()

PDOStatement

方法：

PDOS->setFetchMode();

PDOS->fetch();

PDOS->fetchAll();

PDOS->bindParam()

PDOS->execute()

第六部分 JAVASCRIPT

DAY01 Js 基础

javascript:脚本语言，运行在客户端。html css javascript

特点：存放在服务器上，浏览器执行时，需要下载。

<script type="text/javascript" src="js 文件路径"></script>

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

js 基础知识：

基本语法

1、每行代码后可以有分号，可以没有。

2、js 严格区分大小写

3、单行注释 //

多行注释 /**/

4、排错

打开错误控制台

google F12

火狐 工具--》web 开发者---》web 控制台

变量类型

字符串

整型 浮点 数值

布尔

对象

null

NaN ---- not a number

"a123"-->NaN

undefined

变量

var 变量名称=值;

命名规则：

1、开头不能是数字，可以是_ 字母 $

2、不能使用关键字

function class var

3、可以使用小驼峰，大驼峰命名法

字符串变量

var str='';

var str="";

var str=new String();

\\ \" \' \n

使用

str

赋值

str=值;

销毁

关闭浏览器，自动销毁

数值变量

var num=3.14;

var num=new Number(3.14);

bool 变量

var b=true|false;

var b=new Boolean(true|false);

数组：

var arr=new Array(1,"a",'n');

var arr=Array();

var arr=[1,"a",'n'];

说明：不能指定下标，默认索引下标。

使用：

arr[下标]

赋值：

arr[下标]=值;

类型转化

字符串---》number

规则：从非数值字符开始，全部舍弃，如果第一字符就是非数值，则返回 NaN

判断是否为 NaN,使用函数 isNaN()

----》Array

规则：把字符串当做一个元素

获取变量类型可以使用函数 typeof()

----》bool

规则：''---->false　'非空'--->true

number---->string

规则：类型变量，内容不变

----->bool

规则：0---false 非 0---true

0.0

----->Array

规则：把字符串当做一个元素

array---->string

规则：把所有的元素拼接成一个字符串，用逗号作为分隔符

---->bool

规则：true

---->number

规则：把数组转成字符串，到 number

运算

算术运算

+ - * / %

字符串的拼接

+　参与运算的变量，其中一个为字符串，拼接

比较运算

> >= < <= == != !== ===

逻辑运算

&& || !

赋值运算

=

对象方法及属性的调用

.

一元

i++

i--

++i

--i

二元

+=

-=

*=

/=

%=

三元

var 变量名=表达式 ？ 值 1： 值 2;

流程控制语句

条件语句

if(表达式){

}

if(表达式){

}else{

}

if(表达式 1){

}else if(表达式 2){

}

....

else{

}

选择语句

```
switch(变量){
    case  值 1:
        语句;
        break;
    .....
    default :
        语句;
        break;
}
```

循环语句

```
for(var i=初值;循环条件;变换步长){


}
```

```
初值;
while(循环条件){
    变换步长
    break;
    continue;
}
```

遍历数组

```
for(var k in  数组){
```

数组[k]

}

例子：输出如下效果

```
*
***
*****
*******
```

例子：输出如下效果

```
   *
  ***
 *****
*******
```

例子：九九乘法表

```
document.write("");
```

基本函数库

字符串操作

长度

字符串变量.length

获取某位置字符

char 字符串变量.charAt(下标)

截取

string 字符串变量.substr(开始位置[,长度])

替换

string 字符串变量.replace(找什么,替换成什么);

特点：只处理一个，都处理掉，第一个参数使用正则表达式/正则表达式/g

查找

int 字符串变量.indexOf(找什么)

```
if(re!==-1){
    alert('zhaodao');
}
```

int 字符串变量.lastIndexOf(找什么)

拆分

array 字符串变量.split("分隔符")

说明：分隔符可以是一个正则表达式

大小写转化

字符串变量.toUpperCase()

字符串变量.toLowerCase()

array str.match(正则表达式)

时间日期函数

```
var d=new Date();
```

d.getFullYear()//四位的年

d.getMonth()+1 //0-11

d.getDate()//1-31

d.getHours()//0-23

d.getMinutes()//0-59

d.getSeconds()//0-59

d.getDay() //星期几 0-6

d.toLocaleString();

数值函数

Math.round(number)

Math.ceil(number)

Math.floor(number)

Math.random()　验证码刷新

Math.abs()

Math.pow()

Math.sqrt()

数组函数

获取长度

array.length

追加

int array.push(var1,var2,....)

int array.unshift(var1,var2,....)

删除

mixed array.pop()

mixed array.shift()

替换

array.splice(开始位置,长度[,替换成什么 1,....])

拼接

array array.concat(otherArr,....)

查找

index array.indexOf(找什么)

index array.lastIndexOf(找什么)

排序

```
array.sort(function(a,b){
    return a-b;
})
```

正则表达式函数

bool /正则表达式/.test(被匹配的字符串)

例子：匹配一个字符串，是否符合手机号码格式

```
if(/^1[3578]\d{9}$/.test("13838385438")){
    document.write("是");
}else{
    document.write("不是");
}
```

自定义函数

```
function  函数名(参数,....){
    函数体;
    返回值;
}
```

说明：

函数体外部的变量，有全局性

函数体中，如果不加 var ,全局变量，加了，局部变量

获取两个数中的大者？

```
function getMax(num1,num2){
    if(num1>num2){
        return num1;
    }else{
        return num2;
    }
}
```

getMax(100,5);

作用：实现单词首字母大写的函数。

i like php---->I Like Php

DAY02　事件

事件

document.write("内容");

var ob=document.getElmentById("标签的 id 属性值");//找到标签

ob.innerHTML="内容"//操作标签的内容

作用：通过一个事件的发生，来启动一个 js 语句或函数的执行。

html 标签的事件属性：

光标事件

得到光标

onfocus

失去光标

onblur

内容发生变化

onchange

鼠标事件

单击事件

onclick

双击事件

ondblclick

鼠标移动上来

onmouseover

鼠标移动出去

onmouseout

鼠标移动

onmousemove

鼠标按下

onmousedown

鼠标抬起

onmouseup

键盘事件

键按下

onkeydown 任何键

onkeypress 功能键除外

键抬起

onkeyup

页面事件

onload

滚动条事件

onscroll

bom 对象

brower object model

window

window.alert('内容');//提示框

bool window.confirm('内容');//确认框

案例：

文章删除时，提醒确认后删除

function del(){

return window.confirm('看清楚，你删不删？')

}

定时器

t=window.setInterval(函数名,时长)//每过多长时间，执行一次

t=window.setTimeout(函数名或引号加上括号,时长)//过多长时间，执行一次

关闭定时器：

window.clearInterval(t)

window.clearTimeout(t)

练习：实现页面实时显示年月日，时分秒。

1、新建 date.html 页面

2、在页面上加一个 div id 为 div1

3、定义一个 js 函数，获取当前的时间，

把这个时间写入到 div1 中

4、启动一个定时器，每过 1 秒，执行一次上面定义的函数

navigator

作用：获取浏览器的相关信息，版本 类型 .....

属性：

userAgent

例子：输出浏览器的名称。

ie-----ie

firefox----火狐

chrome----google

```
var str=navigator.userAgent.toLowerCase();
if(str.indexOf('msie')){//包含 msie
    document.write('ie');
}else if(str.indexOf('firefox')){
    document.write('火狐');
}else if(str.indexOf('chrome')){
    document.write('google');
}
```

location

作用：用来操作 url 地址

属性：

href//整个 url 地址的内容

host//域名

pathname//路径

search//传值

hash//锚点名称

方法：

replace('toURL')//跳转

案例：等待几秒后跳转？？？

```
<?php
//跳转
function jump($message,$toUrl,$seconds=3){
$jumpContent=<<<EOF
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
<style type="text/css">
*{padding:0;margin:0;}
.content{margin:100px auto;width:380px;border:1px solid #ddd;}
```

```
.content
h1{padding-left:10px;line-height:28px;background:purple;color:white;font-size:14px;}
.content p{padding:10px;font-size:14px;}
</style>
<script type="text/javascript">
var num={$seconds};
function changeT(){//t 中的值得变化，3-0，遇到 0，跳转到目标地址
    num--;
    if(num<=0){//跳转
        location.replace("{$toUrl}");

    }else{
        document.getElementById("t").innerHTML=num;
    }
}

</script>
</head>
<body>
<div class="content">
    <h1>跳转中...</h1>
    <p>{$message}，请等待<b id='t'>{$seconds}</b>秒后，自动跳转，如果跳转失败，请<a
```

```
href='{$toUrl}'>点击</a></p>
</div>
</body>
</html>
<script type="text/javascript">
window.setInterval(changeT,1000);
</script>
EOF;
echo $jumpContent;
}
```

screen

作用：获取显示器的宽高信息

screen.height

screen.width

screen.availHeight

screen.availWidth

dom:文档标签相关的一类对象。

每个标签在页面中都会对应一个对象，

js 可以通过这个对象来对标签进行操作，这个对象叫节点对象。

找到页面标签：

document.getElementById("id 属性值");

document.getElementsByClassName("class 属性值");

document.getElementsByTagName("标签名称")

document.getElementsByName("name 属性值找")

document.forms

document.images

document.links

document.title

案例：标题滚动

标题内容："PSD1501 全体师生去人民大会堂开会。给赶出来了"

```
var start=1;
var title=document.title;
//截取字的函数
function sub(){
    var str=title.substr(start);
    document.title=str;
    start++;
    if(start==0.5*title.length){
        start=0;
```

```
    }
}
window.setInterval(sub,300);
```

js 通过这个节点对象，可以对相应的标签做任何操作

内容

Element.innerHTML

Element.innerText　火狐不支持

样式

Element.style.样式名称=值;

说明：遇到 - 用小驼峰命名法

Element.className=类别样式名称;

html 标签的属性

方法：

Element.setAttribute(属性名,值)

Element.getAttribute(属性名)

事件

宽高及位置

object HTMLParagraphElement

object HTMLHeadingElement

DAY03　节点对象　表单操作

找到页面元素：

document.getElementById('id 值')

document.getElementsByTagName('标签名称');

document.getElementsByName('name 属性值')

document.title

document.links

document.images

document.forms

document.body

标签进行操作：

样式

Element.style.样式属性名="样式属性值";

Element.className="类别样式名称";

属性

Element.getAttribute("标签属性名称")

Element.setAttribute("标签属性名称","标签属性值")

内容

Element.innerHTML //带有 html 标签

Element.innerText

事件

Element.标签事件属性名称=函数名|匿名函数;

