PHP小白极速入门 —— 流程控制

主讲人与课程设计：耕耕

PHP小白极速入门

流程控制(if 和 while)

流程控制

1. If/else
2. While/do-while
3. For/foreach
4. Break/continue
5. switch
6. Require/require_once
7. Include/include_once
8. goto

If/else

if 结构是很多语言包括 PHP 在内最重要的特性之一，它允许**按照条件执行代码片段**

```php
$a = 12;
$b = 12;
if ($a > $b) {
    echo "<h1>a > b</h1>";
} elseif ($a == $b) {
    echo "<h1>a == b</h1>";
} else {
    echo "<h1>a < b</h1>";
}
```

这种情况可以是**else if** (空格分开)

流程控制的替代语法

PHP 提供了一些流程控制的替代语法，包括 if，while，for，foreach 和 switch
替代语法的基本形式是把左花括号（{）换成冒号（:），把最后一个右花括号（}）分别换成
endi，endwhile，endfor，endforeach以及 endswitch

```php
$a = 12;
$b = 12;
if ($a > $b) {
    echo "<h1>a > b</h1>";
} elseif ($a == $b) {
    echo "<h1>a == b</h1>";
} else {
    echo "<h1>a < b</h1>";
}
```

```php
$a = 12;
$b = 12;
if ($a > $b):
    echo "<h1>a > b</h1>";
elseif ($a == $b):
    echo "<h1>a == b</h1>";
else:
    echo "<h1>a < b</h1>";
endif;
```

While/do-while

while 循环是 PHP 中最简单的循环类型

```php
$i = 1;
while ($i<10) {
    echo "<h1>$i</h1>";
    $i++;
}
```

```php
$i = 1;
while ($i<10) :
    echo "<h1>$i</h1>";
    $i++;
endwhile;
```

```php
$i = 1;
do {
    echo "<h1>$i</h1>";
    $i++;
} while ($i<10);
```

PHP小白极速入门

流程控制(for 和 foreach)

For

for 循环是 PHP 中最复杂的循环结构

循环开始前**执行一次**

在**每次循环开始前**求值，
如果值为 TRUE，则继续循环；
如果值为 FALSE，则终止循环。

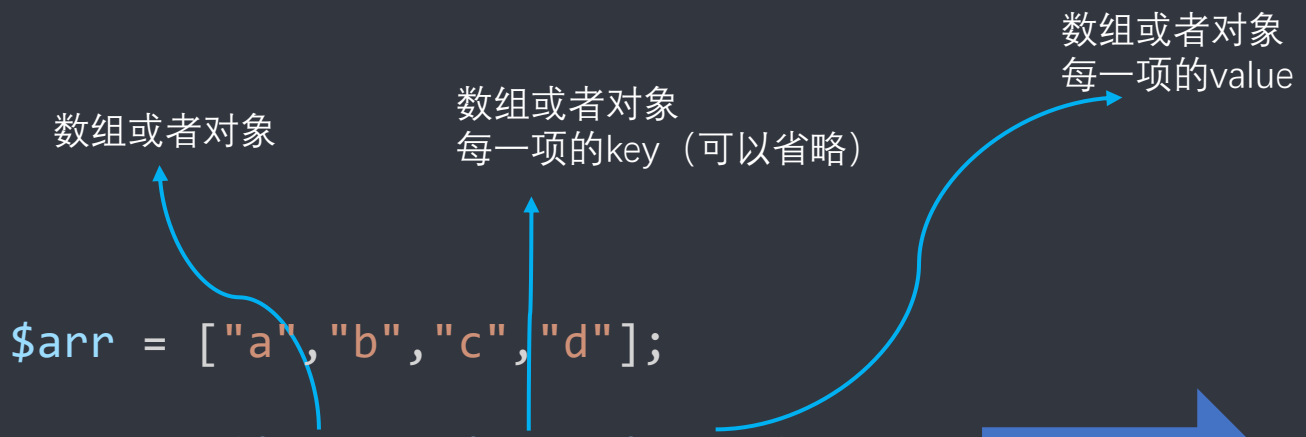
在**每次循环之后**被求值（并执行）

```php
for ($i = 1; $i <= 10; $i++) {
    echo "<h1>$i</h1>";
}
```

```php
for ($i = 1; $i <= 10; $i++) :
    echo "<h1>$i</h1>";
endfor;
```

foreach

foreach 语法结构提供了遍历数组的简单方式，foreach 仅能够应用于**数组**和**对象**




```php
$arr = ["a","b","c","d"];

foreach ($arr as $key=>$value) {
    echo "<h1>$key - $value</h1>";
}
```

```php
$arr = ["a","b","c","d"];

foreach ($arr as $key=>$value) :
    echo "<h1>$key - $value</h1>";
endforeach;
```

PHP小白极速入门

流程控制(break 和 continue)

break

break 结束当前 for，foreach，while，do-while 或者 switch 结构的执行，后面可以跟一个参数，
标示跳出几层循环，默认为1

```php
$list1 = ["a1","b1","c1"];
$list2 = ["a2","b2","c2"];

foreach ($list1 as $key1=>$value1) {
    foreach ($list2 as $key2=>$value2) {
        if ($value1 == "b1" && $value2 == "b2") {
            break 参数;
        }

        echo "<h1>$value1 - $value2</h1>";
    }
}
```

continue

continue 在循环结构用用来跳过本次循环中剩余的代码并在条件求值为真时开始执行下一次循环，用法跟break差不多，也可以跟参数

```php
$list1 = ["a1","b1","c1"];
$list2 = ["a2","b2","c2"];

foreach ($list1 as $key1=>$value1) {
    foreach ($list2 as $key2=>$value2) {
        if ($value1 == "b1" && $value2 == "b2") {
            echo "<h1>-------</h1>";
            continue 参数;
        }
        echo "<h1>$value1 - $value2</h1>";
    }
}
```

PHP小白极速入门

流程控制(switch 和 include)

switch

switch 语句类似于具有同一个表达式的一系列 if 语句，下面左右两边代码是等价的

```php
$i = 1;

if ($i == 0) {
    echo "i equals 0";
} elseif ($i == 1) {
    echo "i equals 1";
} elseif ($i == 2) {
    echo "i equals 2";
}
```

```php
switch ($i) {
    case 0:
        echo "i equals 0";
        break;
    case 1:
        echo "i equals 1";
        break;
    case 2:
        echo "i equals 2";
        break;
}
```

Include/include_once

include 语句包含并运行指定文件，把另外一个php文件引入进来(可以理解为代码复制了一份进来)，include_once标示只引入一次

f1.php

```php
<?php
$a = 100;
```

当前php文件

```php
include 'f1.php';
echo "<h1>value is $a</h1>";
```

```php
$a = 100;
echo "<h1>value is $a</h1>";
```

require/ require _once

require 和 include 几乎完全一样，除了处理失败的方式不同之外

require 在出错时产生 E_COMPILE_ERROR 级别的错误，**将导致脚本中止**
include 只产生警告（E_WARNING），**脚本会继续运行**

goto

goto 操作符可以用来跳转到程序中的**另一位置**
该目标位置可以用目标名称加上冒号来标记，而跳转指令是 goto 之后接上目标位置的标记
PHP 中的 goto 有一定限制，目标位置只能位于**同一个文件和作用域**，也就是说无法跳出一个函数或类方法，也无法跳入到另一个函数，也无法跳入到任何循环或者 switch 结构中
可以跳出循环或者 switch，通常的用法是用 goto 代替多层的 **break，建议少用**

```
goto a;
echo 'Foo';

a:
echo 'Bar';
```