

도전! PHP

01. 다음 코드를 보고 \$b, \$c, \$d, \$e의 결과를 유추해 보시오.

```
<!DOCTYPE html>
<html lang="ko">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>연산자</title>
</head>
<body>
    <?php
        $a = 10;
        $b = $a++;
        $c = ++$a;
        $d = --$a;
        $e = $a--;
        print "$b $c $d $e";
    ?>
</body>
</html>
```

정답

10 12 12 10

결과 화면

해설

3번째 줄의 ++ 연산자는 \$a 뒤에 있기 때문에 저장된 값을 \$b에 저장한 다음 1 증가시킵니다. \$b는 100이 되고 \$a는 11이 됩니다. 4번째 줄의 ++ 연산자는 앞에 있기 때문에 \$a 변수의 값을 1 증가시킨 다음 \$c에 저장되어 두 변수 모두 12가 저장됩니다. 5번째 줄은 -- 연산자가 뒤에 있기 때문에 저장된 값 12를 \$d에 저장하고 나서 1 감소합니다. \$a는 11이 됩니다. 6번째 줄은 --가 앞에 있기 때문에 \$a에 저장된 값 1을 먼저 뺀 다음 \$e에 저장해 두 변수는 100이 저장됩니다.

02. \$a에 저장된 값이 숫자 100도 아니고, 문자열 “홍길동”도 아닌 논리 연산을 세 가지 이상 표현해 보세요.

정답

```
!(($a == 100) || ($a == "홍길동"))
!($a == 100) && (!$a == "홍길동")
($a == 100) && ($a != "홍길동")
```

해설

첫 번째 ((\$a == 100) || (\$a == "홍길동"))는 \$a에 저장된 값이 100이거나 홍길동이면 뜻합니다. 하지만 앞에 거짓을 의미하는 !가 있기 때문에

100이 아니거나 홍길동이 아니면 으로 해석됩니다.

두 번째는 && 연산자를 사용해 두 조건이 참이면 참이 됩니다. 즉 100도 아니고 홍길동도 아니라면 참이 되는 것입니다.

세 번째는 같지 않다는 뜻의 !=를 사용해 100이 아니거나 홍길동이 아니라면 참이 됩니다.

03. 다음 연산자들의 차이점을 설명해 보세요.

== 와 === 연산자

!= 와 ==, < > 연산자

<= 와 >= 연산자

정답

== 은 비교 대상이 되는 두 변수의 값이 동일하면 결과는 참이 됩니다.

=== 은 비교 대상이 되는 두 변수의 값과 데이터 형이 동일하면 참이 됩니다.

!= 와 !=, < > 연산자는 비교 대상이 되는 두 변수의 값이 모두 같지 않다는 뜻입니다. 이중 !=는 두 변수에 저장된 값이 같지 않으면서 데이터형도 다르면 결과는 참이 됩니다.

<= 와 >= 연산자는 크거나 또는 작거나 같다는 뜻입니다. 이 비교 연산자들은 방향이 다르면 무한루프 등으로 원하는 결과를 얻지 못하는 경우가 자주 있습니다.

04. 다음 프로그램을 보고 \$c에 저장될 결과를 설명해 보시오.

```
$a = array("김철민", "김재동");
```

```
$b = array("홍길동", "베트맨", "슈퍼맨", "X맨");
```

```
$c = $a + $b;
```

정답

김철민

김재동

슈퍼맨

X맨

결과 화면

김철민
김재동
슈퍼맨
X맨

해설

다음 코드를 실행하면 완성된 결과를 볼 수 있습니다. 두 배열을 더하게 되면 겹치지 않는 부분만 추가되게 됩니다. 만약 $c = b + c$ 라는 코드가 있다면 모두 겹치기 때문에 b 와 c 의 내용은 같습니다.

```
<!DOCTYPE html>
<html lang="ko">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>비교 연산자</title>
</head>
<body>
    <?php
        $a = array("김철민", "김재동");
        $b = array("홍길동", "베트맨", "슈퍼맨", "X맨");
        $c = $a + $b;
        for($i = 0; $i <= 4; $i++) {
            print $c[$i];
```

```

        print "<br>";
    }
?>
</body>
</html>

```

05. 다음 프로그램을 보고 변수 \$a, \$b, \$c, \$d의 결과를 설명해 보시오.

```

$a = 5 * 2 / 2 + 10 / 2;
$b = $a - 5 * 2 + ( 3 - 2 );
$c = ( $b - 5 ) * 2 + ( 3 - 2 );
$d = $c * 1 - ( 4 - 2 + ( 2 + 3 ));

```

정답

```

<!DOCTYPE html>
<html lang="ko">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>연산자의 우선순위</title>
</head>
<body>
    <?php
        $a = 5 * 2 / 2 + 10 / 2;
        $b = $a - 5 * 2 + ( 3 - 2 );
        $c = ( $b - 5 ) * 2 + ( 3 - 2 );
        $d = $c * 1 - ( 4 - 2 + ( 2 + 3 ));
    ?>
</body>
</html>

```

1
-7
-14

결과 화면

10
1
-7
-14

해설

\$a에 저장되는 값은 곱셈, 나눗셈 등이 왼쪽에서부터 순서대로 연산이 이뤄진 다음 덧셈이 이뤄집니다. $5 * 2$ 연산이 이뤄지고 이 결과를 2로 나눈 다음 10을 2로 나눈 5와 더해 10이란 결과 값이 \$a에 저장됩니다. \$b, \$c, \$d의 값은 어떻게 생겨나는지 천천히 분석해 보면 그리 어렵지는 않을 것입니다.