光标事件

onfocus

onchange

onblur

鼠标事件

onclick

ondblclick

onmouseup

onmousedown

onmousemove

onmouseout

onmouseover

键盘事件

onkeydown

onkeypress

onkeyup

页面事件

onload

滚动条事件

onscroll

案例：

给页面追加 onscroll 事件

window.onscroll=fun2;

宽高及位置

Element.offsetWidth　width+padding+border

Element.offsetHeight

Element.offsetTop　离它最近的那个设置了 position 为 relative|absolute 祖先元素的距离

Element.offsetLeft

滚动条的位置：

Element.scrollTop

Element.scrollLeft

document.documentElement.scrollTop+document.body.scrollTop;

案例：

对联广告

标签的动态添加　删除　找子节点　父节点(了解)

添加：

document.createElement("标签名称");

Element.appendChild(节点对象)

Element.insertBefore(新节点,在谁的前面)

删除：

Element.removeChild(子节点对象)

找子节点：

Element.childNodes

父节点

Element.parentNode

表单操作

<form action="" onsubmit="return 函数()">

</form>

表单元素：

单行文本框

密码框

文本域

内容操作

Element.value

下拉列表框

选中项的值,option 的 value 属性值

selectElement.value

selectElement.options 获取 select 下的 option

找到所有 option

optionElement.selected 确定谁被选中了

optionElement.value option 的 value 属性值

optionElement.text option 的内容

案例

两级联动：

liandong.html

复选框

选中未选中

checkboxElement.checked

案例：

全选反选

selcetAll.html

提交按钮

buttonElement.disabled

置灰，不可用

可用

普通按钮

重置按钮

控制表单提交：

```
<form onsubmit="return  函数();">
<input type="submit" value="提交"/>
</form>

<form>
<input type="button" value="提交"/>
</form>
```

作业：

表单验证

反选全选

三级联动

第七部分 jQuery

DAY01 jQuery 基础

js 总结

基础知识

基本函数库

字符串

str.length

str.indexOf('char')

str.lastIndexOf('char')

str.substr(start[,length])

str.replace("查找谁","替换成什么");

特点：每次替换一个，第一个参数使用正则表达式的匹配模式 g，来实现替换多个。

str.split("分隔符")

str.toUpperCase()

str.toLowerCase()

array str.match(/正则表达式/g)

时间日期

var d=new Date();

年

d.getFullYear()

月

d.getMonth() 0-11

日

d.getDate()

时

d.getHours()

分

d.getMinutes()

秒

d.getSeconds()

d.getDay()

d.toLocaleString()

数值

Math.round(number)

Math.ceil(number)

Math.floor(number)

Math.random()

案例：

验证码刷新

数组

var arr=[];

arr.length

arr.push(var1,....)

arr.unshift(var1,....)

arr.pop()

arr.shift()

arr.splice(start,length[,var1,var2,...])

arr.indexOf('value')

arr.lastIndexOf('value')

arr.sort(function(a,b){

return a-b;

})

array arr.concat(array,....)

string arr.join('分隔符')

string arr.toString()//分隔符默认，

正则表达式

/正则表达式/.test(字符串)

bom

window

window.alert()

window.confirm()

var t=window.setInterval(函数|匿名函数,时长)

var t=window.setTimeout(函数|匿名函数,时长)

window.clearInterval(t)

window.clearTimeout(t)

location

属性：

href

host

pathname

search

hash

方法：

replace()

navigator

属性：

userAgent

screen

属性：

width

height

availWidth

availHeight

dom

如何找到页面标签

var ob=document.getElementById()

document.getElementsByTagName()

document.getElementsByName()

document.body

document.title

document.images

document.links

document.forms

如何操作

内容

Element.innerHTML

样式

Element.style.样式属性名[=样式属性值]

Element.className="类别样式名";

属性

Element.getAttribute()

Element.setAttribute()

标签有什么属性，对应的节点对象就有什么属性

事件

Element.事件名称=函数|匿名函数;

宽高及位置

Element.offsetWidth

Element.offsetHeight

Element.offsetLeft

Element.offsetTop

Element.scrollTop

Element.scrollLeft

节点的动态操作

docuemtn.createElement(tagName)

Element.appendChild(node)

Element.insertBefore(newNode,whoNode)

Element.removeChild(node)

Element.childNodes

Element.parentNode

表单操作

formElement.submit()

formElement.action

formElement.method

inputElement.value

selectElement.value

selectElement.options

optionElement.text

optionElement.value

optionElement.selected

checkboxElement.checked

jquery:一个 js 框架

下载

程序中加载　jquery.js

<script type="text/javascript" src="js 文件路径"></script>

找到页码标签,选择器

object $("选择器内容")

id 选择器

$("#id 值")

类别选择器

$(".类别名")

标签选择器

$("标签名称")

属性选择器：根据标签的属性去找

$("[属性名='值']")

$("[属性名^='值']")

$("[属性名$='值']")

$("[属性名*='值']")

后代选择器

$("选择器 1 选择器 2 .....")

子代选择器

$("选择器 1>选择器 2>.....")

位置选择器

$(":eq(位置值)")//某位置的元素

$(":first")//第一个

$(":last")//最后一个

$(":gt(位置值)")//某位置之后的元素

$(":lt(位置值)")//某位置之前的元素

$(":even")//处在偶数位的元素

$(":odd")//处在奇数位的元素

例子：

找 li 中第一个

$("li:eq(0)")

并集选择器

$("选择器 1,选择器 2,.....")

交集选择器

$("选择器 1 选择器 2.....")

过滤选择器:找表单元素

$(":input") 所有的 input 框

$(":hidden")

$(":text")

$(":checkbox")

$(":redio")

$(":submit")

$(":button")

```
$(":reset")

$(":password")


$(":selected")//找到选中的 option

$(":checked")//找到复选框中选中的那些

$(":disabled")//不可用的元素



//遍历选择器对象

选择器对象.each(function(index){

    alert(index+"-----"+this);

    //把 this 转化成选择器对象

    $(this)

})
```

练习：

有一个学生表格，7 行，第一行头有一个背景颜色，

其他行，隔行背景颜色不一样。

json 对象：轻量级数据存储对象，关联下标数组

格式：

{'下标名':'值','下标名':'值',.....}

[{'下标名':'值','下标名':'值',.....},{'下标名':'值','下标名':'值',.....},...]

例子：一个学生信息

{'id':1,'name':'xiaoli'}

三个学生信息

[{'id':1,'name':'xiaoli'},{'id':2,'name':'xiaozhang'},{'id':3,'name':'xiaowang'}]

读取：

json 对象.下标名

json 对象[0].下标名

对页面标签进行 6 方面操作

样式

Object.css("样式属性名称"[,样式属性值])

Object.css({'样式属性名':'值',.....});

Object.addClass("类别样式名 类别样式名 ....")//类别样式

Object.removeClass("类别样式名称 类别样式名 ....")

Object.toggleClass("类别样式名称 类别样式名 ....");//类别样式交替出现

Object.show(speed)

Object.hide(speed)

speed: 值 默认单位为毫秒

slow

fast

normal

内容

Object.text([content])

object.html([content])

object.val([content])//用来操作表单元素的 value 属性值

属性

object.attr("属性名"[,值])

验证码刷新：

$("#verify").attr('src',"值");

宽高及位置

object.width()

object.height()

object.outerWidth()

object.outerHeight()

获取窗口的宽和高

$(window).width();

$(window).height();

页面的宽和高

$(document).width();

$(document).height();

获取元素所在的位置

object.offset().left

object.offset().top

获取滚动条的位置

$(document).scrollLeft()

$(document).scrollTop()

事件

节点的动态操作

动画

鼠标位置的获取

ajax

xml

练习：

用户名验证。

作业：

使用 jquery 中的样式和内容操作，实现表单验证。

使用 jquery 中位置及样式操作，实现对联广告。

DAY02　事件操作

作业：

使用 jquery 中位置及样式操作，实现对联广告。

1、获取页面滚动条离开上端的距离

2、把这个值赋给两个 div 的样式属性 top

给 window 的滚动条

事件操作

object.bind(事件名称,函数名|匿名函数)//绑定事件

object.unbind(事件名称);

一个事件名对应一个函数

事件名称：

change(函数名|匿名函数)

focus(函数名|匿名函数)

blur(函数名|匿名函数)

click(函数名|匿名函数)

dblclick(函数名|匿名函数)

mouseover(函数名|匿名函数)

mouseenter(函数名|匿名函数)//鼠标移动进来

mouseout(函数名|匿名函数)

mouseleave(函数名|匿名函数)//鼠标离开

mousemove(函数名|匿名函数)

mouseup(函数名|匿名函数)

mousedown(函数名|匿名函数)

keydown(函数名|匿名函数)

keypress(函数名|匿名函数)

keyup(函数名|匿名函数)

$(document).ready()//页面加载完成后事件

说明：document 可以省略

scroll(函数名|匿名函数)

submit(函数名|匿名函数)

作用：加载其他 html 页面或 txt 文本

load(文件地址)

练习：

有六个 li，给 6 个 li 动态追加 click 事件，

当事件发生时，提示它是兄弟第几个。

```
<ul>
    <li>li-1</li>
    <li>li-2</li>
    <li>li-3</li>
    <li>li-4</li>
    <li>li-5</li>
    <li>li-6</li>
</ul>
```

动画

object.show(speed)

object.hide(speed)

object.animate({样式属性名:值,....},speed[,回调函数])

object.slideDown(speed)//滑动出来

object.slideUp(speed)//滑动，消失

案例：

slide.html

节点动态操作

追加

object.append(" nihao ")//追加子元素

object.after(" nihao ")//追加弟弟元素

object.before(" nihao ")//追加的哥哥元素

删除

object.empty()//删除所有的子元素

object.remove()//删除自己

鼠标位置

object.mousemove(e){

e.pageX

e.pageY

}

选项卡

实现原理：

有几个选项，就对应有几个内容块 ，当鼠标移动到当前的选项上，对应的块显示出来。

实现步骤：

1、新建一个 HTML 文件

2、加载 jquery.js

3、页面效果

选项一....

内容

.....

