PHP 시작하기 – II

웹 데이터 베이스

● 한림대학교 금융정보통계학과 이 윤 환, yoonani72@gmail.com ●

배열(Array)

- 배열 (http://ko.wikipedia.org/wiki/배열)
  - 컴퓨터 과학에서 **배열**은 번호(인덱스)와 번호에 대응하는 데이터들로 이루어진 자료 구조를 나타낸다. 일반적으로 배열에는 같은 종류의 데이터들이 순차적으로 저장되어, 값의 번호가 곧 배열의 시작점으로부터 값이 저장되어 있는 상대적인 위치가 된다. 대부분의 프로그래밍 언어에서 사용할 수 있는 가장 기초적인 자료 구조로, 기본적인 용도 외에 다른 복잡한 자료 구조들을 표현하기 위해서 또는 행렬, 벡터 등을 컴퓨터에서 표현하는 용도 등으로도 사용된다.
- 배열의 선언과 할당
  - $array[0] = “Professional”;
  - $array[arr] = “Preprocessor”;
  - $array = array(“Professional”, “Hypertext”, “Preprocessor”);
  - $array = array(“first” => “Professional”);

- 배열의 종류
  - 일반 배열(Index사용 : 숫자에 의한 접근)
    - 배열 할당
      - $fruit = array("사과", "배", "복숭아", "딸기");
    - 배열 사용
      - $fruit[0];
  - 연관 배열(Key 사용 : 문자열로 접근)
    - 배열 할당
      - $phone = array("home" => "123-4567", "office" => "765-4321");
    - 배열 사용
      - $phone["home"];
  - Example : arrex.php

- array **array** ([mixed ...])
  - Returns an array of the parameters. The parameters can be given an index with the => operator.
- bool **print_r** (mixed expression [, bool return])
  - Prints human-readable information about a variable
  - Example : print_r.php

함수

- 함수(function)
  - 함수는 한번 정의될 수 있는 코드 블록이며 프로그램의 다른 부분에서 작동시킬 수 있다.
  - 함수는 반환값, 함수 이름, 전달 인자, 함수 내용으로 구성.
  - 모듈화하여 알기 쉽고 구조화된 응용 프로그램 작성.
    - 자주 반복되어야 하는 코드를 함수로 만들어 한 곳에 저장하고 프로그램의 필요한 곳에서 호출하여 사용한다.

```
function 함수 이름 (매개변수의 리스트) {
        함수 본문
        return 반환값
}  // 함수 정의

호출시 함수 이름(전달인자의 리스트)
```

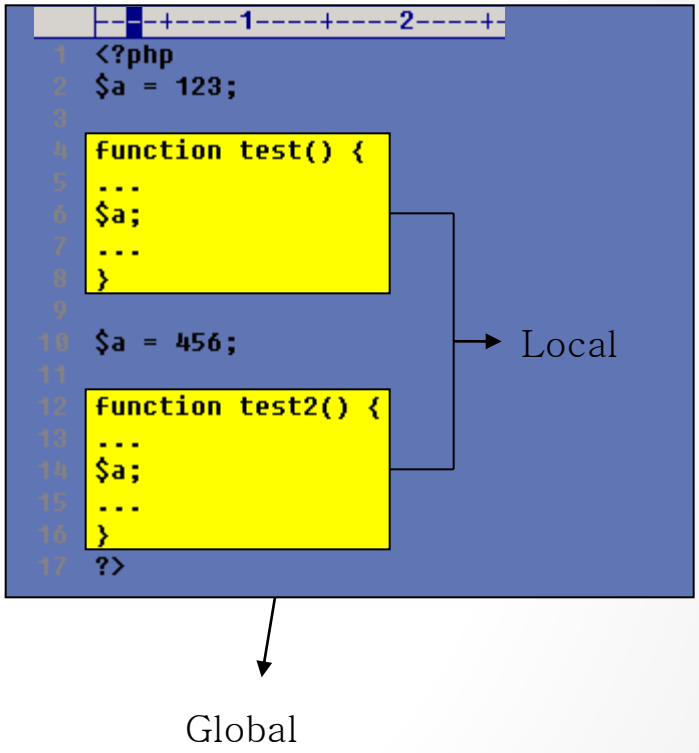
- 함수의 선언과 사용

```
<?php
//fn_ex.php
function mySum($x, $y) {
    $sum = $x + $y;
    return $sum;
}

$a = 3;
$b = 5;

echo "$a과 $b의 합은 " . mySum($a, $b) . "<br>";
echo gettype(mySum($a, $b));
?>
```

