

php 내장함수

by 조쉬 posted **MAR 07, 2017**

*time()

-용도: 현재 시각을 timestamp값으로 구한다.

-방법: time()

*date()

-용도: 사용자가 지정한 형태로 시간을 표시. 특정시간의 날짜와 요일등을 배열로 리턴한다.

-방법: date("표시할 시간의 포맷형태", "특정한 timestamp값")

*mktime()

-용도: 지정된 날짜를 timestamp값으로 변환한다.

-방법: mktime(시,분,초,월,일,년)

*checkdate()

-용도: 날짜와 시간이 올바른 범위 안에 있는지 검사한다.

-방법: checkdate(월,일,년)

*getdate()

-용도: 특정timestamp값으로 시간,요일,날짜정보를 배열로 반환한다.

-방법: getdate(timestamp값) 또는 getdate()

*gettimeofday()

-용도: 현재 시스템의 현재 시간 정보를 배열로 리턴한다.

-방법: gettimeofday()

*gmmktime()

-용도: 그리니치표준 시간으로 지정한 날짜의 timestamp값을 리턴한다.

-방법: gmmktime(시,분,초,월,일,년)

*strftime()

-용도: 특정한 포맷으로 날짜 정보를 출력하는데 언어를 지정할수 있다.

-방법: strftime("표시할 시간의 포맷형태", "특정시간의 timestamp값")

*microtime()

-용도: 현재시간의 마이크로타임 값과 timestamp값을 표시한다.

-방법: microtime()

*localtime()

-용도: 현재 서버의 로컬 타임을 표시

-방법: localtime() 또는 localtime("timestamp값")

2. 문자열 처리함수

*addslashes()

-용도: 작은따옴표 큰따옴표와 같은 특정문자 앞에 역슬래시 문자를 붙인다.

-방법: addslashes()

*stripslashes()

-용도: addslashes로 역슬래시 처리된 문자를 원상태로 되돌린다.

-방법: stripslashes()

*htmlspecialchars()

-용도: HTML코드를 소스 그대로 출력해준다.

-방법: htmlspecialchars()

*nl2br()

-용도: 문자열에 포함된 개행 문자를
태그로 모두 바꿔준다.

-방법: nl2br()

*echo()

-용도: 문자열 출력

-방법: echo("변수명" 또는 "문자열")

*sprintf()

-용도: 포맷 문자열을 통해 지정해 준 포맷으로 문자열을 표시한다.

-방법: sprintf("포맷", "변수명")

*printf()

-용도: 지정할 수 있는 포맷에 따라 문자열을 출력한다.

-방법: printf("포맷", "변수명")

*explode()

-용도: 문자열을 지정해준 구분자로 분리하여 배열에 저장한다.

-방법: explode("구분자", "문자열인수")

*implode()

-용도: 구분자로 나누어 배열에 담은 문자열을 다시 지정해준 구분자로 연결하여 전체 문자열을 반환한다.

-방법: implode("구분자", 배열변수명)

*join()

-용도: implode()와 동일한 기능을 수행한다.

-방법: join("/", "배열변수명")

*split()

-용도: 주어진 문자열을 정규표현식의 패턴에 따라 분리하여 배열에 저장한다.

-방법: split("정규표현식", "문자열")

*strcmp()

-용도: 두개의 문자열을 비교하여 참이면 1, 거짓이면 0을 반환한다.

-방법: strcmp("문자열", "문자열") 또는 strcmp("문자열", "변수명") 또는 strcmp(변수명, 변수명)

*substr()

-용도: 문자열에서 특정시작 문자부터 길이만큼의 문자열을 추출하여 반환한다.

-방법: substr("문자열", 시작문자번호, 문자열의 길이)

*strchr()

-용도: 문자열이 처음으로 나타나는 위치부터 끝까지 반환

-방법: strchr("문자열", "찾을 문자열")

*strstr()

-용도: strchr() 함수와 동일한 수행을 한다.

-방법: strstr("문자열", "찾을 문자열")

`*strrchr()`

-용도: 문자열이 마지막으로 나타나는 위치부터 끝까지 반환한다.

-방법: `strrchr("문자열", "찾을문자열")`

`*strpos()`

-용도: 문자열 대신 처음으로 나타나는 위치를 숫자로 반환, 이때 숫자는 0부터 시작.

-방법: `strpos("문자열", "찾을 문자열")`

`*strrpos()`

-용도: 찾고자 하는 문자열에서 마지막으로 나타나는 위치를 반환한다.

-방법: `strrpos("문자열", "찾을 문자열")`

`*strlen()`

-용도: 문자열의 길이를 반환한다.

-방법: `strlen($변수명)` 또는 `strlen("문자열")`

`*strtolower()`

-용도: 알파벳 대문자를 모두 소문자로 바꾸어 반환한다.

-방법: `strtolower("문자열")`

`*strtoupper()`

-용도: 알파벳 대문자를 소문자로 바꾸어 반환한다.

-방법: `strtoupper("문자열")`

`*ucfirst()`

-용도: 알파벳 문자열의 첫 글자를 대문자로 반환한다.

