

제어문

- 사용자의 입력 즉, 변수에 따라 다른 결과를 제어 할 수 있는 기능을 합니다.
- 제어문에도 다양한 종류가 있지만 여기서 알아볼 제어문은 반복문과 조건문입니다
- 반복문과 조건문을 사용에 능숙하다면 PHP의 절반이상 사용할 수 있습니다.

IF조건문

- IF조건문은 결과를 2가지 경우로 나누어 주는 제어문입니다
- IF를 사용한 후에 ELSE를 사용해야지만 2가지의 경우를 만들 수 있습니다.
- ELSEIF를 사용한다면 이 경우의 수를 늘려 줄 수 있습니다.

```

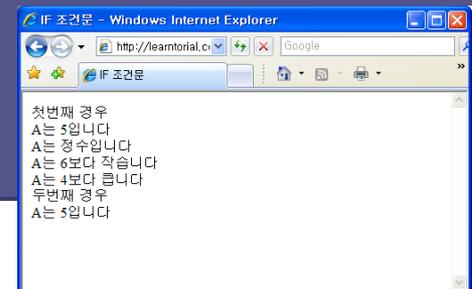
<html>
<head>
<title> IF 조건문 </title>
<body>
첫 번째 경우<br>
<?
  $a=5;
  if($a=="5"){
  echo "A는 5입니다<br>";
  }
  if($a=="5"){
  echo "A는 정수입니다<br>";
  }
  if($a=="5"){
  echo "A는 6보다 작습니다<br>";
  }
  if($a=="5"){
  echo "A는 4보다 큼니다<br>";

```

```

}
?>
두 번째 경우<br>
<?
  $a=5;
  if($a=="5"){
  echo "A는 5입니다";
  }
  elseif($a=="5"){
  echo "A는 정수입니다";
  }
  elseif($a=="5"){
  echo "A는 6보다 작습니다";
  }
  else{
  echo "A는 4보다 큼니다";
  }
?>
</body>
</html>

```



비교 연산자

- 조건문의 경우 변수의 값을 비교 하여 각각 다른 결과를 만들어 줍니다.
- 조건문에서 변수의 값을 비교하는데 사용하는 비교 연산자를 알아보겠습니다.

연산자 예	연산자 이름	설 명
$a==b$	Equal	a 와 b 가 같다
$a===b$	Identical	a 와 b 가 값과 자료형이 같다
$a!=b$	Not Equal	a 와 b 가 같지 않다
$a<>b$	Not Equal	a 와 b 가 같지 않다
$a!==b$	Not Identical	a 와 b 가 값과 자료형이 같지 않다
$a<b$	Less than	a 와 b 가 작다(미만)
$a>b$	Greater than	a 와 b 가 크다(초과)
$a<=b$	Less than or equal to	a 와 b 가 작거나 같다(이하)
$a>=b$	Greater than or equal to	a 와 b 가 크거나 같다(이상)

논리연산자

```
if($sex=="남자" and $age>=20 and  
    $hobby=="soccer"){  
    실행문  
}
```

- 비교연산자와 사용하여 조건문의 변수의 비교에 대한 폭을 넓혀줍니다
- 뿐만 아니라 비교연산자로 구현할 수 없는 특수한 조건에 대한 연산도 가능합니다.

연산자 예	연산자 이름	설 명
\$a and \$b	And	\$a 와 \$b 모두가 참이면 참
\$a or \$b	Or	\$a 나 \$b가 참이면 참
\$a xor \$b	Xor	\$a 와 \$b 중 하나만 참일 때만 참
! \$a	Not	\$a가 참이면 참
\$a && \$b	And	\$a 와 \$b 모두가 참이면 참
\$a \$b	Or	\$a 나 \$b가 참이면 참