- **전달 인자의 여러가지 성질**
  - Passed by reference
    - 일반적으로 전달인자에 의해 값이 전달되는 것은 값에 의한 전달을 하나 참조에 의한 전달이 필요할 때 함수 선언시 전달 인자의 이름 앞에 “&”를 붙힌다.
    - Example : cbr.php
  - 기본 전달 인자
    - 호출시 해당 인자를 호출하지 않아도 기본값처럼 인식되는 전달 인자를 만들수 있다.
    - 선언시 전달인자 = “기본값”의 형태로 선언한다.
    - Example : dparam.php

변수의 범위와 수명

- 변수의 범위는 프로그램의 어떤 부분에서 변수를 액세스 할 수 있는지 결정한다.
  - global 변수명, $GLOBALS["변수명"]
  - Example : globals.php
- 정적 변수 : 한번만 초기화되고 계속 값을 유지한다.
  - static 변수명
  - Example : static.php



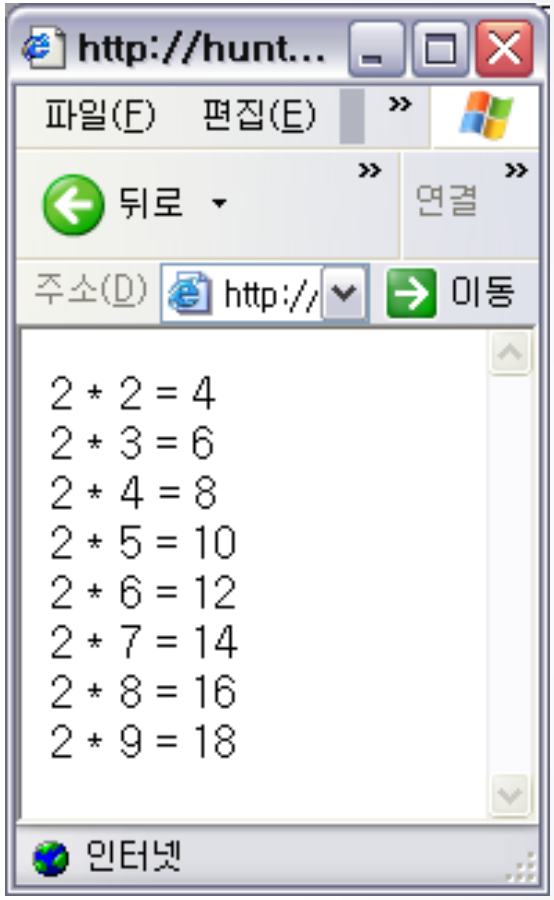
반복문(Loop)

- for : 전통의 반복문
  - for문은 전통의 반복문으로 조건식이 참인 상황에서만 반복 수행할 문을 수행한다.
  - for문은 진입조건 Loop이며 그 의미는 처음에 for문에 진입하기 위해서 조건식이 참이어야 한다는 뜻이다. 즉 진입시(처음부터) 조건식이 거짓이면 Loop를 수행하지 않는다.

```
for(초기식 ; 조건식 ; 증감식) {
        반복수행할 Code Block
}
```

