

echo曦

博客园

首页

联系

订阅

管理

随笔 - 74 文章 - 0 评论 - 1

昵称：echo曦  
园龄：2年1个月  
粉丝：5  
关注：18  
+加关注

< 2019年1月 >						
日	一	二	三	四	五	六
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

常用链接

我的随笔

我的评论

我的参与

最新评论

我的标签

随笔分类

html(13)

js(12)

laravel(6)

linux(3)

mysql(5)

other(3)

php(27)

yii(2)

随笔档案

2019年1月 (1)

2018年12月 (4)

2018年11月 (3)

2018年10月 (6)

php7新特性的理解和比较

1. null合并运算符(??)

??语法: 如果变量存在且值不为NULL,它就会返回自身的值,否则返回它的第二个操作数.

```
1 //php7以前 if判断
2 if(empty($_GET['param'])) {
3     $param = 1;
4 }else{
5     $param = $_GET['param'];
6 }
7
8 //php7以前 三元运算符
9 $param = empty($_GET['param']) ? 1 : $_GET['param'];
10
11 //PHP7 null合并运算符
12 $param = $_GET['param'] ?? 1;//1
```

2. define() 定义常量数组

```
1 //php7以前
2 define("CONTENT", "hello world");
3 echo CONTENT;//hello world
4
5 //PHP7
6 define('ANIMALS', [
7     'dog',
8     'cat',
9     'bird'
10 ]);
11 echo ANIMALS[2];//bird
12
13 //PHP7 类外也可使用const来定义常量
14 const CONSTANT = 'Hello World';
15 echo CONSTANT;//Hello World
```

3. 组合比较符(<=>)

组合比较符用于比较两个表达式.当\$a小于、等于或大于\$b时它分别返回-1、0或1. 比较的原则是沿用PHP的常规比较规则进行的.

```
1 //整数
2 echo 1 <=> 1; // 0
3 echo 1 <=> 2; // -1
4 echo 2 <=> 1; // 1
5
6 //浮点数
```

2018年9月 (2)
2018年8月 (6)
2018年7月 (17)
2018年6月 (3)
2018年5月 (1)
2018年4月 (3)
2018年3月 (3)
2018年2月 (3)
2018年1月 (1)
2016年12月 (12)
2016年11月 (9)

最新评论

1. Re:mysql基础知识语法汇总整理

很不错的一篇文章，基础很薄弱，正好可以学习

--小刁猿

阅读排行榜

1. Js文件函数中调用另一个Js文件函数的方法(9987)

2. php常用端口号(1632)

3. Yii 1.1 常规框架部署和配置(952)

4. 全国地区sql表(712)

5. PHP读取XML文件数据获取节点值(627)

```
7 echo 1.5 <=> 1.5; // 0
8 echo 1.5 <=> 2.5; // -1
9 echo 2.5 <=> 1.5; // 1
10
11 //字符串
12 echo "a" <=> "a"; // 0
13 echo "a" <=> "b"; // -1
14 echo "b" <=> "a"; // 1
```

4. 变量类型声明

两种模式: 强制(默认)和严格模式. 可以使用下列类型参数: string,int,float,bool

```
1 //... 操作符: 表示这是一个可变参数. php5.6及以上的版本可使用: 函数定义的时候变量前使用.
2 function intSum(int ...$ints){
3     return array_sum($ints);
4 }
5 var_dump(intSum(2, '3.5'));//5
6
7 //严格模式
8 //模式声明: declare(strict_types=1); 默认情况值为0, 值为1代表为严格校验的模式
9 declare(strict_types=1);
10 function add(int $a,int $b){
11     return $a+$b;
12 }
13 var_dump(add(2, '3.5')); //Fatal error: Uncaught TypeError: Argument 2 passed to add() must be of the type integer
```

5. 返回值类型声明

增加返回类型声明的支持.类似于参数类型声明.(用法在函数定义的后面加 :类型名)

```
1 //有效的返回类型
2 declare(strict_types = 1);
3 function getInt(int $value): int {
4     return $value;
5 }
6 print(getInt(6));//6
```

```
1 //无效返回类型
2 declare(strict_types = 1);
3 function getNoInt(int $value): int {
4     return $value+'2.5';
5 }
6 print(getNoInt(6));//Fatal error: Uncaught TypeError: Return value of getNoInt() must be of the type integer
```

