jinyPHP 1권 - 18. 콘솔작업

**18. 콘솔**

PHP는 웹 브라우저를 통해 실행뿐만 아니라 콘솔에서 직접 명령으로 PHP를 실행할 수 있습니다.

콘솔 명령을 다른 말로는 CLI(command Line Interface)라고도 합니다. PHP CLI 기능은 PHP 4.3.x 버전 이후부터 적용된 기능입니다. PHP CLI는 SAPI(Server Application Programming Interface) 형식을 지원합니다.

**18.1 콘솔이란?**



컴퓨터와 사용자 사이에 대화를 할 수 있는 커맨드 기반의 입출력 장치를 말합니다. 옛날 DOS 나 리눅스의 bash shell, 윈도우 CMD 창 등이 콜솔 명령 작업을 할 수 있는 환경입니다.

콘솔을 이용하면 명령을 통해 컴퓨터에게 직접 작업을 실행하거나 결과를 얻을 수 있습니다.

**18.2 콘솔 옵션**

콘솔 창에서 php -help 명령을 입력하면 php 콘솔 실행과 관련된 여러 명령 옵션을 확인할 수 있습니다.

|명령|

**#] php -help**

|  |
| --- |
| Usage: php [options] [-f] <file> [--] [args...]  php [options] -r <code> [--] [args...]  php [options] [-B <begin\_code>] -R <code> [-E <end\_code>] [--] [args...]  php [options] [-B <begin\_code>] -F <file> [-E <end\_code>] [--] [args...]  php [options] -S <addr>:<port> [-t docroot]  php [options] -- [args...]  php [options] -a  -a Run interactively  -c <path>|<file> Look for php.ini file in this directory  -n No php.ini file will be used  -d foo[=bar] Define INI entry foo with value 'bar'  -e Generate extended information for debugger/profiler  -f <file> Parse and execute <file>.  -h This help  -i PHP information  -l Syntax check only (lint)  -m Show compiled in modules  -r <code> Run PHP <code> without using script tags <?..?>  -B <begin\_code> Run PHP <begin\_code> before processing input lines  -R <code> Run PHP <code> for every input line  -F <file> Parse and execute <file> for every input line  -E <end\_code> Run PHP <end\_code> after processing all input lines  -H Hide any passed arguments from external tools.  -S <addr>:<port> Run with built-in web server.  -t <docroot> Specify document root <docroot> for built-in web server.  -s Output HTML syntax highlighted source.  -v 버전을 확인합니다.  -w Output source with stripped comments and whitespace.  -z <file> Load Zend extension <file>.  args... Arguments passed to script. Use -- args when first argument  starts with - or script is read from stdin  --ini Show configuration file names  --rf <name> Show information about function <name>.  --rc <name> Show information about class <name>.  --re <name> Show information about extension <name>.  --rz <name> Show information about Zend extension <name>.  --ri <name> Show configuration for extension <name>. |

**PHP 버전 확인 -v**

-v 옵션은 버전을 확인할 수 있는 콘솔 명령 옵션입니다.

|명령|

#] php -v

|  |
| --- |
| C:\php-7.1.4-Win32-VC14-x86>php -v  PHP 7.1.4 (cli) (built: Apr 11 2017 20:08:12) ( ZTS MSVC14 (Visual C++ 2015) x86 )  Copyright (c) 1997-2017 The PHP Group  Zend Engine v3.1.0, Copyright (c) 1998-2017 Zend Technologies |

**PHP 정보 확인 -i**

-i 옵션은 PHP의 환경 설정 정보를 확인할 수 있습니다. 웹에서 phpinfo() 함수를 실행하는 것과 동일합니다.

|명령|

#] php -i

|  |
| --- |
| C:\php-7.1.4-Win32-VC14-x86>php -i  phpinfo()  PHP Version => 7.1.4  System => Windows NT LAPTOP-M0820HEF 10.0 build 14393 (Windows 10) i586  Build Date => Apr 11 2017 19:53:44  Compiler => MSVC14 (Visual C++ 2015)  Architecture => x86  Configure Command => cscript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=c:\php-sdk\oracle\x86\instantclient\_12\_1\sdk,shared" "--with-oci8-12c=c:\php-sdk\oracle\x86\instantclient\_12\_1\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--with-mcrypt=static" "--without-analyzer" "--with-pgo"  Server API => Command Line Interface  Virtual Directory Support => enabled  Configuration File (php.ini) Path => C:\Windows  ….  ….  ….  …. |

**PHP 강조 표시 -s**

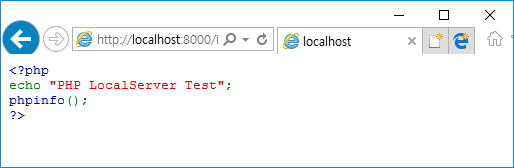
-s 옵션은 PHP 소스의 문법을 강조하여 HTML 파일 형태로 출력합니다.