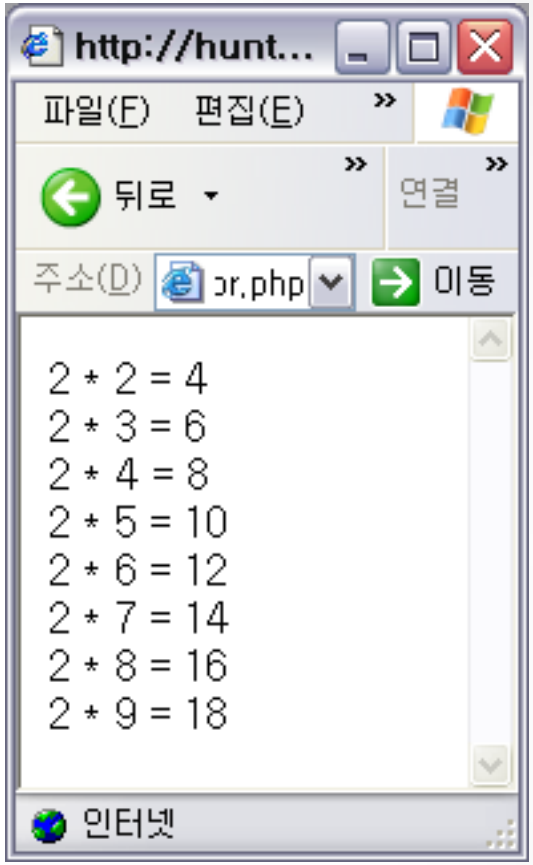
- for를 이용하지 않고 2단 구하기

```
<?php
// 2dan.php
echo "2 * 2 = ", 2*2, "<br />";
echo "2 * 3 = ", 2*3, "<br />";
echo "2 * 4 = ", 2*4, "<br />";
echo "2 * 5 = ", 2*5, "<br />";
echo "2 * 6 = ", 2*6, "<br />";
echo "2 * 7 = ", 2*7, "<br />";
echo "2 * 8 = ", 2*8, "<br />";
echo "2 * 9 = ", 2*9, "<br />";
?>
```



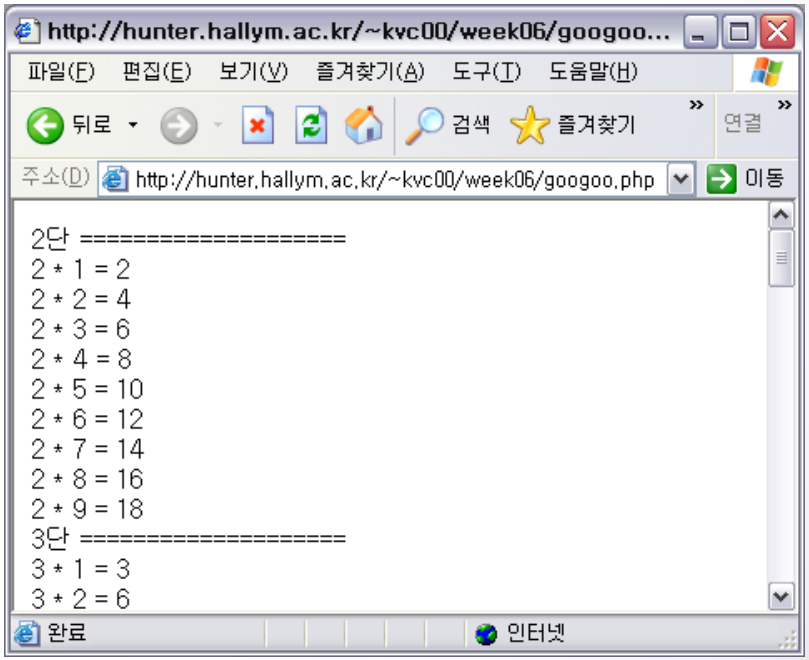
- for를 사용하여 2단 구하기

```
<?php
// 2dan-for.php
for($i = 2; $i < 10; $i++) {
    echo "2 * $i = ", 2*$i, "<br />";
}
?>
```



- 중첩된 for

```
<?php
// googoo.php
 for($i = 2; $i < 10; $i++) {
    echo "${i}단 ======================" . "<br />";
    for($j = 1; $j < 10; $j++) {
      echo "$i * $j = ", $i*$j, "<br />";
    }
}
?>
```