-방법: `ucfirst("문자열")`

`*ucwords()`

-용도: 알파벳 문자열의 첫 글자를 대문자로 반환한다.

-방법: `ucwords("문자열")`

`*strrev()`

-용도: 인자로 전달한 문자열의 순서를 역전시킨다.

-방법: `strrev("문자열")`

`*strtr()`

-용도: 문자열에서 변경하고자 하는 문자열을 변경할문자열로 1:1 변경해준다.

-방법: `strtr("문자열", "변경하고자하는 문자열", "변경할 문자열")`

`*str_replace`

-용도: 문자열 중에 변경하고자 하는 문자열을 찾아 변환한다.

-방법: `str_replace("찾을 문자열", "변경할 문자열", "문자열 원본")`

`*chop()`

-용도: 문자열의 뒷부분의 공백 문자를 제거한 후 반환한다.

-방법: `chop("문자열")`

`*trim()`

-용도: 문자열의 앞과 뒤에 있는 공백을 제거한 후 문자열을 반환한다.

-방법: `trim("문자열")`

`*quotemeta()`

-용도: 메타문자 앞에 역슬래시를 붙인다.

-방법: `quotemeta()`

`*ord/chr()`

-용도: `ord()` 함수는 문자열에서 첫번째 문자에 아스키코드값을 반환하고 `chr()` 함수는 아스키함수에 해당하는 문자를 반환한다.

-방법: ord("문자열"), chr("아스키값")

*parse_str()

-용도: get방식과 같은 형태의 변수형을 변수와 값으로 받을 수 있게 해준다.

-방법: parse_str(변수명)

3.파일관련함수

*fopen()

-용도: 로컬경로의 파일이나 url경로의 파일을 읽기 모드나 쓰기 모드로 열어준다.

-방법: fopen("절대경로명/파일", "모드")또는 fopen("url주소", "모드")

*fclose()

-용도: fopen으로 열린 파일을 닫는다.

-방법: fclose(\$fp)

*fread()

-용도: 파일포인터가 가리키는 파일의 내용을 지정한 크기만큼 읽어 들인다.

-방법: fread(\$fp,"읽어들일 바이트")

*feof()

-용도: 현재 파일 포인터가 파일의 끝인지를 알려주는 함수

-방법: feof(\$fp)

*fgets()

-용도: 파일을 읽어올때 한줄씩 읽어온다.

-방법: fgets(\$fp,"읽어들일 바이트")

*fputs()

-용도: 파일을 a 또는 w 모드로 열어 문자열을 파일에 기록한다.

-방법: fputs(\$fp, "문자열")

*fwrite()

-용도: 파일을 기록한다.

-방법: fwrite(\$fp, "문자열")

*fpassthru()

-용도: 파일포인터의 현재 위치에서 파일의 끝까지 읽어 들이도록 한다.

-방법: fpassthru(\$fp)

*readfile()

-용도: fpassthru와 비슷하지만 fopen()없이 직접 바로 파일을 읽어들일 수 있다. 읽어들인 파일의 절대경로를 포함한다.

-방법: readfile("파일의 절대경로 또는 상대경로/파일명")

*fgetc()

-용도: 파일로부터 한문자를 읽어들인다.

-방법: fgetc(\$fp)

*filesize()

-용도: 파일의 크기를 바이트수로 반환한다.

-방법: filesize("파일경로/파일명")

*file_exists()

-용도: 파일이름으로 지정한 파일이 존재하는지 검사한다.

-방법: file_exists("파일경로/파일명")

*is_file()

-용도: 파일이름으로 지정한 이름이 파일면 true를 반환한다.

-방법: `is_file("파일경로/파일명")`

`*is_dir()`

-용도: 파일이름으로 지정한 이름이 디렉토리일경우 `true`를 반환한다.

-방법: `is_dir("파일경로/파일명")`

`*is_link()`

-용도: 파일이름으로 지정한 이름이 심볼릭 링크일경우 `true`를 반환한다.

-방법: `is_link("파일경로/파일명")`

`*is_executable()`

-용도: 파일이름으로 지정한 파일이 실행 가능한 파일이면 `true`를 반환한다.

-방법: `is_executable("파일경로/파일명")`

`*is_readable()`

-용도: 파일이름으로 지정한 파일이나 디렉토리가 읽기 가능한경우 `true`를 반환한다

-방법: `is_readable("파일경로/파일명 또는 디렉토리명")`

`*is_writable()`

-용도: 파일이름으로 지정한 파일이나 디렉토리가 쓰기 가능한 경우 `true`를 반환한다

-방법: `is_writable("파일경로/파일명 또는 디렉토리명")`

`*copy()`

-용도: 파일을 복사하고 제대로 수행이 되면 `true`를 반환한다.

-방법: `copy("파일경로/복사할 파일명", "파일경로/복사된 파일명")`

`*rename()`

-용도: 파일을 이름을 바꾼다. 제대로 수행이 되면 `true`를 반환한다.

-방법: `rename("파일경로/변경할 파일명", "파일경로/변경후 파일명")`

`*unlink()`

-용도: 파일을 삭제한다. 제대로 수행될 경우 `true`를 반환한다.