**예제 파일** index.php

|  |
| --- |
| <?php  echo "PHP LocalServer Test";  phpinfo();  ?> |

|  |
| --- |
| C:\php-7.1.4-Win32-VC14-x86>php -s index.php  <code><span style="color: #000000">  <br /></span><span style="color: #007700">echo&nbsp;</span><span style="color: #DD0000">"PHP&nbsp;LocalServer&nbsp;Test"<br /></span><span style="color: #0000BB">?&gt;fo</span><span style="color: #007700">();  </span>  </code> |

**변환된 HTML파일:**



**PHP 명령행 실행 -w**

-w 옵션은 소스의 주석과 공백을 제거한 소스를 다시 출력합니다.

|  |
| --- |
| C:\php-7.1.4-Win32-VC14-x86>php -w index.php  <?php  echo "PHP LocalServer Test"; phpinfo(); ?> |

**PHP 명령행 실행 -r**

-r 옵션은 php 콘솔상에서 PHP의 명령어를 직접 입력하여 실행할 수 있습니다.

|  |
| --- |
| C:\php-7.1.4-Win32-VC14-x86>php -r "echo 'hello world!';"  hello world! |

위의 콘솔 예제처럼 스크립트 파일을 작성하지 않고도 직접 간단한 명령을 php 입력으로 실행할 수 있습니다.

**PHP 대화식 실행 -a**

-a 옵션은 PHP 콘솔에서 대화식으로 PHP 명령을 실행하고 결과를 확인할 수 있습니다.

|  |
| --- |
| C:\php-7.1.4-Win32-VC14-x86>php -a  Interactive shell  php > echo "hello world!";  hello world!  php > |

대화식을 종료할 때는 exit 를 명령하면 됩니다.

**PHP 문법 검사 -l**

-l 옵션은 PHP의 문법적 오류를 체크합니다. 오류가 없는 경우에는 No syntax errors를 출력합니다.

|  |
| --- |
| C:\php-7.1.4-Win32-VC14-x86>php -l index.php  No syntax errors detected in index.php |

**PHP 모듈 출력 -m**

-m 옵션은 PHP의 내장된 PHP모듈과 Zend 모듈을 출력합니다.

|  |
| --- |
| C:\php-7.1.4-Win32-VC14-x86>php -m  [PHP Modules]  bcmath  calendar  …  …  xmlwriter  zip  zlib  [Zend Modules] |

**18.3 콘솔 실행**

PHP스크립트는 웹 브라우저를 통해 실행도 가능하나 콘솔로 실행할 수 도 있습니다. PHP 스크립트를 콘솔로 실행을 할 수 있다는 것은 몇 가지 장점이 있습니다.

|명령|

**#] php 실행 파일명.php**

php 명령과 php 파일명을 콘솔상에 적으면 PHP는 콘솔상 php 스크립트를 실행하여 콘솔 화면에 출력합니다.

**예제**

|  |
| --- |
| C:\php-7.1.4-Win32-VC14-x86>php ./jinyphp/1-03/hello.php  <!DOCTYPE html>  <html>  <body>  <h1> PHP 페이지를 만들어 봅니다.</h1>  Hello World! </body>  </html>  C:\php-7.1.4-Win32-VC14-x86> |

PHP의 콘솔 실행 가능 여부는 PHP를 웹 페이지 제작 이 외에도 시스템의 처리 등을 위한 스크립트로도 사용이 가능하다는 것입니다.

윈도우의 스케줄러, 리눅스의 crontab에 넣어서 정기적으로 php 스크립트를 실행할 수 있습니다.

**18.4 실행 인자**

PHP 소스를 실행할 때 외부에서 입력하는 데이터 값을 입력 받아 처리를 할 수 있습니다.

웹기반에서는 URL 의 GET 이나 폼의 POST 값을 통하여 PHP 소스에 데이터를 전달할 수 있었습니다.

콘솔에서 PHP 스크립트 실행과 더불어 인자도 같이 전달 가능합니다. 이것을 외부 실행 인자라고 이야기합니다.

**예제 파일** consol-01.php

|  |
| --- |
| <?php  var\_dump($argv);  ?> |

위와 같은 예제가 있습니다. 예제는 외부에서 입력된 배열변수의 값 $argv 를 출력하는 것입니다.

다음과 같이 콘솔상에서 php 스크립트 실행과 외부 입력값을 전달할 수 있습니다.

|명령|

**#] php consol-01.php arg1 arg2 arg3**

**결과)**

array(4) {

[0]=>

string(23) "./jinyphp/consol-01.php"

[1]=>

string(4) "arg1"

[2]=>

string(4) "arg2"

[3]=>

string(4) "agr3"

}

$argv 배열 변수는 콘솔을 통하여 스크립트를 실행할 때 외부에서 입력된 값을 배열 형태로 참조할 수 있는 특별한 변수입니다.

첫 번째 값으로는 실행하는 스크립트 파일의 경로를 가지고 있습니다.

두 번째부터는 인자값으로 전달 되는 값들이 순차적으로 저장되고 참조가 가능합니다.

18.5 백그라운드 실행

PHP 스크립트는 리눅스와 같은 시스템에서 백그라운드로 실행시킬 수 있습니다. 리눅스의 명령어 & 는 명령 동작을 백드라운드로 실행합니다.

#] php test.php **&**