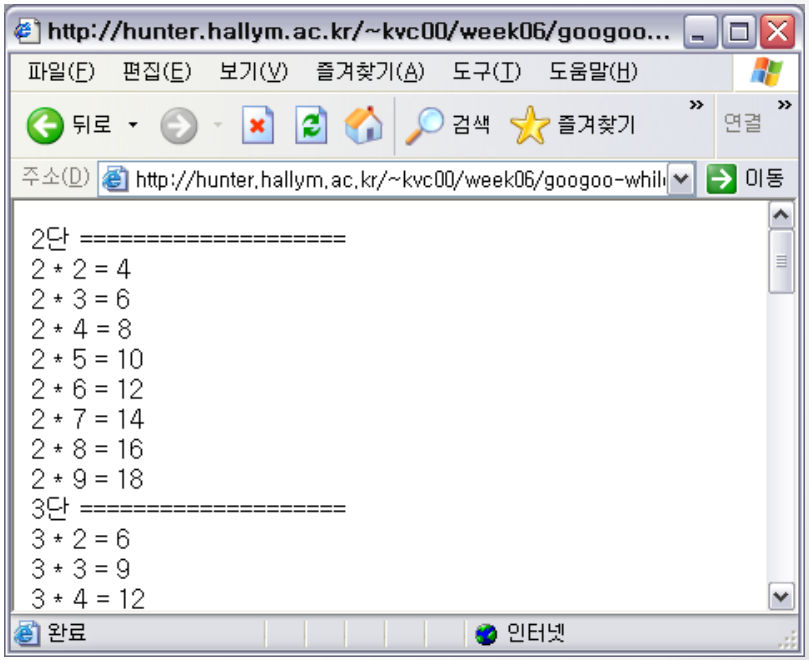
조건식만 존재하는 반복문 – while

- while문은 역시 반복문으로 조건식이 참인 상황에서만 반복 수행할 문을 수행한다.
- 코드 블록 내에서 조건의 대상이 상태가 변하여 false가 되면 반복 블록을 탈출한다.
- while 역시 진입조건 Loop이다

```
while(조건식) {
        반복수행할 Code Block
}
```

- while을 이용한 구구단

```
<?php
//googoo-while.php
$i=2;
while($i < 10) {
    echo "${i}단 ======================" . "<br />";
    $j = 2;
    while($j < 10) {
      echo "$i * $j = ", $i*$j,
           "<br />";
      $j++;
    }
    $i++;
}
?>
```



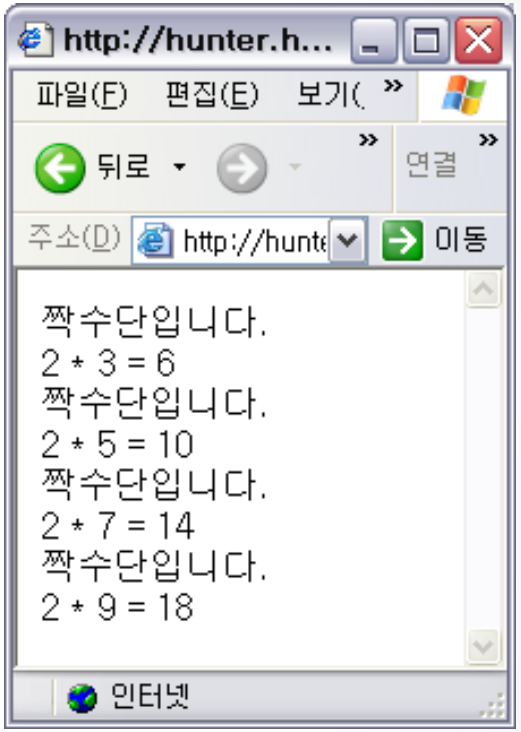
조건문

- **양자 택일문 - if**
  - if문은 전통의 조건 처리문으로 조건이 참인지 거짓인지 두가지만을 판단하여 처리하는 문이다.
  - if문은 확장 사용하여 다중 조건 처리가 가능하다. (else if 사용)

```
if (조건) {
        조건이 참일때 실행할 Code Block
} else {
        조건이 거짓일때 실행할 Code Block
}
```

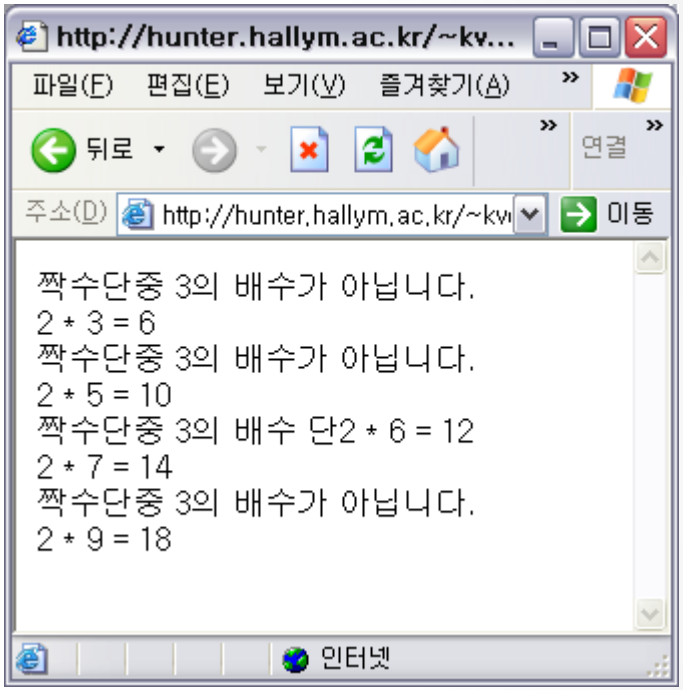
- 앞선 2단의 예에서 홀수단만 구하기

```
<?php
// 2dan-for-if.php
for($i = 2; $i < 10; $i++) {
    if($i % 2 == 1) {
      echo "2 * $i = ", 2*$i, "<br />";
    } else {
      echo "짝수단입니다." . "<br />“;
    }
}
?>
```