반복문

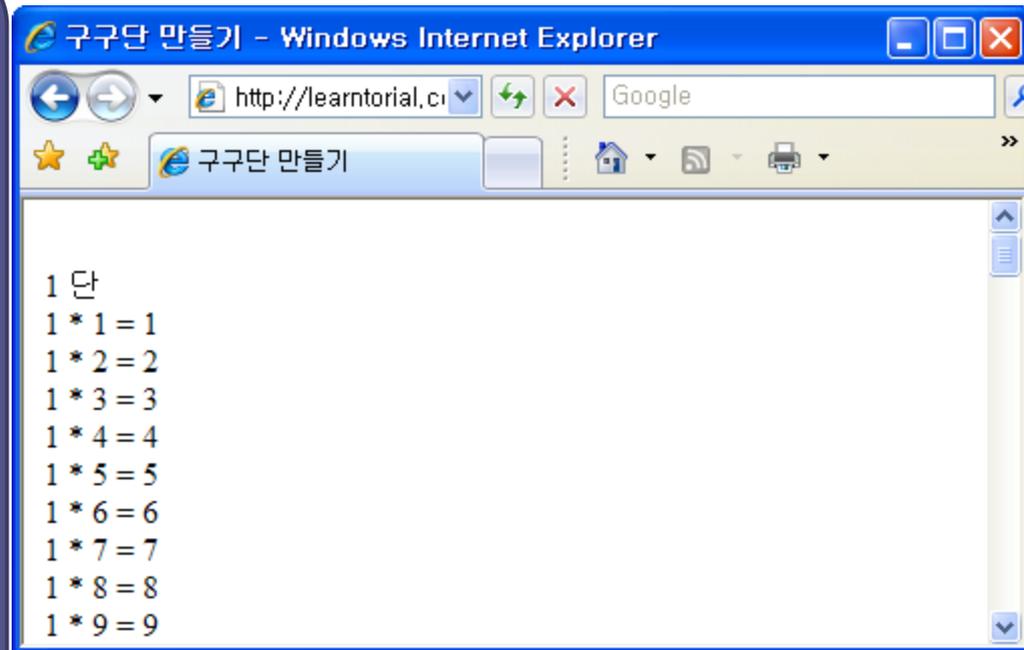
- IF조건문과 함께 제어문에서 중요한 반복문은 번거로운 작업을 대신 해줍니다.
Ex) 100칸 100줄 짜리 표를 만들기
- 반복문에는 2가지 종류가 있습니다. 조건을 만들어주는 반복문 for와 조건을 부여하지 않는 반복문 while가 있습니다.
- 여기서 알아볼 반복문은 for반복문입니다.

```
for(초기 값; 조건; 증감 값) {  
    실행문  
}
```

1부터 100까지 써 보기

```
for($i=1; $i<=100; $i++){  
    echo "$i<br>" ;  
}
```

```
<html>
  <head>
<title>구구단 만들기</title>
</head>
<body>
<?
  for($a=1; $a<=9; $a++){
    echo "<br> $a 단 <br>";
    for($b=1; $b<=9; $b++){
      $c=$a*$b;
      echo "$a * $b = $c<br>";
    }
  }
?>
</body>
</html>
```

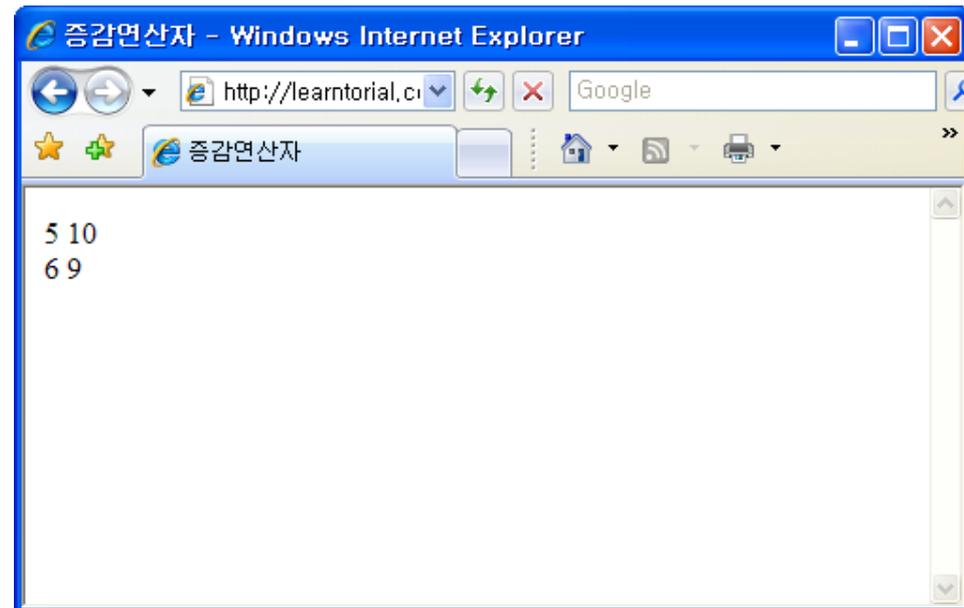


증감 연산자

- 조건문과 자주 사용하는 비교, 논리 연산자와 같이 반복문도 자주 사용하는 연산자 있습니다. 바로 증감 연산자입니다.
- 증감 연산자는 일정한 정도로 숫자나, 문자의 값을 증가 시켜 줍니다.
- ++, -- 기호를 사용합니다.

