

PHP Programming Reference Sheet – Advanced

This the companion cheat sheet to the “PHP Programming Reference Sheet – Intermediate”. We cover more advanced topics in the PHP Programming language including error handling, classes, and objects.

1) Error Handling: Try and Catch

PHP try & catch are similar to try & catch in Java.

```
//we will simulate division by zero exception
try {
    //some code with exceptions
    $y=10/0;
    //some more code
} catch(Exception $e) {
    echo 'Message: ' . $e->getMessage();
    //do something to fix the issue
}
```

Note that you can create your own custom exceptions and throw exceptions. To create your own exception, you just need to extend the PHP built in Exception class.

2) Variable Scope

In general, variables defined outside of classes and functions have global scope providing they are referenced by the global keyword within the function. Here is an example.

```
$a=200;
$b="some string";

function myFunction() {
    global $a;
    $a = 300;
    $b ="another string"; #this creates a function variable
}
myFunction();

print($a."\n");
print($b);
```

Here is the output of code above:

```
300
some string
```

Note that variable “b” does not see another variable “b” created outside of the function while the variable “a” within the function is declared of the global scope outside of the myFunction().

3) PHP Functions

There are many ways to define functions. Here are some examples.

```
function myFirstFunction(){ #no arguments
    print("Hello World\n");
}
function mySecondFunction($age){ //single argument
    print("You are " . $age . " years old\n");
}
function myThirdFunction($numA, $numB){ //2 arguments
    print("The sum of 2 numbers is: " . ($numA + $numB)
    ."\n");
}
function myFourthFunction(...$aParams) { //arbitrary
parameters
    print_r($aParams);
}
function myFifthFunction($name = "John"){ #default
parameter
    print("Your name is " . $name);
}
$someVar="Old Value";
function mySixthFunction(&$someVar){ #pass by
reference
    $someVar = "New value";
}
//calling functions above
myFirstFunction();
mySecondFunction(20);
myThirdFunction(10, 30);
myFourthFunction($color = "Red", $temp = 25);
myFifthFunction();
mySixthFunction($someVar);
print("\n".$someVar);
```

Here is the output of code above:

```
Hello World
You are 20 years old
The sum of 2 numbers is: 40
Array
(
    [0] => Red
    [1] => 25
)
Your name is John
New value
```

4) Classes and Objects

The following example creates 2 classes, Car and ElectricCar. The Car class is the parent class while ElectricCar class inherits methods and variables of the Car (parent) class.

```
//Sample Car class with 2 functions
class Car {
    public $model;
    public $make;
    public function __construct($model, $make) {
        $this->model = $model;
        $this->make = $make;
        $this->year = 2021; //hard coded
    }
    public function myPrintCarInformation() {
        print("\nMake: " . $this->make);
        print("\nModel: " . $this->model);
    }
    public function myCustomHonkFunction() {
        print("\nHonk!! Honk!!");
    }
}
//Instantiate new car
$car = new Car("Honda", "Civic");
$car->myPrintCarInformation();
$car->myCustomHonkFunction();
//sample child class illustrating class inheritance
class ElectricCar extends Car {
    static $color= "Black"; //class variable
    public function myCustomSignature() {
        print("\nHi, I am an electric car!");
    }
}
#Instantiate a new (electric) car
$car2 = new ElectricCar("Tesla", "Model S");
$car2->myPrintCarInformation();
$car2->myCustomSignature();
print("\n".$car2::$color); //printing class variable
print("\n".$car2->year); //printing instance variable
```

Here is the output of code above:

```
Make: Civic
Model: Honda
Honk!! Honk!!
Make: Model S
Model: Tesla
Hi, I am an electric car!
Black
2021
```