

PHP Programming Reference Sheet – Intermediate

This is the companion cheat sheet to the “PHP Programming Reference Sheet – The Basics”. We cover more advanced topics in the PHP such as conditional statements and reading and writing files.

1) PHP Conditional statements – if statement

Here is an example of an if statement.

```
$x = 8;
if($x < 10) { print("x is less than 10");}
```

2) PHP Conditional statements – if-else statement

Here is an example of if and else.

```
$numericalGrade = 45;
if($numericalGrade >= 50)
    print("Grade: Pass");
else{
    print("Grade: Fail");
}
```

3) PHP Conditional statements – if-elseif-else statement

Here is an if-elseif-else example when you have more than 2 conditions.

```
$numGrade = 85;
if( ($numGrade >= 50) && ($numGrade < 60) ){
    print("Student Grade: D");
} elseif ( ($numGrade >= 60) && ($numGrade < 70) ) {
    print("Student Grade: C");
} elseif (($numGrade >= 70) && ($numGrade < 80) ){
    print("Student Grade: B");
} elseif ($numGrade >= 80){
    print("Student Grade: A");
} else{
    print("Student Grade: F");
}
```

4) PHP Switch Statement

The PHP Match statement is a good alternative to if-elseif-else syntax.

```
$grade = "pass";
switch ($grade) {
    case "pass":
        echo "Student has a passing grade";
        break;
    case "fail":
        echo "Student has a failing grade";
        break;
    default:
        echo "invalid entry";
}
```

5) PHP Multiline Strings

PHP multiline strings are enclosed as follows.

```
$myMultilineString = <<<'EOD'
This is my mutliline
string
EOD;
```

6) PHP Dates

PHP comes with prebuilt dates and time functions

```
$cYear = date("Y") . "\n"; //year
$cDay = date("l") . "\n"; //day
$cMonth = date("F") . "\n"; //month
```

For a full list of date and time functionality go to an internet search engines and search for “PHP date function ABC”. Replace ABC with your specifics.

7) PHP Math functions

PHP comes with prebuilt math functions.

```
echo(sqrt(9)."<br>");//square root function
echo 10%3; //finding remainder with modulus operator
```

For a full list go to an internet search engine and search for “PHP math function ABC”. Replace ABC with the function you are interested in.

8 a) Reading Files - text files (line by line)

Reading text files in PHP is easy.

```
foreach(file('myfile.txt') as $currentline) {
    //do something with the line text
    echo $currentline;
}
```

8 b) Reading Files - text files into a variable

Reading text files into a variable is easy.

```
$fileContent = file_get_contents('myfile.txt');
```

9) Reading Files - JSON files

We can read json files in a similar way.

```
$jsonText =
json_decode(file_get_contents("myjsonfile.json"), true);
print_r($jsonText);
```

10) Writing Files - text files

You can use the file_put_contents method.

```
$myFilename = 'myfilename.txt';
$myData = 'Some data to write to file';
file_put_contents($myFilename, $myData);
```

11) Writing Files - JSON files

You can use a json module to write json content to a file.

```
$jsonData = json_encode($jsonArray);
file_put_contents('myjsonfile.json', $jsonData);
```

The \$jsonArray variable holds your JSON data in PHP array.

12) PHP file imports

As your PHP programs get larger you will likely need to include external files for additional functionality. There are 2 ways.

```
include 'utility.php';
//or you can use
require 'utility.php';
```

13) Command Line User Input

Obtaining user input is very common for command line PHP scripts. Here is an example script to obtain user first name and last name from a command line PHP script.

```
echo "Please enter your first name: ";
$fname = rtrim(fgets(STDIN));
echo "Please enter your last name: ";
$lname = rtrim(fgets(STDIN));
print("You have entered: " . $fname . " " . $lname)
```

14) PHP Command Line Arguments

When creating PHP scripts, sometimes you may wish to pass command line arguments to your programs. For example, if you have a PHP script called **myscript.php** and you want to pass a numerical value 10 which will be used later in the program you can use this command line:

```
php myscript.php 10
```

Then later in your script you can access the passed argument like this:

```
$cmdArgument = $argv[1];
```