

河北师大软件学院 @Software College

前端开发与HTML5 程序设计基础

王岩

2.2 PHP语法基础

第一个PHP程序

❖ Demo2-2-1

- ❖ 使用PHP程序在页面输出“Hello PHP！”

```
<?php  
    echo "Hello PHP!";  
?>
```

PHP程序执行原理

- ❖ PHP程序可以嵌入到HTML中
- ❖ PHP程序文件的扩展名为.php
- ❖ PHP程序文件需要放在服务器的网站根目录下，通过浏览器才可以看到PHP程序执行的效果。
- ❖ PHP程序必须放到一对PHP标记中，才能解释为PHP程序

PHP标记

- ❖ 标记方式
 - ❖ 标准标记 `<?php ?>`
 - ❖ 短标记 `<? ?>`

PHP基本语法

PHP语法概述

- ❖ PHP程序结构同C语言非常相似，它是由 C 和 Perl 语言发展而来。
- ❖ PHP是弱类型语言。比C语言更灵活，方便。

PHP基本语法

- ❖ PHP中每一条语句以分号结束
- ❖ PHP中的注释
 - ❖ `//`
 - ❖ `/* */`
- ❖ 标识符命名：使用字母、数字、下划线且不能以数字开头

变量 & 数据类型

定义变量

- ❖ 定义变量
 - ❖ 以“\$”符号开头，标识符区分大小写
 - ❖ 定义变量时不需要指定变量的数据类型
- ❖ Demo2-2-2

命名规范

- ❖ 变量的命名规范：
 - ❖ 使用Camel命名法，首个单词首字母小写，其他单词首字母大写
 - ❖ 不鼓励使用数字
 - ❖ 不鼓励使用下划线
 - ❖ 使用能描述具体作用的名字，即做到见名知意

数据类型

- ❖ 内置数据类型（8种）

- ❖ 整型
- ❖ 浮点型
- ❖ 字符串
- ❖ Bool型
- ❖ 数组
- ❖ 对象
- ❖ 资源
- ❖ NULL

数据类型特征

- ❖ 字符串
 - ❖ 使用单引号或双引号引用
- ❖ Bool:
 - ❖ 取值：true、false

Bool类型转换

- ❖ 其他类型转换为Bool型，以下值被认为是false：
 - ❖ 整型值 0（零）、浮点型值 0.0（零）
 - ❖ 空字符串和字符串“0”（注意空字符串和空格的区别）
 - ❖ 没有成员变量的数组
 - ❖ 没有单元的对象
 - ❖ 特殊类型 NULL

判断数据类型

- ❖ 获得某变量的数据类型: `gettype()`
- ❖ 判断某变量是否为int类型: `is_int()`
 - ❖ `is_bool()`
 - ❖ `is_string()`
 - ❖ `is_array()`
 - ❖ ...
- ❖ Demo2-2-3

常量

- ❖ 常量：定义之后，其值不能改变。
- ❖ `define (标识符, 值);`
- ❖ 常量的命名规范：所有字母大写，出现多个单词时使用下划线进行分割。
- ❖ Demo2-2-4

运算符

运算符

- ❖ 算术操作符：+、-、*、/、%
- ❖ 自增自减：++、--
- ❖ 逻辑运算：&&、||、!
- ❖ 位运算操作符
- ❖ 字符串连接符：.
- ❖ 比较操作符：>、<、>=、<=、==、===、!=、!==
- ❖ 其它运算符：=、?:、@、[]、new等

$3 / 2 = ?$ $3 - \text{true} = ?$ $4 * \text{false} = ?$ $5 \% \text{true} = ?$

$5 \&\& \text{false} = ?$ $(\text{true} == \text{"false"}) = ?$ $(\text{true} === 1) = ?$

$3 + \text{"hello"} = ?$ $3 . \text{"hello"} = ?$ $\text{"hello"} + \text{"world!"} = ?$

流程控制

流程控制

- ❖ 程序的执行顺序：
 - ❖ 自上而下逐条顺序执行
 - ❖ 流程控制语句
 - ❖ 选择
 - ❖ 循环

选择语句

- ❖ if...else语句
- ❖ switch语句
- ❖ Demo2-2-5 随机数游戏
- ❖ Demo2-2-6 点菜游戏

循环语句

- ❖ while语句
- ❖ do...while语句
- ❖ for语句
- ❖ foreach语句（在数组中）

练习

- ❖ 1. 输出<20的奇数（分别使用for、while和do while来完成）
- ❖ 2. 循环输出一个4行3列的表格

break和continue

- ❖ break;
- ❖ break n; // 退出n层循环
- ❖ continue;
- ❖ continue n; // 继续n层循环

结束程序执行

- ❖ `exit()`
 - ❖ 结束程序执行
- ❖ `die()`
 - ❖ `exit()`的别名

包含代码

包含代码

- ❖ 作用：
 - ❖ 代码的可重复性利用
 - ❖ 应用：页面的导航、页脚重复使用；引入函数库，类库...
- ❖ 两种方式：
 - ❖ `require`、`require_once`
 - ❖ `include`、`include_once`
- ❖ Demo2-2-7

require、include

- ❖ require和include的区别
 - ❖ include在引用失败时产生一个警告，之后的代码继续运行
 - ❖ require在引用失败时产生一个致命错误，程序结束运行
- ❖ require_once和include_once
 - ❖ 在同一个页面中，被包含的文件只会被包含一次

谢谢！