4、给 li 追加事件 mouseenter,执行一个 js 函数

5、封装 js 函数

二级联动

1、省怎么放在 json 中

{'1':'河北','2':'河南','3':'山东'}

2、市如何放在 json 中

{

'1':{'11':'承德','12':'石家庄'},

'2':{'21':'郑州','22':'洛阳','23':'开封'},

'3':{'31':'济南'}

}

程序算法：

1、新建 html 文件

2、加载 jquery.js

3、两个下拉列表框

4、为第一个下拉列表，追加 option

5、给第一个下拉框加上事件 onchange

6、封装一个函数，根据省 id，找到所有市，对应给第二个下拉框，追加 option

遮罩层

图片轮换

图片无缝滚动

ajax

xml

DAY03　　AJAX 和 XML

图片轮换

1、新建 html lunhuan.html

2、加载 jquery.js

3、实现页面效果

```
<div class="lunhuan">
    <img src=""/>
    <div>1</div>
    <div>2</div>
    <div>3</div>
    <div>4</div>
```

```
</div>
```

4、加上事件，能够实现鼠标移动到小 div 上时，图片改变。

5、定时器，实现每过一段时间，图片改变。

图片无缝滚动

1、新建 html lunhuan.html

2、加载 jquery.js

3、实现页面效果

```
<div class="content">
    <div id="dong">
        <img src="images/t1.jpg"/>
        <img src="images/t1.jpg"/>
        <img src="images/t1.jpg"/>
        <img src="images/t1.jpg"/>
        <img src="images/t1.jpg"/>
    </div>
</div>
```

4、加上 jquery 代码，实现无限滚动

使用定时器，每过一段时间，让 dong 向左走一点。

ajax：不刷新页面（不改变 url 地址），就能请求服务器。

语法格式：

```
$.ajax({
    url:'',//url 地址，请求谁
    data:'名=值&名=值.....',//传递的数据
    type:'get|post',//数据传输方式
    dataType:'text|htm|script|xml|json',
    success:function(re){
        //对回传值 re 进行处理
    },
    cache:false,
    timeout:3000
})
```

案例：

ajax.html,这个页面中有一个 div 和一个按钮，

但按钮被单击时，请求服务器上的程序 a.php

xml:可扩展标记语言

xml 的标记可变，自定义。

作用：存放数据。任何语言都能识别。

标记的格式：

<标记名 属性='值' ...>内容</标记名>

<标记名 属性='值' .../>

例子：学生数据

student.xml

调用 xml 文件中的数据，呈现一个学生的表格。

例子：省市信息 xml

province.xml

作业：创建一个 xml 文件包含省市县信息，实现一个三级联动。

DAY04 文本编辑器和图片放大镜

```
$.ajax({
    url:,
    data:{a:'1',page:5},
    type:,
    dataType:,
    success:function(re){
        函数体;
```

},

cache:false,

timeout:2000

})

文本编辑器

ckeditor

ueditor

1、加载文件

<script type="text/javascript" charset="utf-8" src="ueditor.config.js"></script>

<script type="text/javascript" charset="utf-8" src="ueditor.all.min.js"> </script>

<script type="text/javascript" charset="utf-8" src="lang/zh-cn/zh-cn.js"></script>

2、启动编辑器。

UE.getEditor(表单元素的 id 值);

图片放大镜

jquery.zoom.js

1、页面结构

<!--呈现大图-->

<img src="imgProd/triumph_small1.jpg" title="triumph" style="border: 4px solid #666;">

呈现小图

<img src='imgProd/thumbs/triumph_thumb1.jpg'>

<img src='imgProd/thumbs/triumph_thumb2.jpg'>

<img src='imgProd/thumbs/triumph_thumb3.jpg'>

2、加载文件

<script src="../js/jquery-1.6.js" type="text/javascript"></script>

<script src="../js/jquery.jqzoom-core.js" type="text/javascript"></script>

<link rel="stylesheet" href="../css/jquery.jqzoom.css" type="text/css">

3、启动放大镜效果

$(document).ready(function() {

$('.jqzoom').jqzoom({

```
            zoomType: 'standard',
            lens:true,
            preloadImages: false,
            alwaysOn:false
        });

});
```

布置项目：企业平台。

第八部分　　项目一　：企业平台

DAY01

布置项目：企业平台。

1、规划目录

class

common

admin

images

js

css

public

栏目

单页面

文章

login.html

login.php

index.php

images

js

css

content

news

index.php

config.ini.php

2、设计数据库（有多少个模块）

```
-- admin
create table admin(
    id int(6) primary key auto_increment,
    username char(20) unique,
    password char(32) not null,
    lastlogin int(10)
);
```

```
-- categroy
create table categroy(
    id tinyint(3) primary key auto_increment,
    cname char(10) not null unique,
    num tinyint(3) default 0,
    type enum('单页面','文章','产品'),
    title varchar(100),
    keywords varchar(200),
    des varchar(250)
);
-- content
create table content(
    cateid tinyint(3) unique,
    con text
);
-- news
create table news(
    id int(8) primary key auto_increment,
    title varchar(50) not null,
    content text not null,
    pubtime int(10),
    clicknum int(6) default 0,
```

imagename varchar(50),

author char(20),

cateid tinyint(3) not null

);

3、封装公共类

Model.class.php

Page.class.php

Water.class.php

Upload.class.php

Thumb.class.php

Verify.class.php

Admin.class.php

4、网站的后台功能

后台登录，首页面，菜单页面，退出

select * from admin where username='1' or 1='1' and password='dfsafd'

5、栏目管理，网站的导航

```
foreach($arr as $k=>$v){
$varName="p".$index;
$$varName=$v;
```

$pdoS->bindParam($k,$v,PDO::PARAM_STR);

}

6、内容管理

企业文章

关于我们

联系我们

工程案例

1、在左侧菜单处，显示栏目列表

不同的栏目，连接不一样，连接地址和 type 有关

封面-----》空

单页面----》内容的更新表单页面

文章-----》文章的列表 ，列表页上有一个连接，添加

产品-----》

2、单页面

3、文章管理

7、实现前台页面

http://localhost/psd1501/project/admin/news/oper.php?page=2&wc=nan&cateid=5&keywords=%E7%9A%84

http://localhost/psd1501/project/admin/news/oper.php?page=1&cateid=5&keywords=%

E7%9A%84

http://localhost/psd1501/project/content.php?cateid=2

参照点 ：当前的文章 id 5

上一篇 5-1 小于 5，的 id 最大的那一个

select * from news where id<5 order by id desc limit 1

$a=1;

下一篇 5+1 大于 5，id 最小的那一个

select * from news where id>5 order by id asc limit 1

第九部分 MySQL 高级

DAY 01-03 MySQL 高级和优化

1. 自我介绍：
   a) 李捷
   b) 本人世代忠良，无不良嗜好
   c) 小清新，捷哥
   d) 年龄：1985

e) 爱运动

2. 课程体系：

a) MySQL 高级和优化

b) 缓存技术

c) 框架 MVC 设计模式

d) Smarty 模板引擎

e) PHP 框架（Yii 和 ThinkPHP）

3. MySQL 高级内容和优化：

a) MySQL 表复制：

i. 复制结构

1. Create table t1 like cms_user;

ii. 复制数据

1. Insert into t1 select * from cms_user;

b) MySQL 索引：

i. 什么是索引？

1. 相当于是书的目录，一条目录就是一条索引

ii. 为什么要使用索引？

1. 快速查询，快速定位数据

iii. 什么时候使用索引？

1. 只要查询，通常情况下，我们必须要创建索引

iv. 索引的分类？

1. 主键索引（Primary Key）

a) 有唯一性约束

b) 一张数据表只允许有一个主键

2. 唯一索引（Unique）

a) 有唯一性约束

b) 一张数据表可以有多个

3. 常规索引（Key）

a) 快速查询

4. 全文索引（FullText）

a) 对全文数据的索引

5. 外键索引（Foreign Key）

a) 用来关联主键的

v. 索引的操作：

1. 创建索引

a) 主键：

i. 在创建数据表时，字段之后直接加 primary key

ii. 在创建数据表时，在所有字段之后加 primary key(id)

iii. 修改表的方式，Alter table t3 add primary key(id);

b) 常规索引：

i. 在创建数据表时，在所有字段之后加 key [索引名称](字段名称)

1. Create table t2(

2. Username varchar(32),

3. Key t2_username(username)

4.)engine=innodb default charset=utf8;

ii. 创建数据表之后：

1. Alter table t2 add key[/index] t2_username(username);

c) 唯一索引：

i. 在创建数据表时，在所有字段之后加 unqiue [索引名称](字段名称)

1. Create table t2(
2. Username varchar(32),
3. unque t2_username(username)
4.)engine=innodb default charset=utf8;

ii. 创建数据表之后：

1. Alter table t2 add unique t2_username(username);

2. 删除索引

a) 主键：

i. Alter table t2 drop primary key

ii. 注意：删除主键之前，需要先删除 auto_increment

1. Alter table t2 modify id int unsigned not null;

b) 常规索引：

i. Alter table t2 drop key[/index] t2_username;

c) 唯一索引：

i. Alter table t4 drop index t4_username;

3. 查询索引

a) Show indexes from t1;

c) MySQL 的视图操作：

i. 什么是视图？

1. 就是对一个 sql 语句执行结果的窗口或者是快捷方式

ii. 为什么使用视图？

1. 安全

2. 简化代码

iii. 如何操作视图？

1. 创建视图

a) Create view 视图名称 as sql 语句

i. Create view cms_user_v as select * from cms_user where id>2 and id<=4;

2. 查询视图

a) 把视图当做数据表来查询即可

i. Select * from cms_user_v;

3. 删除视图

a) Drop view cms_user_v;

d) MySQL 的内置函数：

i. 字符串函数

1. Concat 字符串拼接

a) Select concat(‘cms.’ ,username) from cms_user;

2. Lcase 字符串转小写

a) Select lcase(username) from cms_user;

3. Ucase 字符串转大写
4. Length 字符串长度
   a) Select length(username) from cms_user;
5. Trim/ltrim/rtrim 删除字符串两端的空白符
6. Repeat 重复输出字符串
7. Replace 替换字符串
8. Substr 字符串截取
9. Space 重复输出空格

ii. 数学函数

1. Count 统计
2. Max 最大值
3. Min 最小值
4. Avg 平均值
5. Ceiling 进一法取整
6. Floor 舍去法取整
7. Rand 随机

iii. 日期函数

1. Curdate() 当前日期
2. Curtime() 当前时间
3. Now() 当前日期+当前时间
4. Unix_timestamp(date) 返回对应的日期的时间戳
5. From_unixtime(timestamp) 将时间戳格式化

6. Week(date) 返回日期对应本年的第几周
7. Year(date) 返回日期的年份
8. Datediff(date1,date2) 返回两个日期相差的天数

e) MySQL 预处理：

i. 什么是预处理？

1. 将 SQL 语句提前交给 MySQL 服务器进行语法解析

ii. 为什么要使用预处理？

1. 提高 SQL 语句的执行效率
2. 防止 SQL 注入

iii. 什么时候使用预处理？

1. 尽量使用

iv. 如何操作预处理？

1. MySQL

a) 准备预处理对象：

i. Prepare 预处理对象名称 from ‘预处理 SQL 语句’；

b) 设置变量：

i. Set @i=1;

c) 绑定参数并且执行

i. Execute stmt using @i;

2. PDO

a) Try{

i. ……

ii. $stmt = $pdo->prepare(.....);

iii. $stmt->execute(array(....));

b) }catch(PDOException $e){

i. //处理异常

c) }

f) MySQL 的事务处理？

i. 什么是事务处理？

1. 就是一组有逻辑性的数据的操作，为了保证数据的完整性和一致性

ii. 为什么要使用事务处理？

1. 完整性

2. 一致性

iii. 什么时候使用？

1. 对于金钱相关的操作

2. 对于重要的数据的操作

iv. 事务处理使用的前提条件：

1. 数据表的引擎（类型）必须是 InnoDB

v. 事务处理的操作：

1. 开启事务

a) 所有的 sql 语句并不会真正执行，而是存放在队列当中

b) Begin;

2. 执行 sql 语句

a) Update zhanghu set account=account-1000 where id=1;

b) Update zhanghu set account=account+1000 where id=2;

3. 提交或者回滚

a) Commit/rollback;

vi. 在 PDO 当中如何做事务处理？

g) MySQL 的存储过程？

i. 什么是 MySQL 的存储过程？

1. 类似于 PHP 的函数 function

ii. 为什么要使用？

1. 可以帮助我们简化代码

2. 将一些负责的操作封装起来

iii. 如何操作：

1. \d //

2. Create procedure test()

3. Begin

4. .....