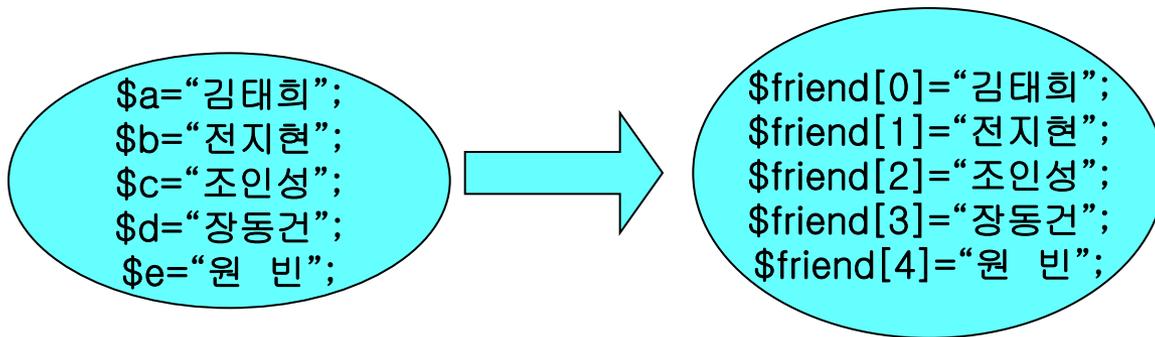
\$a++ \$a--

```
<html>
<head>
<title> 증감연산자 </title>
<body>
  <?
    $a=5;
    $b=10;
    echo $a++, "&nbsp;", $b--;
  ?>
```



배열의 사용

- 배열은 변수를 나열하기 위해서 사용합니다.
- 배열은 0을 시작으로 1씩 증가하면서 그 변수의 값을 저장하게 됩니다.



```
$friend=array(김태희, 전지현, 조인성, 장동건, 원빈);
```

```
for($i=0; $i<$cnt; $i++){  
    echo "$a[$i]<br>"; }
```

```

<html><head>
<title> 배열변수 </title></head>
<body>
<?
$a="김태희";
$b="전지현";
$c="조인성";
$d="장동건";
$e="원 빈";

echo "$a<br>";
echo "$b<br>";
echo "$c<br>";
echo "$d<br>";
echo "$e<br>";
?>

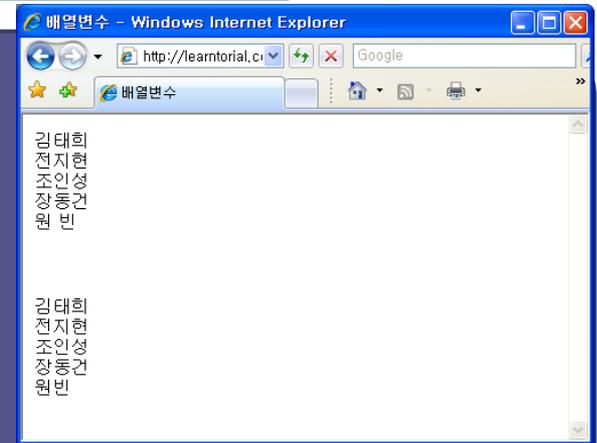
```

```

<br>
<br>
<br>
<?
$friend = array(김태희, 전지현,
조인성, 장동건, 원빈);
$cnt=count($friend);

for($i=0; $i<$cnt; $i++){
echo "$friend[$i]<br>";
}
?>
</body>
</html>

```



파일의 사용

- 파일함수들은 임의의 데이터 파일을 만들어서 그것을 관리하고 사용하는 함수
- 데이터베이스의 가장 기초적인 부분을 담고 있다.

```
<html>
<head>
<title> 카운터 </title>
<body>
<?
  $fp = fopen("visitor.txt","r");
  $cnt = fgets($fp,10000);

  $cnt=$cnt+1;
  fclose($fp);
?>
```

```
당신은 <?= $cnt?>번째 방문자
입니다.
<?
  $fp = fopen("visitor.txt","w");
  fputs($fp,$cnt);
  fclose($fp);
?>
</body>
</html>
```

파일함수 사용의 진행

1. fopen구문에 의해서 visitor.txt파일이 “r”(read)읽기 모드로 열라는 구문을 \$fp라는 변수에 할당했습니다.
2. fgets구문으로 \$fp에 의해 열려있는 visitor.txt의 내용, 다시 말해 그 동안 방문한 사람들의 숫자를 10000자리만큼 불러와서 그 숫자를 \$cnt라는 변수에 할당했습니다.
3. \$cnt라는 변수에 “+1”을 시켜줍니다.
4. \$fp변수로 열린 visitor.txt파일이 fclose구문에 의해서 닫히게 됩니다.
5. 한명이 증가된 \$cnt는 echo에 의해서 방문한 사용자에게 몇번째 방문자인지를 보여줍니다.
6. 다시 한번 fopen구문에 의해서 visitor.txt파일이 “w”(write) 쓰기 모드로 열라는 구문을 \$fp라는 변수에 정의되었습니다.
7. fputs구문으로 \$fp에 의해 열려있는 visitor.txt의 내용에 새로운 \$cnt변수값이 덮어졌습니다.
8. 새로운 방문자를 위해서 \$fp변수로 열린 visitor.txt파일이 fclose구문에 의해서 닫히게 됩니다.

