Buổi 22

Nối chuỗi

Dùng dấu chấm "."

VD: $hovaten = "Nguyen" . " " . "Văn" . " ". "A";

Tích lũy giá trị

$tong = $tong + 5; có thể viết tắt lại thành $tong += 5;

+= : cộng dồn

*= : nhân dồn

.= : nối chuỗi dồn

Mảng kết hợp (*Associative array):* dùng chuỗi để làm chỉ số

Sự khác nhau giữa mảng số nguyên so với mảng kết hợp là chỉ số.

- Đối với array thường thì chỉ mục của nó là số nguyên không âm
- Đối với array kết hợp thì chỉ mục của nó là chuỗi.

Truy xuất/cập nhật mảng kết hợp: tương tự như mảng thường

$a = array("Táo" => 3, "Quýt" => 4, "Cam" => 2 , "Lê" => 10, "Ổi" => 5);

- "Táo" là chỉ số, 3 gọi là phần tử (giá trị)
- "Quýt" là chỉ số, 4 gọi là phần tử (giá trị)
- "Cam" là chỉ số, 2 gọi là phần tử (giá trị)
- "Lê" là chỉ số, 10 gọi là phần tử (giá trị)
- "Ổi" là chỉ số, 5 gọi là phần tử (giá trị)

- Truy xuất giá trị của các phần tử trong array
  - $a["Táo"] ;//giá trị là 3
  - $a["Quýt"];//giá trị là 4
- Cập nhật giá trị của các phần tử trong array
  - $a["Táo"] = 4; //giá trị cũ là 3, đã cập nhật lên 4

Mảng nhiều/đa chiều (*Multidimensional array*)

Là mảng chứa một hay nhiều phần tử mà những phần tử này là mảng.

Truy xuất/cập nhật mảng nhiều chiều

Sử dụng nhiều chỉ mục để truy xuất đến giá trị của mảng

```
$diem = array(
        "nga" => array("toan" => 7, "ly" => 4, "hoa" => 8.5),
        "nam" => array("toan" => 4, "ly" => 9, "hoa" => 3.5),
        "nhan" => array("toan" => 7, "ly" => 5, "hoa" => 9.5));
```

- Truy xuất mảng nhiều chiều
  - $diem["nga"]["toan"];// giá trị nhận được là 7
  - $diem["nhan"]["ly"];// giá trị nhận được là 5
- Cập nhật giá trị của mảng nhiều chiều
  - $diem["nga"]["toan"] = 8;//giá trị được cập nhật là 8, giá trị cũ là 7
  - diem["nhan"]["ly"] = 5.5;//giá trị được cập nhật là 5.5, giá trị cũ là 5
- Lệnh điều khiển (rẻ nhánh)

if, switch. Lưu ý switch phải dùng break

```
if (condition) {
  //  code for condition is true
}

if (condition) {
  //  code for condition is true
} else {
  //  code for condition is false
}

switch(expression) {
  case a:
    // code here
    break;
  case b:
    // code here
    break;
  default:
    // code here
}
```

Vòng lặp

for, foreach, while, do while

- for: Vòng lặp dùng để chạy một số lần

```
for (statement 1; statement 2; statement 3) {
  // code here
}
```

statement 1: chạy 1 lần duy nhất trong vòng for, dùng để khởi tạo giá trị

statement 2: kiểm tra điều kiện, nếu thỏa mãn thì mới chạy đoạn code trong vòng for

statement 3: sau khi code trong vòng for chạy xong, thì statement 3 này sẽ thực thi

- foreach: Vòng lặp dùng để truy xuất đến các phần tử của mảng

```
foreach (array as element) {
  // code here
}
```

- while: Lặp đến khi nào điều kiện là false

```
while (condition) {
  // code here
}
```

- do/while: Thực hiện code trong do rồi mới xét điều kiện trong while để biết rằng có nên lặp tiếp hay không

```
do {
  // code here
}
while (condition);
```

Từ khóa break

- Được dùng để thoát khoải vòng lặp

Từ khóa continue

- Được dùng để tiếp tục chạy vòng lặp mà không cần chạy đoạn code phía dưới nó

Hàm (function)

Cú pháp:

```
function function_name(parameter1, parameter2,..., parameter n) {
 // code here
}
```

Gọi hàm (calll function)

function_name(*arg 1, arg 2,..., arg n);*

Hàm đệ quy

- Phải có điểm dừng
- Hàm tính tổng theo cách lặp

```
function tinhtong($n){
 $tong = 0;
 for ($i = 1; $i <= $n; $i++){
  $tong += $i; // mỗi vòng lặp cộng lại với nhau
 }
 return $tong;
}
```

- Viết lại theo đệ quy

```
function tinhtong($n)
{
  if ($n == 1){ return $n; }
  return $n + tinhtong($n-1);
}
echo tinhtong(100);
```

- Giải thuật đệ quy tư duy toán học và nhìn nó khá hay, viết code ngắn gọn nhưng nó chiếm bộ nhớ rất nhiều nên trong trường hợp bộ nhớ ít hoặc để chương trình chiếm ít bộ nhớ người ta khử đệ quy bằng cách chuyển về vòng lặp for/foreach

Biến global (biến toàn cục)

Bài tập

- Bài 1
  - Viết đoạn code tính tổng từ 3 đến 15 dùng vòng lặp for
- Bài 2
  - Viết đoạn code tính tổng các giá trị của các phần tử trong array bao gồm các con số 3, 5, 4, 9, 17, 20
- Bài 3
  - Viết đoạn code tính tổng các giá trị của các phần tử **chẵn** trong array bao gồm các con số 3, 5, 4, 9, 17, 20
- Bài 4
  - Viết hàm isTongChan(a, b);//Hàm này trả về true nếu kết quả tổng 2 số a và b là số chẵn, ngược lại trả về false
- Bài 5
  - Viết hàm isPassed($diem);//Hàm này nhận vào tham số là array và trả về kết quả là true hoặc false. True là đậu, false là rớt
    - Điểm là array có cấu trúc như sau: array("toan" => 7, "ly" => 4, "hoa" => 8.5)
    - Quy tắc tính đậu, rớt như sau: Nếu (toán + lý) * 2 + hóa > 24 điểm thì đậu, ngược lại là rớt
- Bài 6
  - Viết hàm passedList($danh_sach_diem_sv);//Hàm này trả về danh sách tên sinh viên đậu.
    - Biến $danh_sach_diem_sv có cấu trúc như sau:

```
$danh_sach_diem_sv = array(
    "nga" => array("toan" => 7, "ly" => 4, "hoa" => 8.5),
    "nam" => array("toan" => 4, "ly" => 9, "hoa" => 3.5),
    "nhan" => array("toan" => 7, "ly" => 5, "hoa" => 9.5)
);
```

    - Quy tắc tính đậu rớt giống như bài tập 5 ở trên