Table 1: Different datatypes available in PHP

S.no	Type	Datatype name	Description	Values
1.	Scalar	Boolean	Boolean value	TRUE or FALSE
2.		Integer	Numbers from set Z (Base 10,16,8)	15,200,5,-52,2589,-1000etc.
3.		Float	Floating type of numbers, doubles, or real numbers	2.5,3.0,58.23,906, 7E-10 etc.
4.		String	Sequence of characters	"php",'php'
5.	Compound	Array	Collection of data values	A[0]=1,A[1]=2.So A is an array having values{1,2}
6.		Object	Concept from Object Oriented Programming	Instance of a class having data members and functions
7.	Special	Resource	Is a reference to an external resource	Files, database connections, image etc.
8.		Null	No value	It represents a variable having nothing

"PHP is a weakly/loosely typed language"

Unlike compiled languages like C,C++,Java variables are not declared and specified with their data types before using them. This implies type of variables can be changed as soon as new values are assigned or manipulated. PHP automatically converts the variable to the correct data type, depending on its value.

Example :

```
<?php

$a = 562;      //integer value assigned.
$a = 25.6;     // float value assigned.
$b = 0123;     // octal value assigned.
$c ="123"      //String value assigned.

?>
```

Is this good or bad??

One of the prime reason of PHP popularity, is the less efforts required on the part of the programmer. Whenever we code in PHP, we need not to define/specify the data type of variable in advance. We can use the variable to store a value of any data type.

Though it provides flexibility to a large extent but it also increases the chances of errors in code. It gives the opportunity to programmers for writing code in a bad style. Directly or indirectly, this convenient approach somehow makes programmers lazy in proper structuring and organisation of data.

TYPE JUGGLING

PHP does not require (or support) explicit type definition in variable declaration; a variable's type is determined by the context in which the variable is used. That is to say, if a string value is assigned to variable $var, $var becomes a string. If an integer value is then assigned to $var, it becomes an integer.

Example :

```
<?php
$v=9;          //Integer value assigned
$f=4;          // Integer value assigned
echo $v.$f;  //Output is 94 due to concatenation operator
echo $v+$f;  //Output is 13 due to arithmetic operator
?>
```

TYPE CASTING

Type casting in PHP works much as it does in C: the name of the desired type is written in parentheses before the variable which is to be cast.

- (int), (integer) - cast to integer
- (bool), (boolean) - cast to boolean
- (float), (double), (real) - cast to float
- (string) - cast to string
- (array) - cast to array
- (object) - cast to object
- (unset) - cast to NULL

```
<?php
$foo = 10;              // $foo is an integer
$fst = (string) $foo; // $fst is also a string
```