5. End;

6. //

h) MySQL 的触发器？

i. 什么是触发器？

1. 在想一张数据表执行某种操作的时候，会自动触发对其他数据表的相关操作.

ii. 触发器的操作：

1. Create trigger tg1 before insert/update/delete for each row

2. Begin

3. ……

4. End

i) 删除数据常见的问题：

i. 删除数据后，字段的主键自增 ID 的值会继续往下延续，如果想要修改

1. Alter table cms_user auto_increment=1;

ii. 同样，可以使用 truncate 语句将数据表恢复到初始状态

1. Truncate cms_user;

iii. 对 MyISAM 数据表引擎，它在删除数据之后，会在对应的位置产生碎片空间，后续插入的新数据会被存放到该碎片空间当中，所以 ，对于这种表引擎，我们需要经常性的执行 optimize table cms_user;innodb 不需要。

4. **MyISAM 数据表引擎和 InnoDB 数据表引擎的区别**：

a) InnoDB 支持事务处理，而 MyISAM 不支持

b) InnoDB 支持外键的创建，而 MyISAM 不支持

c) MyISAM 数据表效率更高，InnoDB 效率稍低

d) MyISAM 具有表锁，InnoDB 的锁是行级锁

e) MyISAM 会产生碎片空间，会造成磁盘空间的浪费，而 InnoDB 不会

f) InnoDB 数据表创建成功后，会产生一个文件 t1.frm，存储表结构，表数据存储在 ibdata 文件（共享表空间，所有 innodb 数据表的数据全部保存在这个文件当中），MyISAM 数据表创建成功后，会产生三个文件,t2.frm(表结构),t2.MYD（表数据）,t2.MYI（表索引）

5. MySQL 的 help 使用

a) \c 取消当前命令

b) \d 临时修改执行符

c) \G 将结果垂直显示

d) \q 退出客户端

e) source 执行外部的 sql 文件

6. ?的使用

a) ? Create

b) ? Insert

c) ? Cre%

7. 索引的优化：

a) 索引可以帮助我们提高查询效率，但是会降低写入效率

b) 会使索引失效的情况：

i. 条件当中使用 or 语句

1. Select * from cms_user where username=’ zhangsan’ or password=’ 123’;

2. 解决方案：

a) 分开查询

b) Select * from cms_user where username=’ zhangsan’;

c) Select * from cms_user where password=’ 123’;

ii. 条件当中使用 in 语句

1. Select * from cms_user where id in(1,2,3,4)

2. 解决方案：

a) 分开查询

iii. 大范围查询：

1. Select * from cms_user where id>=1 and id<=30;
2. 解决方案：
    a) 尽量避免大范围搜索

iv. 模糊查询%放前边：

1. Select * from cms_user where username like ‘%zhang’；
2. 解决方案：
    a) 使用全文搜索引擎 sphinx 来解决

v. 查询条件的值没有加引号：

1. Select * from cms_user where username=123123;
2. 解决方案：
    a) Select * from cms_user where username=’123123’；

c) 复合（组合）索引：

i. 什么是复合索引：

1. 就是一个索引同时针对多个字段生效

ii. 为什么要使用复合索引？

1. 当我们进行复合查询的时候，从某种意义上来说，会造成全表扫描

iii. 什么时候使用复合索引？

1. 当我们进行复合查询的时候

iv. 复合索引失效的情况：

1. Index(key1,key2,key3)
2. 索引可以使用的情况：
    a) Where key1 and key2 and key3

b) Where key1

c) Where key1 and key2

3. 索引效率的低下：

a) Where key1 and key3

b) Where key2 and key3(失效)

c) Where key3(失效)

8. MySQL 的关联方式？

a) 数据关联关系的形式：

i. 一对一关联

1. 拥有

a) User 拥有 profile

2. 属于

a) Profile 属于 User

ii. 一对多关联

1. 拥有

a) User 拥有 多个 article

2. 属于

a) 多个 article 属于 user

iii. 多对多的关联

b) 关联的方式：

i. 普通关联查询：

1. Select u.id,u.username,u.password,u.rtime,u.rip,p.age,p.sex from cms_user

u,cms_profile p where u.id=p.uid;

ii. 连接关联查询：

1. 左连接

a) left join....on

b) 以左表为主，首先先将 left join 左边的表内容查询出来，根据 on 后的条件，将 left join 右边的表的关联内容查询出来，将这些内容按照 select 后的字段顺序依次输出。

c) Select u.id,u.username,u.password,u.rtime,u.rip,p.age,p.sex from cms_user u left join cms_profile p on u.id=p.uid;

2. 右连接

a) Right join ... On

b) 以右表为主 ,首先先将 right join 右边的表内容查询出来 ,根据 on 后的条件，将 right join 左边的表的关联内容查询出来，将这些内容按照 select 后的字段顺序依次输出。

c) Select u.id,u.username,u.password,u.rtime,u.rip,p.age,p.sex from cms_user u right join cms_profile p on u.id=p.uid;

3. 内连接

a) Inner join ... On

b) 不以任何表为主，直接查询满足 on 的条件的数据

c) Select u.id,u.username,u.password,u.rtime,u.rip,p.age,p.sex from cms_user u inner join cms_profile p on u.id=p.uid;

iii. 嵌套关联查询:

1. 用一条 sql 语句的查询结果作为另外一条 sql 语句的条件
    a) Select * from cms_user where id in (select uid from cms_article);

DAY03 缓存 cache

1. 什么是缓存？
    a) 就是一段在文件或者内存当中临时存放数据的一块区域
2. 为什么要使用缓存？
    a) 降低 MySQL 服务器的压力
    b) 提高页面的响应速度
3. 什么时候使用缓存？
    a) 当用户访问量较大的时候
4. 缓存的分类？
    a) 介质：
        i. 文件
            1. 数据
                a) 序列化缓存
                    i. Serialize 序列化
                    ii. Unserialize 反序列化
                b) JSON
                    i. Json_encode

ii. Json_decode

c) XML

i. Foreach

ii. Simplexml_load_string

d) Array

i. Var_export

ii. Include/require

2. HTML 静态化

a) 将数据组装好后直接缓存成 html 代码，这种缓存方式最为高效

ii. 内存

1. 数据

2. HTML 静态化

5. 伪静态（URL 地址重写）：

a) 什么是伪静态？

i. URL 地址重写

ii. URL rewrite

iii. 地址栏当中的地址是 html 后缀，但是实际访问的还是 php 文件

b) 为什么要使用伪静态？

i. 有利于 SEO

c) 什么时候使用伪静态？

i. 只要我们的网站想被收录，尽量使用伪静态

d) 如何使用伪静态？

i. 开启 apache 的重写模块

1. LoadModule rewrite_module modules/mod_rewrite.so

ii. 允许我们使用外部文件对 apache 配置进行覆盖修改

1. AllowOverride All [有 3 处]

iii. 重启 apache 服务

iv. 在项目根路径下创建.htaccess 文件

1. RewriteEngine On
2. RewriteRule ^([0-9a-z]+)\.html$ $1.php
3. RewriteRule

^([0-9a-z]+)-(\d+)-(\d+)-(\w+)\.html$ $1.php?id=$2&age=$3&username

=$4

作业：

1. 整理课堂笔记
2. 做课堂练习
3. 背函数

a) 字符串 15-22 数组 18-21

b) 预习：

i. 如何将数据缓存到内存当中

ii. 什么是 memcache

iii. Memcache 如何进行安装？

DAY 04 memcache

1. 什么是 Memcache？

   a) Memcache 是一个内存对象缓存系统

   b) 高性能的

   c) 保存键/值对

   d) Memcache 也是 C/S 模式软件

   e) Memcache 是一个项目名称

   f) Memcached 是程序的守护进程名称，memcached 也是软件的名称

2. Memcache 软件的安装：

   a) 客户端

      i. Windows7 和 windows8 安装 telnet

         1. 控制面板-----程序和功能----启用或关闭 windows 功能----telnet 客户端(勾中)

         2. 点击确定

   b) 服务器端

      i. 以管理员身份运行命令提示符（win+x）

      ii. 在命令提示符下：

1. F:/phpdev/memcache/memcached -d install
2. F:/phpdev/memcache/Memcached -d start
3. F:/phpdev/memcache/Memcached -d stop
4. F:/phpdev/memcache/Memcached -d uninstall