-방법: `unlink("파일경로/삭제할 파일명")`

`*mkdir()`

-용도: 디렉토리를 생성하고 동시에 권한도 함께 지정할 수 있다.

-방법: `mkdir("경로/디렉토리명")`

`*rmdir()`

-용도: 디렉토리 삭제

-방법: `rmdir("경로/디렉토리명")`

`*basename()`

-용도: 경로명과 파일명을 받아 파일명만을 반환한다.

-방법: `basename("파일경로/파일명")`

`*dirname()`

-용도: 경로명과 파일명을 받아 경로명만을 반환한다.

-방법: `dirname("파일경로/파일명")`

`*file()`

-용도: 줄단위로 읽어서 각 줄의 문자열을 배열로 하여 반환한다.

-방법: `file("파일경로/파일명")`

4.디렉토리 관련함수

`*opendir()`

-용도: 지정한 디렉토리에 대한 핸들을 얻는다.

-방법: opendir("디렉토리 경로")

*closedir()

-용도: opendir() 함수 호출로 열어둔 디렉토리에 대한 핸들을 닫는다.

-방법: closedir(\$od)

*readdir()

-용도: 디렉토리에 있는 모든 디렉토리 및 파일들의 이름을 읽어 배열의 형태로 반환

-방법: readdir(\$od)

*rewinddir()

-용도: 디렉토리 핸들의 포인터 위치를 제일 처음으로 되돌린다.

-방법: rewinddir(\$od)

*chdir()

-용도: 지정한 디렉토리로 바꾼다.

-방법: chdir("경로명/디렉토리명")

*dir()

-용도: 디렉토리 관련 함수를 사용할 수 있는 클래스

-방법: dir("경로/디렉토리명")

5. URL관련함수

*parse_url()

-용도: url을 입력하면 해당 url에 대한 정보를 연관 배열로 저장하여 반환

-방법: parse_url(URL주소)undefined

*urlencode()

-용도: 데이터를 url인코딩해준다.

-방법: urlencode("문자열")

*urldecode()

-용도: url을 통해 넘어온 값 중 인코딩된 문자열을 디코딩한다.

-방법: urldecode("문자열")

6. 변수 및 수학 관련함수

*gettype()

-용도: 변수형을 반환한다.

-방법: gettype("변수명")

*settype()

-용도: 변수형을 변환한다.

-방법: settype("변수명", "변수타입")

*intval()

-용도: 변수값을 정수형 값으로 변환

-방법: intval("변수값")

*doubleval()

-용도: 변수값을 실수형 값으로 변환한다.

-방법: doubleval("변수값")

*strval()

-용도: 변수를 문자형 값으로 변환한다.

-방법: strval("변수값")

*isset()

-용도: 변수가 존재하는지 확인한다.

-방법: isset("변수명")

*unset()

-용도: 변수를 삭제한다.

-방법: unset("변수명")

*empty()

-용도: 변수에 값이 존재하는지 확인하여 없으면 true를 있으면 false를 반환한다.

-방법: empty("변수명")

*is_array()

-용도: 변수의 형이 배열인지 확인한다.

-방법: is_array("변수명")

*is_double

-용도: 변수의 형이 더블형인지 확인한다.

-방법: is_double("변수명")

*abs()

-용도: 변수의 절대값을 반환

-방법: abs("변수명")

*round()

-용도: 변수의 반올림한 값을 구한다.

-방법: round("변수명")

*ceil()

-용도: 변수 값의 무조건 올림 값을 구한다.

-방법: ceil("변수명")

*floor()

-용도: 변수값의 무조건 내림 값을 구한다.

-방법: floor("변수명")

*pow()

-용도: 인자 값만큼 제곱승 한 값을 구한다.

-방법: pow("변수명", "제공할 횟수")

*max & min()

-용도: 최대값과 최소값을 구한다.

-방법: max("숫자값", "숫자값", ...) min("숫자값", "숫자값", ...)

7.기타 알아두어야 할 함수

*count()/sizeof()

-용도: 배열의 크기를 구한다.

-방법: count("변수명") sizeof("변수명")

*each()/list()

-용도: 배열의 내용을 출력할때 사용한다.

-방법: each("배열변수명"), list("배열의 인자를 받을 변수명", "배열의 인자를 받을 변수명"...)

*end()/next()/reset()

-용도: 현재 배열의 포인터를 끝이나 다음으로 이동시킨다. reset은 배열포인터를 다시 처음으로 되돌린다.

-방법: end("배열변수명") next("배열변수명")

*exec()

-용도: 리눅스 서버의 내부 명령어를 사용하게 한다.

-방법: `exec("리눅스명령어", "결과를 저장할 변수명", "에러발생시 저장할 변수명")`

*number_format()

-용도: 천단위마다 ","를 붙여준다.

-방법: `number_format("변수명", "소수점자리수", "소수점 표시마크변경", "천단위 표시 마크변경")`

*show_source() / highlight_string()

-용도: php파일의 소스를 보여준다.

-방법: `show_source("경로명/파일명") highlight_string("변수명")`

*uniqid()

-용도: 유일한 아이디 값을 만들어준다

-방법: `uniqid()`