PHP – Special Types

PHP’s two data types – **resource** and **NULL** – are classified as **special types**. An object of resource type refers to external resources like database connection, file streams etc. On the other hand, a NULL data type is a variable without any data assigned to it. In this chapter, we shall learn more about these types.

Resource Type

A PHP program often needs to interact with an external environment such as a database, or a disk file etc. These are treated as resources in PHP. Resource is a special data type that refers to any such external resource. PHP uses relevant functions to create these resources. For example, fopen() function opens a disk file and its reference is stored in a resource variable.

PHP's Zend engine uses reference counting system. Hence, a resource with zero reference count is destroyed automatically by garbage collector and the memory used by resource data type need not be freed manually.

Different built-in PHP functions return respective resource variables. Subsequently, PHP uses them for interacting with the corresponding external environment. For example, the fopen() function returns a file resource, which acts as a file handle and the read/write operations on the file are facilitated by this resource variable.

The following table summarizes different functions that return resource variables −

Resource Type	Built-in functions		Definition
	Produced	**Sold**	
bzip2	bzopen()	bzclose()	Bzip2 file
curl	curl_init()	curl_close()	Curl session
ftp	ftp_connect(),	ftp_close()	FTP stream
mssql link	mssql_connect()	mssql_close()	Link to Microsoft SQL Server database
mysql link	mysql_connect()	mysql_close()	Link to MySQL database
mysql result	mysql_db_query(),	mysql_free_result()	MySQL result

oci8 connection	oci_connect()	oci_close()	Connection to Oracle Database
ODBC link	odbc_connect()	odbc_close()	Link to ODBC database
pdf document	pdf_new()	pdf_close()	PDF document
stream	opendir()	closedir()	Dir handle
stream	fopen(), tmpfile()	fclose()	File handle
socket	socket_create()	Socket_close()	Socket handle
xml	xml_parser_create()	xml_parser_free()	XML parser
zlib	gzopen()	gzclose()	gz-compressed file
zlib.deflate	deflate_init()	None()	incremental deflate context
zlib.inflate	inflate_init()	None()	incremental inflate context

PHP has get_resource_type() function that returns resource type of a variable.

```
get_resource_type ( resource $handle ) : string
```

where $handle is the resource variable whose type is to be obtained. This function returns a string corresponding to resource type.

There is also get_resource_id() function an integer identifier for the given resource.

```
get_resource_id(resource $resource): int
```

Example

This function provides a type-safe way for generating the integer identifier for a given resource.

```
<?php
   $fp = fopen("hello.php", "r");
   $resource = get_resource_type($fp);
   $id = get_resource_id($fp);
   echo "The resource type is : $resource The resource ID is : $id";
?>
```

It will produce the following **output** –

```
The resource type is : stream The resource ID is : 5
```

NULL type

In PHP, a variable with no value is said to be of null data type. Such a variable has a value defined as NULL. A variable can be explicitly assigned NULL or its value been set to null by using unset() function.

```
$var=NULL;
```

It is possible to cast variable of other type to null, although casting null to other type has been deprecated from PHP 7.2. In earlier versions, casting was done using (unset)$var syntax

Example

The following example shows how to assign NULL to a variable

Open Compiler

```
<?php
   $var=NULL;
   var_dump($var);
?>
```

It will produce the following **output** –

```
NULL
```

Example

The following example performs null variable to other primary variables –

Open Compiler

```
<?php
   $var = NULL;
   var_dump( (int)   $var);
   var_dump((float)$var);
   var_dump((bool)  $var) ;
   var_dump( (boolean) $var);
?>
```

It will produce the following **output** −

```
int(0)
float(0)
bool(false)
bool(false)
```