3. Memcache 的命令：

a) 登录 memcache 客户端

i. Telnet mecached 服务器 ip 地址 端口

ii. Telnet 127.0.0.1 11211

b) stats 查看服务器当前状态

c) 添加数据(不能覆盖)

i. add 键名 标示位 有效期 长度

ii. 存储值内容

1. add name 1 3600 8
2. zhangsan
3. STORED

d) 删除数据

i. delete 键名

1. delete name

e) 修改数据

i. set 键名 标示位 有效期 长度

1. set name 1 3600 4
2. lisi

f) 查看数据

i. get 键名

1. get name

g) 退出客户端

i. Quit

h) 遍历数据

i. 查找标志位

1. stats items

ii. 根据标志位获取键名

1. stats cachedump 标志位 0

iii. 根据键名获取键名对应的值

1. get 键名

4. 安装 memcache 扩展的步骤：

a) 拷贝 memcache 扩展文件至 php 安装路径/ext 目录当中

i. Php_memcache.dll

b) 修改 php.ini 文件

i. Extension_dir = “f:/phpdev/php5.4/ext”

ii. Extension=php_memcache.dll

c) 重启 apache

5. Memcache 扩展和 memcached 扩展的区别：

a) Memcache 扩展可以使用面向对象和面向过程两种写法

b) Memcached 扩展不提供过程化的写法

c) Memcached 的扩展的方法更为丰富

d) Memcached 是基于 libmemcached 库的扩展，memcached 效率高于 memcache

6. 如何使用 Memcache 扩展？

a) addServer 添加服务器节点（连接服务器）

b) Connect 连接一台服务器

c) Pconnect 持久化连接一台服务器

d) add 方法

i. 最后一个参数：过期时间

1. 时间段（秒数）

a) 3600

2. 时间点

a) Time()+3600

3. 0

a) 永久有效

e) Close 关闭连接

f) Delete 删除一个键

g) Flush 删除所有键

h) Replace 修改

i) Set 添加或者修改

j) Decrement 将键值-1

k) Increment 将键值+1

l) Get 查询

m) getStats 查询服务器信息

7. 遍历数据：

a)

作业

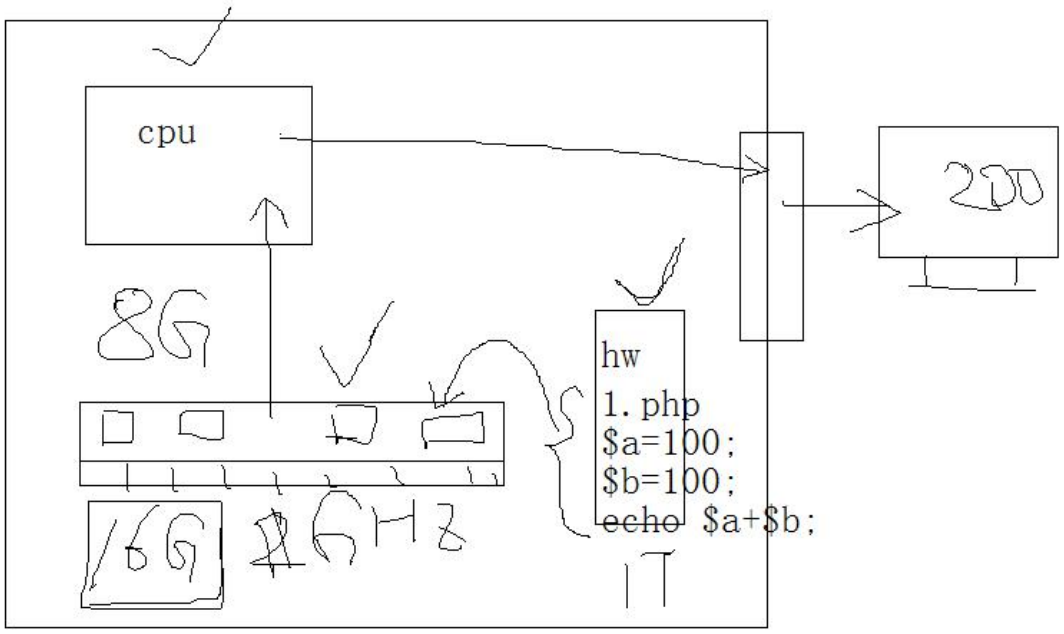
1. 整理课堂笔记
2. 练习课堂代码
3. 背函数

a) 字符串 22-27 数组 23-28

b) 预习：

i. 思考，如何将 session 数据存储到 memcache 当中

ii. Session_set_save_handler();

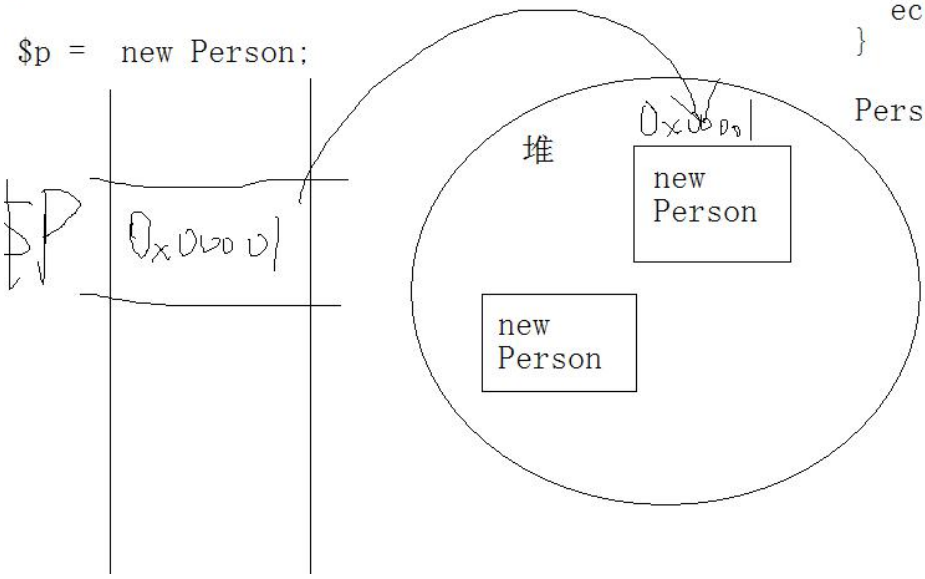


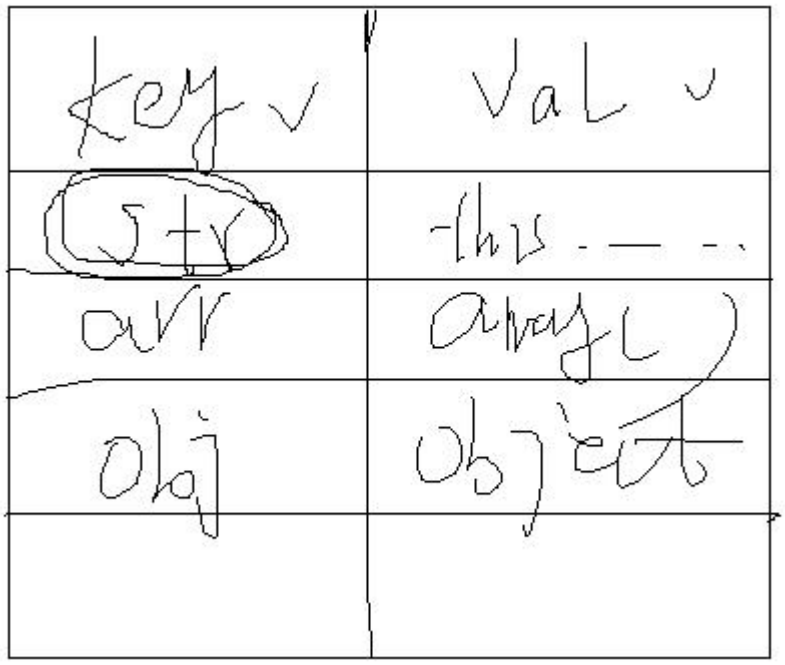
初始化静态段 static define const		效率高
栈 空间小 存储 定长数据 int float	堆 空间大 不定长数据 str arr obj	代码段 function while if(){}

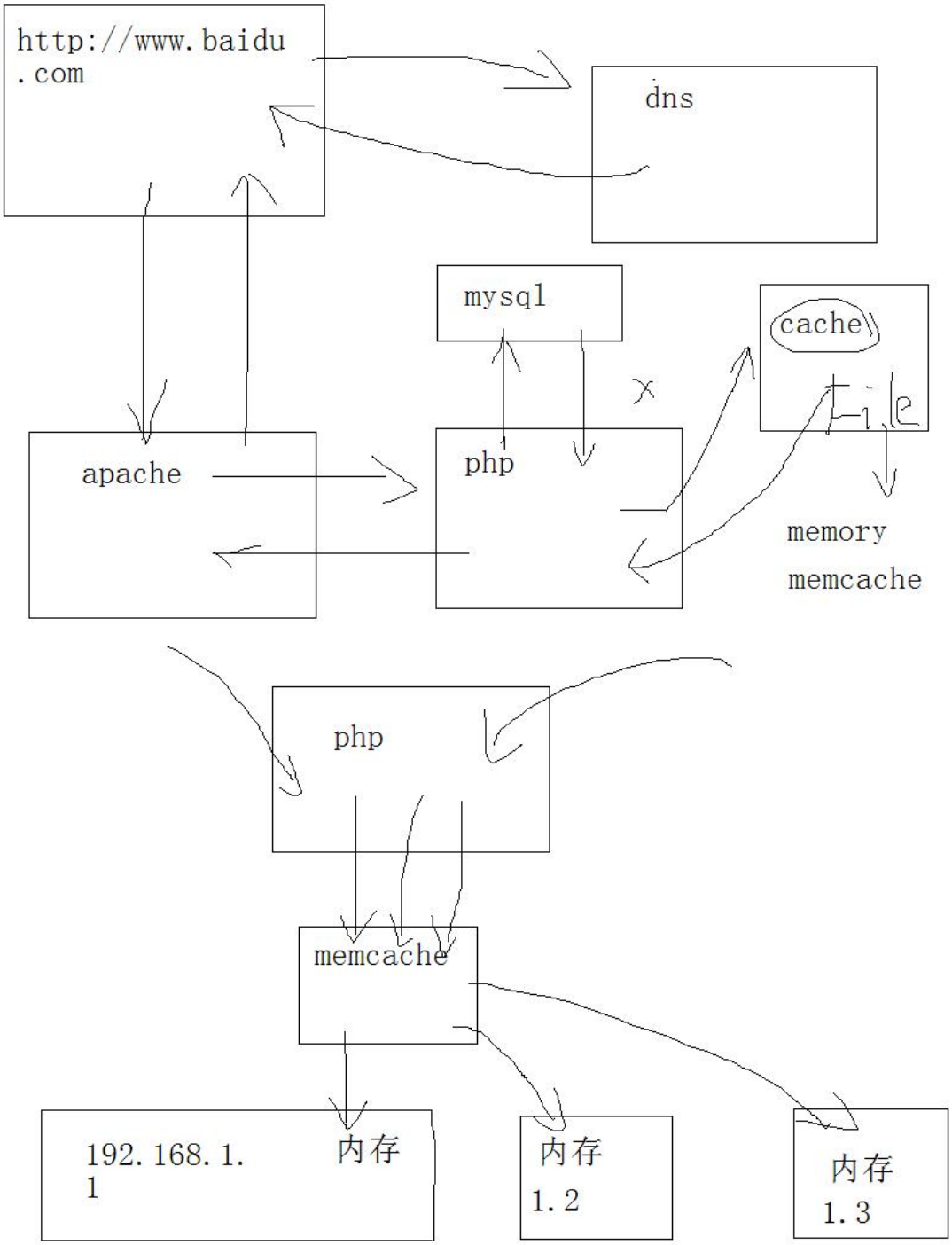
```
class Person{
  public $name="zhangsan";
}

$p =  new Person;
```

```
$p1 = new
Person;
```

```
public static
function test
(){
  echo 123;
}

Person::test()
```






http://www.baidu
.com
dns
mysql
cache
file
apache
php
memory
memcache
php
memcache
192.168.1.
1
内存
内存
1.2
内存
1.3