- 2단 중에서 홀수단과 짝수단 중에서 3의 배수단 구하기

```
<?php
// 2dan-for-if-2.php
for($i = 2; $i < 10; $i++) {
    if($i % 2 == 1) {
      echo "2 * $i = ", 2*$i, "<br />";
    } else if($i % 3 == 0) {
      echo "짝수단중 3의 배수 단" .
           "2 * $i = ",
           2*$i, "<br />";
    } else {
      echo "짝수단중 3의 배수가
            아닙니다." . "<br />";
    }
}
?>
```

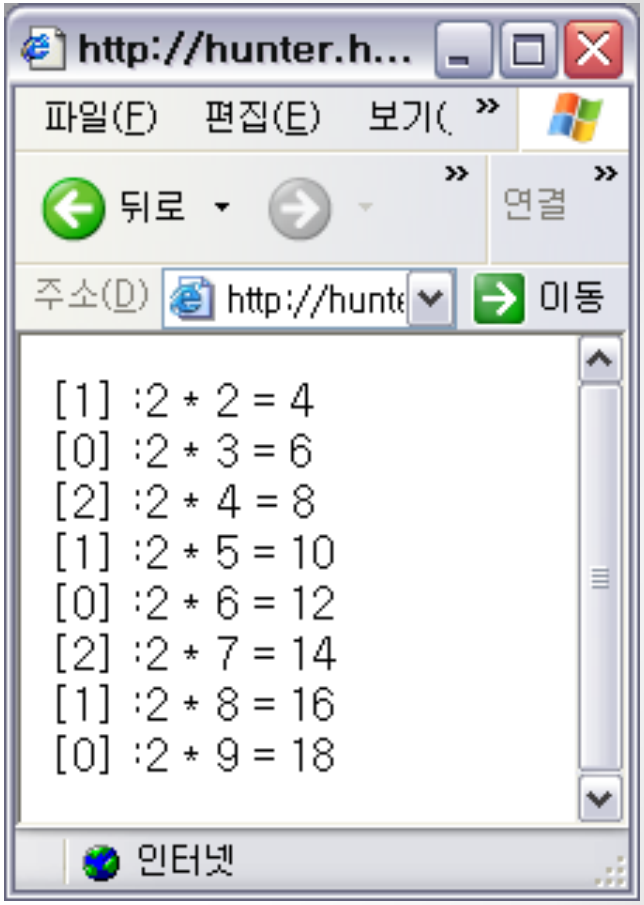


- # 다중 택일문 switch
  - switch문은 다중 택일문으로 조건의 상황에 따른 실행 코드를 선택할 수 있도록한다.
  - if문과의 차이는 if는 조건이 참과 거짓만을 구분하는 비해 조건의 상태에 따른 구분을 한다.

```
switch (조건) {
        case 경우 1:
                문장;
                break;
                …
        default :
                문장;
                break;
}
```

- 2단에서 곱해지는 단에 따라 출력을 다르게 한다.

```php
<?php
// 2dan-for-switch.php
for($i = 2; $i < 10; $i++) {
    $times = $i * 2;
    switch($times % 3) {
      case 0 :
        echo "[0] :" . "2 * $i = ", 2*$i, "<br />";
        break;
      case 1 :
        echo "[1] :" . "2 * $i = ", 2*$i, "<br />";
        break;
      default :
        echo "[2] :" . "2 * $i = ", 2*$i, "<br />";
        break;
    }
}
?>
```



다음 시간에는 …

- DB 와의 접속에 대해 알아봅니다.