6. 匿名类

允许new class {} 创建一个匿名的对象.

```

1 <?php
2 //php7以前 接口实现
3 interface User{
4     public function getDiscount();
5 }
6 class VipUser implements User{
7     //折扣系数
8     private $discount = 0.6;
9     public function getDiscount() {
10         return $this->discount;
11     }
12 }
13 class Goods{
14     private $price = 200;
15     private $objectVipUser;
16     //User接口VipUser类实现
17     public function getUserData($User){
18         $this->objectVipUser = $User;
19         $discount = $this->objectVipUser->getDiscount();
20         echo "商品价格:". $this->price*$discount;
21     }
22 }
23 $display = new Goods();
24 //常规实例化接口实现对象
25 $display ->getUserData(new VipUser);//商品价格:120

```

```

1 <?php
2 //php7 创建一个匿名的对象
3 interface User{
4     public function getDiscount();
5 }
6 class Goods{
7     private $price = 200;
8     private $objectVipUser;
9     public function getUserData($User){
10         $this->objectVipUser = $User;
11         $discount = $this->objectVipUser->getDiscount();
12         echo "商品价格:". $this->price*$discount;
13     }
14 }
15 $display = new Goods();
16 //new匿名对象实现user接口
17 $display ->getUserData(new class implements User{
18     private $discount = 0.6;
19     public function getDiscount() {
20         return $this->discount;
21     }
22 }); //商品价格:120

```

## 7. Closure::call()

Closure::call() 方法被添加为一个简短的方式来临时绑定一个对象作用域到一个闭包并调用它. 与PHP5的bindTo相比,它的性能要快得多.

```

1 <?php
2 //php7以前
3 class A {
4     private $attribute = 'hello world';
5 }
6
7 $getClosure = function(){
8     return $this->attribute;
9 };
10

```

```

11 $getAttribute = $getClosure->bindTo(new A, 'A');//中间层闭包
12 echo $getAttribute();//hello world

```



```

1 <?php
2 //PHP7
3 class A {
4     private $attribute = 'hello world';
5 }
6
7 $getClosure = function(){
8     return $this->attribute;
9 };
10
11 echo $getClosure->call(new A);//hello world

```



## 8. unserialize()

unserialize()函数:过滤的特性,可以防止非法数据进行代码注入,提供了更安全的反序列化数据



```

1 <?php
2 class A{
3     public $name = 'admin_a';
4 }
5 class B{
6     public $name = 'admin_b';
7 }
8
9 $objA = new A();
10 $objB = new B();
11
12 $serializedObjA = serialize($objA);
13 $serializedObjB = serialize($objB);
14
15
16 //默认行为是接收所有类; 第二个参数可以忽略
17 $dataA = unserialize($serializedObjA , ["allowed_classes" => true]);
18 var_dump($dataA);//object(A)#3 (1) { ["name"]=> string(7) "admin_a" }
19
20 //如果allowed_classes设置为false,unserialize会将所有对象转换为__PHP_Incomplete_Class对象
21 $dataA = unserialize($serializedObjA , ["allowed_classes" => false]);
22 var_dump($dataA);//object(__PHP_Incomplete_Class)#4 (2) {
23 ["__PHP_Incomplete_Class_Name"]=> string(1) "A" ["name"]=> string(7) "admin_a" }
24
25 //转换所有对象到 __PHP_Incomplete_Class对象,除了对象"B"
26 $dataB = unserialize($serializedObjB , ["allowed_classes" => ["B"]]);
27 var_dump($dataB);//object(B)#3 (1) { ["name"]=> string(7) "admin_b" }

```



## 9. IntlChar

IntlChar:提供了一些可用于访问Unicode字符信息的实用方法的访问. 注意:必须安装Intl扩展才能使用!

```

1 var_dump(IntlChar::CODEPOINT_MAX);//int(1114111)
2 echo '<br>';
3 var_dump(IntlChar::charName('+'));//string(9) "PLUS SIGN"
4 echo '<br>';
5 var_dump(IntlChar::ispunct('?'));//bool(true)

```

## 10. CSPRNG

CSPRNG 函数提供一种简单的机制来生成密码的随机数.


random\_bytes() -加密生存被保护的伪随机字符串.

random\_int() -加密生存被保护的伪随机整数.


```
1 $bytes = random_bytes(8);
2 echo(bin2hex($bytes)); //随机2073a110a2e3c497
3 echo '<br>';
4 echo(random_int(1, 999)); //随机786
5 echo '<br>';
6 print(random_int(-999, -1)); //随机-357
```