DAY05 SESSION

1. Cookie 的工作原理：
   a) 服务器会向客户端浏览器发送方一个设置 cookie 的请求，将 cookie 数据写入到 cookie 文件当中，下次再来访问的时候可以通过$_COOKIE 直接读取 cookie 文件当中的内容，cookie 文件是保存在客户端浏览器的。
2. Session 的工作原理：
   a) 当程序遇到 session_start()时
      i. 会产生一个 sessionid 号
      ii. 在服务器上会生成一个以名称为 sess_sessionid 的 session 文件
      iii. 会向客户端浏览器设置一个 cookie 数据，将 sessionid 的值存储到该 cookie 当中
      iv. 此时，我们就可以使用$_SESSION 变量，该变量的内容即为 session 文件读取出来的内容
   b) 当我们第二次访问该 session_start()时，
      i. 浏览器携带 cookie 当中的 sessionid 一起访问服务器
      ii. 程序会判断 sessionid 对应的 session 文件是否存在，如果不存在，则重新产生一个 sessionid，重新生成一个 session 文件，重新设置一个 cookie
      iii. 如果对应的 session 文件存在，则将该 session 文件当中的内容读取出来构建到 $_SESSION 变量当中
3. Cookie 和 session 的区别和联系：
   a) Cookie 保存在客户端浏览器当中，session 保存在服务器当中
   b) Cookie 通常情况下保存一些不是很重要的数据，session 保存一些比较重要的数据
   c) Session 是基于 cookie 来进行存储的，如果 cookie 被客户端禁用，默认情况下，session

无法使用

4. 如果禁用掉客户端浏览器的 cookie，session 是否还可以使用？

   a) 默认情况下，cookie 禁用 session 会失效，因为 sessionid 是保存在客户端 cookie 当中，我们可以通过 get 传参的方式将 sessionid 的值在页面当中进行传输，可以借助 session_name()和 session_id()来拼接参数，也可以使用 SID，推荐使用 SID,如果 cookie 被禁用 SID 就是 session_name 和 session_id 的拼接，否则，SID 为空字符串。

5. Session 的相关配置：

   a) session.save_handler = files　　session 存储的方式

   b) session.save_path = ""　　session 的存储路径

   c) session.use_cookies = 1　　sessionid 是否使用 cookie 传递

   d) session.cookie_secure =　　保存 sessionid 的 cookie 是否使用 ssl

   e) session.use_only_cookies = 0　　sessionid 是否仅能使用 cookie 传递

   f) session.name = PHPSESSID　　session 的名称

   g) session.auto_start = 0　　是否自动开启 session_start()

   h) session.cookie_lifetime = 0　　保存 sessionid 的 cookie 的生命周期

   i) session.cookie_path = /　　保存 sessionid 的 cookie 的有效路径

   j) session.cookie_domain =　　保存 sessionid 的 cookie 的有效域名

   k) session.cookie_httponly =　　是否只能通过 http 协议发送 cookie

   l) session.serialize_handler = php　　session 数据是通过什么方式进行序列化

   m) session.gc_probability = 1

   n) session.gc_divisor = 1000

   o) session.gc_maxlifetime = 1440

i. 当执行 1000 次 session_start 时，会有 1 次执行垃圾回收，回收最后修改时间超过 1440 秒的文件

DAY06 MVC

1. 什么是 MVC？
   a) M
      i. Model
      ii. 模型（数据模型）
      iii. 对数据进行相关操作（数据验证、添加、删除、修改、查询等）
   b) V
      i. View
      ii. 视图（HTML 页面）
      iii. 用来显示页面，用 来与用户进行交互的接口
   c) C
      i. Controller
      ii. 控制器
      iii. 用来处理业务逻辑
2. 为什么要使用 MVC？
   a) 便于团队协作
   b) 分工更加明确
3. 什么时候使用 MVC？

a) 99%的项目都是基于 MVC

b) PHP 基本上所有的框架都是基于 MVC 的

4. 如何设计 MVC？

a) MVC 的访问模式：

i. Index.php

1. 入口文件

ii. index.php?m=user&a=reg

1. M:module

a) 功能模块名称（控制器名称）

2. A:action

a) 功能模块的操作名称（方法名）

5. 什么是模板引擎：

a) 就是一段 PHP 代码

b) 这段代码当中包含着非常强大的正则表达式，帮助我们加载模板，替换模板

6. 为什么使用模板引擎？

a) 就是为了使业务逻辑层（控制器层）和视图层（表现层）相分离

7. 什么时候使用模板引擎？

a) 基于 MVC 的开发

8. PHP 的模板引擎的分类？

a) Smarty

9. 什么是 Smarty：

a) Smarty 是编译型模板引擎

10. Smarty 的优势：
    a) 效率高（相较于其他模板引擎）
    b) 编译型
    c) 可扩展性强（插件技术）
    d) 可以缓存
    e) 使用模板继承
11. Smarty 的缺点：
    a) 降低代码的运行效率
    b) 实时性要求高的项目
12. Smarty 的安装：
    a) 实例化对象
        i. $smarty = new Smarty;
    b) 修改配置
        i. $smarty->setTemplateDir(“./View”);
        ii. $smarty->setCompileDir(“./Runtime/Compile”);
        iii. $smarty->setCacheDir(“./Runtime/Cache”);
        iv. $smarty->addPluginsDir(“./Plugins”);
        v. $smarty->setConfigDir(“./Conf”);
        vi. $smarty->caching = false;
        vii. $smarty->cache_lifetime = 3600;
        viii. $smarty->left_delimiter = “<{”;
        ix. $smarty->right_delimiter = “}>”;

c) 发送变量

i. $smarty->assign();

d) 显示模板

i. $smarty->display();

作业：

1. 整理课堂笔记
2. 练习课堂代码
3. 背函数 字符串+5 数组+5
4. 预习：

a) Smarty 的学习要点（分为哪两个方向学习）

b) Smarty 基本语法

c) Smarty 如何忽略解析

DAY07 Smarty

1. Smarty 的学习方向：

a) 模板的程序员篇

i. Smarty 的安装

ii. Smarty 的发送变量和模板显示

iii. Smarty 的插件扩展

iv. Smarty 的缓存技术

b) 模板的设计者篇

i. 模板的基本语法

ii. 模板的插件的使用

iii. 模板的继承

2. Smarty 模板引擎的基本语法：

a) Smarty 模板引擎的注释：

i. <{*.....*}>

b) 忽略 smarty 解析：

i. 定界符和变量之间加空格

1. <{空格 $content 空格}>

2. auto_literal = false

a) 加空格也会解析

ii. 可以将定界符转为 HTML 实体

1. <{$content}>

iii. 可以使用函数替换定界符

1. <{ldelim}>

2. <{rdelim}>

iv. 可以使用它保留变量：

1. <{$smarty.ldelim}>

2. <{$smarty.rdelim}>

v. 可以块状函数

1. literal 块状函数

a) <{literal}>

i. 被忽略的内容

b) <{/literal}>

3. Smarty 配置文件的使用:

a) 指定配置文件路径

i. $smarty->setConfigDir(“./Conf”);

b) 新建配置文件 siteconfig.conf

i. Sitename=psd1501

ii. Siteurl=www.psd1501.com

c) 在模板当中使用配置项

i. 加载配置文件

1. <{config_load file=”配置文件名称”}>

ii. <{#配置项名称#}>

1. <{#sitename#}>

iii. 保留变量

1. <{$smarty.config.sitename}>

4. Smarty 的保留变量：

a) $smarty.get

b) $smarty.post

c) $smarty.cookies

d) $smarty.session

e) $smarty.env

f) $smarty.server

i. REMOTE_ADDR 客户端 IP

ii. SERVER_ADDR 服务器 IP

iii. HTTP_USER_AGENT 客户端浏览器信息

iv. HTTP_REFERER 上级请求来源

v. SCRIPT_NAME 脚本请求地址

vi. REQUEST_URI 带参数的请求地址

vii. QUERY_STRING 所有的 get 参数字符串

viii. REQUEST_TIME 当前脚本的请求时间

g) $smarty.request

h) $smarty.now 当前时间时间戳

i) $smarty.template 当前脚本请求的模板

j) $smarty.current_dir 模板保存的路径

k) $smarty.ldelim 左边定界符

l) $smarty.rdelim 右边定界符

m) $smarty.config 获取配置项

5. Smarty 的插件机制（核心）：

a) 变量调节器（修改器）

i. 使用方式：

1. PHP 系统内置函数
2. 自定义函数
3. Smarty 封装的函数
    a) <{变量|函数名称}>

ii. smarty 内置

1. Date_format 日期格式化
2. Escape 转码
3. Unescape 反转码
4. Upper 转大写
5. Lower 转小写
6. String_format 字符串格式化
7. Strip 去除空白
8. Strip_tags 删除标签
9. Truncate 字符串截取

iii. 自定义拓展

1. 局部注册
    a) 自定义一个函数
        i. 第一个参数：
            1. 要处理的变量
        ii. 第二个参数.....
    b) 注册成修改器插件
        i. $smarty->registerPlugin(‘modifier’,’模板当中使用的插件名称’,’

自定义函数名称’);

c) 缺点：

i. 只针对当前脚本有效

2. **全局注册**：

a) 添加插件路径

i. $smarty->addPluginsDir(“./Plugins”);

b) 在 Plugins 目录当中新建插件文件

i. Modifier.插件名称.php

c) 编辑该插件文件，声明插件函数

i. Function smarty_modifier_插件名称(){}

b) 常规函数（单标签函数）

i. 使用方式：

1. 不能直接使用 PHP 内置函数

2. 不能使用自定义函数直接使用

3. Smarty 内置函数

a) Config_load 载入配置文件

b) Debug 调试页面

c) Include 加载模板文件

d) Ldelim 左定界符

e) Rdelim 右定界符

f) Counter 计数器

g) Cycle 交换器

h) Html_checkboxes 复选框

i) Html_radios 单选框

j) Html_options 下拉菜单

k) Mailto 发邮件

l) Html_select_date 选择日期

4. 自定义函数的拓展（注册）

a) 局部注册

i. 只能在当前脚本内有效

ii. 自定义一个函数

1. Function myFunc(){}

iii. 注册成为插件

1. $smarty->registerPlugin(‘function’,’editor’,’myFunc’);

b) 全局注册

i. 全局插件完全可以取代局部注册的插件

ii. 指定插件文件路径

1. $smarty->addPluginsDir(“./Plugins”);

iii. 新建插件文件

1. function.editor.php

iv. 声明插件函数

1. Function smarty_function_editor(){}

c) 块状函数（双标签函数）

i. 使用方式：

1. <{strip}>
    a) .....
2. <{/strip}>

ii. Smarty 内置

1. 流程
    a) 顺序流程
    b) 分支流程
        i. If(){}
        ii. If(){}else{}
        iii. If(){}elseif(){}else{}
        iv. Switch(){case}
    c) 循环流程
        i. For(){}
        ii. While(){}
        iii. Do{}while();
        iv. Foreach(){}
    d) 特殊流程
2. 内置函数：
    a) <{if}><{elseif}><{else}><{/if}>
    b) <{for}><{/for}>
    c) <{while}><{/while}>
    d) <{foreach}><{foreachelse}><{/foreach}>

e) <{section}><{sectionelse}><{/section}>

f) Block 继承块

g) Literal 忽略解析

h) Nocache 局部不缓存

i) Strip 压缩页面

iii. 自定义函数

1. 局部注册

a) 自定义一个函数

i. Function myFunc(){}

ii. 第一个参数：所有的属性

iii. 第二个参数：要处理的内容

iv. 第三个参数：smarty 对象

v. 第四个参数：是否重复

b) 注册成为插件

i. $smarty->registerPlugin(‘block’,’模板当中使用的插件名称’,’自定义函数的名称’);

2. 全局注册

a) 指定插件路径

i. $smarty->addPluginsDir(“./Plugins”);

b) 新建插件文件

i. Block.插件名称. php

c) 新建插件函数

i. Function smarty_block_插件名称(){}

6. Smarty 三种插件的应用场景：

a) 变量调节器

i. 对发送来的数据进行调节修改

1. <{$str|strip_tags}>

b) 单标签函数

i. 调用某种功能

1. <{html_checkboxes}>

c) 双标签函数

i. 对模板当中的一大段内容进行修改

1. <{literal}><{/literal}>

7. Smarty 的高级应用：

a) 模板继承

i. 父模板当中可以使用<{block name=”” }><{/block}>将要可以继承后修改的内容包含起来

ii. 在子模板当中继承父模板

1. <{extends file=”父模板” }>

iii. 声明一个与父模板当中<{block}>相同名称的<{block}>，即可将父模板当中对应的<{block}>块的内容覆盖掉

1. 在子模板的 block 块后添加关键字 append，代表的意思是子模板内容会在父模板内容后追加

2. Prepend 在前追加

iv. 可以在父模板当中的<{block}>包含的内容当中添加<{$smarty.block.child}>，代表的意思是，将子模板的内容会放入到该位置

v. 可以在子模板当中的<{block}>包含内容当中添加<{$smarty.block.parent}>，代表的意思是，将父模板的内容会放入该位置

b) 缓存

i. $smarty->setCacheDir(“./Runtime/Cache”);

ii. $smarty->caching = 0|1|2;

iii. $smarty->cahe_lifetime = 3600;

iv. Caching:

1. 0

a) 关闭缓存

2. 1

a) 开启缓存，当缓存策略发生改变时，原有的缓存会立即失效，现有的缓存策略会立即生效

3. 2

a) 开启缓存，当修改缓存策略时，原有的缓存不会立即失效，而是要等到上一个缓存生命周期到期后才会失效，新策略才会生效

v. 可以让模板局部不发生缓存：

1. <{nocache}>

a) 中间内容不会缓存

2. <{/nocache}>

vi. 根据不同的参数，产生不同的缓存文件

1. $smarty->display(‘1.html’ ,$_SERVER[‘REQUEST_URI’]);

vii. 缓存相关的方法：

1. $smarty->isCached(“1.html” ,$_SERVER[‘REQUEST_URI’]);
2. $smarty->clearCache();
3. $smarty->clearAllCache();

8. 请简述 echo，print，var_dump，var_export，print_r，printf，sprintf 函数的区别？

a) Echo

i. 输出，不是函数，是一种语言结构

ii. Echo($a)

iii. Echo 可以不适用括号

iv. Echo 可以同时输出多个变量，变量与变量之间用逗号隔开

v. 无返回值

b) Print

i. 输出一个变量，不是函数，是语言结构

ii. Print 只能输出一个变量，不能同时输出多个

iii. 有返回值，返回值为整型 1

c) Var_dump

i. 是一个函数，无返回值

ii. 可以同时输出多个变量的值

iii. 可以打印变量的相信信息

d) Var_export

i. 是一个函数

ii. 表示对一个变量的字符串输出

iii. 如果第二个参数为 true，输出的内容不再输出，而是返回值

e) Print_r

i. 是一个函数

ii. 返回值为布尔类型，与 var_dump 不同的是，输出的内容不会显示数据类型

f) Printf

i. 函数

ii. 按照指定格式输出字符串

iii. 返回输出字符串的个数

g) Sprintf

i. 函数

ii. 按照指定的格式返回字符串

iii. 返回字符串

作业：

1. 整理课堂笔记
2. 练习课堂代码
3. 背函数 +10
4. 预习：

a) Smarty 的单标签函数如何自定义

b) 尝试完成一个插件

i. <{editor}> 调用出 ueditor

c) 尝试完成日期选择器

i. <{date_select}> 调用出日期选择器的特效

附录 1：Sublime 编辑器

Sublime 编辑器：

一、介绍：

1. 下载官网: http://www.sublimetext.com/ 版本最新：3.0 稳定版：2.0

2. 特征：

1.启动速度非常的快。

2. 编辑区，码农每天工作的地方，当然越大越好！

3.自动保存未保存到硬盘上的文件！

二、功能和菜单：

1.File：

New file: 新建文件 Ctrl + n

Open file: 打开文件 Ctrl + o 。可以自动识别文件类型，并且给语法高亮，右下角可以手动修改！

2. Edit：

常规：

复制： Ctrl + c

剪切： Ctrl + x

黏贴： Ctrl + v

保持缩进黏贴：Ctrl + Shift + v

撤销： Ctrl + z

恢复撤销： Ctrl + y

Line：

左缩进： Ctrl + [

右缩进： Ctrl +]

合并行： Ctrl + j

合并代码显示：Ctrl + Shift + [

关闭合并代码：Ctrl + Shift +]

行上移： Ctrl + Shift + Up 键

行下移： Ctrl + Shift + Down 键

复制行： Ctrl + Shift + d

删除行： Ctrl + Shift + k

Comment：

注释一行（注释选中）: Ctrl + /

注释父标记（注释选中）: Ctrl + Shift + /

Text：

插入行前：Ctrl + Enter

插入行后：Ctrl + Shift + Enter

删除光标之后该行：Ctrl + kk

Selection：

选择该行：Ctrl + L 可重复选择多行

选择该单词：Ctrl + d 可重复选择多个单词

File：

查找： Ctrl + f 替换： Ctrl + h

View：

显示隐藏项目菜单栏：Ctrl + k + b

Goto:

Goto Anthing: Ctrl +p

A :数字

比如： :20 回车锁定第 20 行

B 文件锁定（模糊匹配），如果知道在哪个目录下更好

比如： aa/index.php index.html

C @ 锁定 css 选择器：bgc background-color

D # 匹配页面标签和内容

比如：#body，default_encoding

Tools:

命令模式： Ctrl + Shift + p

A：切换语法模式。（右下角也行）

比如：比如切换到 javascript,输入 set syntax javascript, js 会模糊匹配到 js 的语法

B：右侧 minimap 关闭和启动：比如：mini 就可以了

Preferences:

Settings-Default：

中是标准的json文件 ,是sublim的默认配置项。比如 :font_size:10字体大小 "line_numbers": true 行号

Settings-User: 用户的个性化修改配置。

Color-Scheme: 文本底色，个人喜欢：Sunburst

三、多行游标：

1. 选择该单词：Ctrl + d 可重复选择多个单词

2. 选中多个单词：Alt + F3：

3. Shift + 鼠标右键拖动

四、补全：

alt + . 可以完成补全，但是依旧不够炫酷 Emmet 插件：补全用 Ctrl + e 或者 Tab 键

html:5 或!：HTML5 文档类型

html:xt：XHTML 过渡型文档类型

html:xs：XHTML 严格型文档类型（xhtml）

修改 HTML 初始化文件：Sublime Text 2.0.2\Data\Packages\HTML\html.sublime-snippet

五、符号使用：

1.“#”表 id，“.”表类，“+”表并列：

比如：#header+.bodys+#footer 打印 3 个 div

注意：#和.的简写 div.ok 等同于.ok，div#no 等同于#no，所以直接写.或#时，默认为 Div 标签

是也

ul 和 ol 中输入指的是 li

table、tbody、thead 和 tfoot 指的是 tr

tr 中指的是 td

select 和 optgroup 指的是 option

注意：默认情况下，如.item 和 div.item 起到的作用是一致的 。

2.“>”指子嵌套内容： ul>li>img+p

3. 上一层嵌套内容：

比如：section>div>p>a^p 相当于 select>div>(p>div)+p

4.“()”指同级范围，“[]”表属性：

(ul>li*10>img)+p

h1[title=someting]

html>(head>title+style+script)+body[bgcolor="red"]>(div.content>div.nav+div.sidebar+div.main)+div.footer