## 11. use 语句

可以使用单个use语句从相同的命名空间导入类,函数和常量,而不是使用多个use语句.



```
1 //PHP7之前
2 use some\namespace\ClassA;
3 use some\namespace\ClassB;
4 use some\namespace\ClassC as C;
5 use function some\namespace\fn_a;
6 use function some\namespace\fn_b;
7 use function some\namespace\fn_c;
8 use const some\namespace\ConstA;
9 use const some\namespace\ConstB;
10 use const some\namespace\ConstC;
11
12 // PHP7之后
13 use some\namespace\{ClassA, ClassB, ClassC as C};
14 use function some\namespace\{fn_a, fn_b, fn_c};
15 use const some\namespace\{ConstA, ConstB, ConstC};
```



## 12. intdiv

新增加intdiv()函数,接收两个参数,返回值为第一个参数除于第二个参数的值并取整.

```
1 echo intdiv(8,4); //2
2 echo intdiv(10,4); //2
3 echo intdiv(5,10); //0
```

## 13. PHP7 错误处理

PHP7 改变了大多数错误的报告方式. 不同于PHP5的传统错误报告机制, 现在大多数错误被作为Error异常抛出.

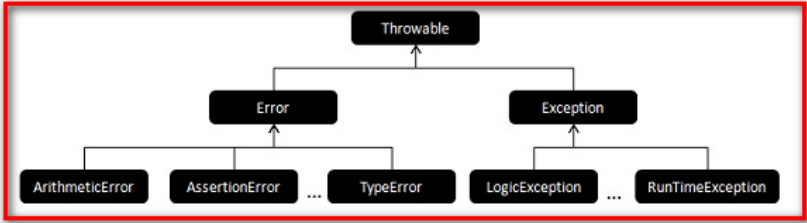
这种Error异常可以像普通异常一样被try / catch块所捕获. 如果没有匹配的try / catch块, 则调用异常处理函数 (由 set\_exception\_handler() 注册) 进行处理.

如果尚未注册异常处理函数, 则按照传统方式处理: 被报告为一个致命错误 (Fatal Error) .

Error类并不是从Exception类扩展出来的, 所以用catch (Exception \$e) { ... } 这样的代码是捕获不到Errors的. 你可以用 catch (Error \$e) { ... } 这样的代码, 或者通过注册异常处理函数 ( set\_exception\_handler() ) 来捕获Error.

Error 异常层次结构

- Error
  - ArithmeticError
  - AssertionError
  - DivisionByZeroError
  - ParseError
  - TypeError
- Exception
  - ...



```
1 <?php
2 //php7以前 自定义异常处理
3 class getException extends Exception{
4     public function errorMsg(){
5         return '错误的信息' . $this->getMessage() . '<br>错误的代码' . $this->getCode();
6     }
7 }
8
9 try {
10     $num =10;
11     if($num > 1) {
12         throw new getException($num,404);
13     }
14 } catch (getException $e) {
15     echo $e->errorMsg();
16 }
```

```
1 <?php
2 //php7 异常处理
3 try {
4     test();
5 } catch(Error $e) {
6     echo $e->getMessage();//Call to undefined function test()
7 }
```

分类: php

好文置顶

关注我

收藏该文

echo曦

关注 - 18

粉丝 - 5

+加关注

0 0

« 上一篇: php匿名函数和闭包的理解  
» 下一篇: mysql基础知识语法汇总整理(二)

**注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。**

**【推荐】超50万VC++源码: 大型组态工控、电力仿真CAD与GIS源码库！**

#### 相关博文：

- [php7新特性](#)
- [PHP7的新特性](#)
- [Linux常用命令大全（非常全!!!）](#)
- [毕业设计 python opencv实现车牌识别 预处理](#)
- [<input type="text">和<textarea>的区别](#)

#### 最新新闻：

- [三星步苹果后尘，财报业绩大幅跳水](#)
  - [地图公司HERE开发打车应用 要用“中立”挑战Uber](#)
  - [联想中国区调整构架：聚焦大客户等三大客户群](#)
  - [特斯拉上海工厂动工 马斯克与蔚来们的“中国战事”](#)
  - [繁荣抖音背后，焦虑的底层内容工厂](#)
- » [更多新闻...](#)