5.“*”表标签克隆： div*10 表示打印 10 个 div

6.“{}”表标签内容： h1{这是标题} -- <h1>这是标题</h1>

7.“$”表连续数： .user$*3 打印 3 个 class 分别为 user1 user2 user3 的 div

http://www.w3cplus.com/tools/using-emmet-speed-front-end-web-development.html

教程

六、CSS：

缩写：

margin(m)　　m10-12-12-12

padding(p)　　p12

color(c)

border:1px solid green(bd1-solid-green)

border-bottom(bdb)

border-top(bdt)

font-size(fz)

font-size:18px(fs)

font-size:12em(fz12e)

px→ 默认

p→ %

e→ em

r→ rem

x→ ex

c#fff + bd1-solid-green

附录 2：常用函数总结

字符串函数：

1、去空格或其他字符

trim（ ） 去除字符串两端空格

ltrim（ ） 去除左端空格

rtrim（ ） 去除右端空格 chop（ ） rtrim 的别名

2、字符串生成或转化

str_pad() 将字符串填充为指定长度

str_repeat() 重复使用指定字符串

str_split() 将字符串分割到数组中

strrev() 反转字符串

wordwrap() 按照指定长度将字符串进行折行处理

str_shuffle() 随机打乱字符串

parse_str() 将字符串解析成变量

number_format() 通过千位分组来格式化数字

3、大小写转换

strtolower() 将字符串转化为小写

strtoupper() 将字符串转化为大写

ucfirst() 将字符串首字母转化为大写

ucwords() 将每个单词首字母转化为大写

4、html 标签关联

htmlentities() 将字符转化为 html 实体

htmlspecialchars() 预定义字符转 html 编码

nl2br()	n\转化为
strip_tags	剥去标签
addcslashes()	在指定的字符前添加反斜线转义字符串中字符
stripcslashes()	删除由 addcslashes（ ）添加的反斜线
addslashes()	指定预定义字符前添加反斜线
stripslashes()	删除由 addslashes()添加的反斜线
quotemeta()	在字符串中某些预定义字符前添加反斜线
char()	从指定的 ASCII 码值返回字符
ord()	返回字符串第一个字符的 ASCII 码值

5、字符串比较

strcasecmp()	不分大小写比较两个字符串
strcmp()	区分大小写比较两个字符串
strncmp()	比较字符串的前 n 个字符，区分大小写
strncasecmp()	比较字符串的前 n 个字符，不分大小写
strnatcmp()	自然顺序法比较字符串长度，区分大小写
strnatcasecmp()	自然顺序法比较字符串长度，不分大小写

6、字符串的切割与拼接

chunk_split()	将字符串切割成小块
strtok()	切开字符串
explode()	使用一个字符串为标志分割另一个字符串
implode()	同 join，将数组值用预定字符连接成字符串
substr()	截取字符串

7、字符串查找替换

str_replace() 字符串替换操作，区分大小写

str_ireplace() 字符串替换操作，不分大小写

substr_count() 统计一个字符串在另一个字符串中出现的次数

substr_replace() 替换字符中某串为另一字符串

similar_text() 返回两字符串相同字符的数量

strrchr() 返回一个字符串在另一个字符串中最后一次出现的位置

strstr() 返回一个字符串在另一个字符串中出现的位置

strchr() strstr（ ）的别名

stristr() 返回一个字符串在另一个字符串中出现的位置，不区分大小写

strtr() 转换字符串中的某些字符

strops() 寻找字符串中某个字符最先出现的位置

stripos() 寻找字符串中某个字符最先出现的位置，不分大小写

strrpos() 寻找字符串中某个字符最后出现的位置

strripos() 寻找字符串中某个字符最后出现的位置，不分大小写

strspn() 返回字符串中首次符合 mask 的子字符串长度

strcspan() 返回字符串中不符合 mask 的子字符串的长度

8、字符串统计

str_word_count() 统计字符串中含有的单词数

strlen() 统计字符串的长度

count_chars() 统计字符串中所有字母出现的次数

9、字符串编码：

md5()	32 位加密
sha1()	40 位加密

数学函数

abs()	求绝对值
ceil()	进一步取整
floor()	向下取整
fmod()	浮点数取余
pow()	幂运算
round()	四舍五入
sqrt()	求平方根
max()	求最大值
min()	求最小值
mt_rand()	更好的随机数
rand()	随机数
pi()	获取圆周率值

数组函数

1、 数组创建

array()	生成一个数组
array_combine()	生成一个数组，用一个数组的值做键名，另一个数组的值做值

range() 创建并返回一个包含指定范围元素的数组

compact() 创建一个由参数所带变量组成的数组

array_fill() 用给定的填充数组

2、 数组合并和拆分

array_chunk() 把一个数组分割为新的数组块

array_merge() 把两个或多个数组合并为一个数组

array_slice() 在数组中根据条件取出一段值，并返回

3、 数组比较

array_diff() 返回两个数组的差集数组

array_intersect() 返回两个或多个数组的交集数组

4、 数组的查找替换

array_search() 在数组中查找一个键值

array_splice() 把数组中的一部分删除，用其他值代替

array_sum() 返回数组中所有值的总和

in_array() 在数组中搜索给定的值，区分大小写

array_key_exists() 判断某个数组中是否存在指定的 key

5、 数组指针操作

key() 返回数组内部指针当前指向元素的键名

current() 返回数组中的当前元素

next() 把指向当前元素的指针移动到下一个元素的位置，并返回当前元素的值

prev() 把指向当前元素的指针移动到上一个元素的位置，并返回当前元素的值

end() 将数组内部指针指向最后一个元素，并返回该元素的值

reset() 把数组的内部指针指向第一个元素，并返回该元素的值

list() 用数组中的元素为某一变量赋值

array_shift() 删除数组中的第一个元素，并返回被删除元素的值

array_unshift() 在数组的开头插入一个或多个元素

array_push() 在数组的最后压入一个或多个元素

array_pop() 删除数组中的最后一个元素

6、 数组键值操作：

shuffle() 将数组打乱，保留键名

count() 计算数组中的单元数目或对象中的属性个数

array_flip() 返回一个键值反转后的数组

array_keys() 返回数组所有的键，组成一个数组

array_values() 返回数组所有的值，组成一个数组

array_reverse() 返回一个元素顺序相反的数组

array_count_values() 统计数组中所有的值出现的次数

array_rand() 从数组中随机抽取一个或多个元素，是键名！

each() 每次把指针移动一下

array_unique() 删除重复值，返回剩余数组

7、 数组排序

sort() 按升序对给定数组的值排序，不保留键名

rsort() 对数组逆向排序，不保留键名

asort() 对数组排序，保持索引关系

arsort() 对数组逆向排序，保持索引关系

ksort()　　按键名对数组排序

krsort()　　将数组按照键逆向排序

natsort()　　用自然顺序算法对数组中的元素排序

natcasesort()　自然排序，不区分大小写

附录 3：面试题解析

2015-04-23

1、 mysql数据库优化的一些方法

2、 如何判断SQL语句的运行效率？如何优化一个查询SQL？

Desc select * from cms_user \G

3、 设定网站的用户数量在千万级，但是活跃用户的数量只有1%，如何通过优化数据库提高活跃用户的访问速度？

4、 服务器在运行的过程中，随着用户访问数量的增长，如何通过优化，保证性能？如果数据库已经达到最优化，请设计出继续升级的解决方案

5、 请写出5个常用的SQL操作关键字.5分

6、请列出5个常用的PHP操作MySQL的函数 5分

7、请写出下面MySQL数据类型表达的意义。5分

int(10) char(16) varchar(16) datetime text

2015-04-29

1. 默写 MVC 入口文件 index.php。

```
<?php
    //index.php?m=user&a=reg
    //m:module 模块
    //a:action 操作

    //1.接收参数
    $module = empty($_GET['m'])?'index':$_GET['m'];//USER
    $action = empty($_GET['a'])?'index':$_GET['a'];//doreg

    //2.拼接类名
```

```
$className = ucfirst(strtolower($module))."Controller";//UserController


//3.设置自动加载路径
$include_path = get_include_path();
$include_path .= PATH_SEPARATOR."./Controller";
$include_path .= PATH_SEPARATOR."./Org";
$include_path .= PATH_SEPARATOR."./Model";
set_include_path($include_path);


//4.设置自动加载
function __autoload($className){
    include $className.".class.php";
    //include UserController.class.php
    //include UserModel.class.php
}


//3.实例化对象
$controller = new $className;//new UserController


//4.调用方法
$controller->$action();//$controller->doreg()
```

2. 如何获取客户端 IP 地址和服务器端 IP 地址，如何将客户端 IP 地址存入到数据库当中

a) 客户端：$_SERVER[‘REMOTE_ADDR’]

b) 服务器：$_SERVER[‘SERVER_ADDR’]

c) Ip2long($_SERVER[‘REMOTE_ADDR’])

3. 以下哪个选项是获取客户端浏览器信息：B

a) $_SERVER[‘HTTP_AGENT’]

b) $_SERVER[‘HTTP_USER_AGENT’]

c) $_SERVER[‘USER_AGENT’]

d) $_SERVER[‘HTTP_BROWSE_USER_AGENT’]

4. 如何从数组开头删除一个元素，开头添加一个元素，末尾删除一个元素，末尾添加一个元素

a) Array_shift()

b) Array_unshift()

c) Array_pop()

d) Array_push()

5. 请输出以下程序的输出结果：

a) <?php

i. If(function_exists(‘print’)){

1. Echo “print is a function”；

ii. }else{

1. Echo “print is not a function”；

iii. }

答： print is not a function

6. 请说出以下结果：

a) $arr = array(0=> ‘a’ ,1=>’ b’ ,2=>’ c’);

b) $randValue = array_rand($arr);

c) If(is_string($randValue)){

i. Echo “true”;

d) }else{

i. Echo “false”;

e) }

答：false

2015-04-30

1. 尝试不使用第三个变量，交换两个变量的值

a) $a = ‘abc’;

b) $b = ‘bcd’;

c) $a = array($a,$b);

d) $b = $a[0];

e) $a = $a[1];

2. 编写一个函数，使用尽可能多的方式获取一个文件的后缀名

<?php

//1.

$file = "2.1.jpg";

```
function getExt($file){
    $arr = explode(".",$file);
    return array_pop($arr);
}

//2.
function getExt($file){
    $pos = strrpos(".",$file);
    return substr($file,$pos);
}

//3.
function getExt($file){
    return strrchr($file,".");
}

//4.
function getExt($file){
    return pathinfo($file,PATH_EXTENSION);
}

//5.
```

```
function getExt($file){
    $file = strrev($file);
    return strrev(substr($file,0,strpos($file,".")));
}
```

3. 请简述 printf 和 sprintf 的区别？

4. 请简述 include 和 require 的区别，如何避免重复加载文件？

5. 定义一个函数，尝试将 1234567890，格式化成 1,234,567,890

```
<?php
    $str = "1234567890.1201";

    function format($str){
        return number_format($str);
    }

    echo format($str);
```

6. 尝试使用冒泡排序将一个数组进行从小到大的排序 array(2,4,1,6,3,1,10)

```
<?php
    $arr = array(9,2,8,4,10,6,7);

    for($j=0,$c=count($arr);$j<$c;$j++){
        for($i=0;$i<$c-1-$j;$i++){
            if($arr[$i]>$arr[$i+1]){
```

```
                $tmp = $arr[$i];
                $arr[$i] = $arr[$i+1];
                $arr[$i+1] = $tmp;
            }
        }
    }

var_dump($arr);
```