PHP - Type Juggling

PHP is known as a dynamically typed language. The type of a variable in PHP changes dynamically. This feature is called "type juggling" in PHP.

In C, C++ and Java, you need to declare the variable and its type before using it in the subsequent code. The variable can take a value that matches with the declared type only.

Explicit type declaration of a variable is neither needed nor supported in PHP. Hence the type of PHP variable is decided by the value assigned to it, and not the other way around. Further, when a variable is assigned a value of different type, its type too changes.

Example 1

Look at the following variable assignment in PHP.

</> Open Compiler

```php
<?php
   $var = "Hello";
   echo "The variable \$var is of " . gettype($var) . " type" .PHP_EOL;

   $var = 10;
   echo "The variable \$var is of " . gettype($var) . " type" .PHP_EOL;

   $var = true;
   echo "The variable \$var is of " . gettype($var) . " type" .PHP_EOL;

   $var = [1,2,3,4];
   echo "The variable \$var is of " . gettype($var) . " type" .PHP_EOL;
?>
```

It will produce the following **output** –

```
The variable $var is of string type
The variable $var is of integer type
The variable $var is of boolean type
The variable $var is of array type
```

You can see the type of "$var" changes dynamically as per the value assigned to it. This feature of PHP is called "type juggling".

Example 2

Type juggling also takes place during calculation of expression. In this example, a string variable containing digits is automatically converted to integer for evaluation of addition expression.

</> Open Compiler

```
<?php
   $var1=100;
   $var2="100";
   $var3=$var1+$var2;
   var_dump($var3);
?>
```

Here is its **output** –

```
int(200)
```


Example 3

If a string starts with digits, trailing non-numeric characters if any, are ignored while performing the calculation. However, PHP parser issues a notice as shown below –

</> Open Compiler

```
<?php
   $var1=100;
   $var2="100 days";
   $var3=$var1+$var2;
   var_dump($var3);
?>
```

You will get the following **output** –

```
int(200)

PHP Warning:  A non-numeric value encountered in /home/cg/root/53040/main.php on li
```

Type Casting vs Type Juggling

Note that "type casting" in PHP is a little different from "type juggling".

- In type juggling, PHP automatically converts types from one to another when necessary. For example, if an integer value is assigned to a variable, it becomes an integer.
- On the other hand, type casting takes place when the user explicitly defines the data type in which they want to cast.

Example

Type casting forces a variable to be used as a certain type. The following script shows an example of different type cast operators –

</> Open Compiler

```php
<?php
   $var1=100;
   $var2=(boolean)$var1;
   $var3=(string)$var1;
   $var4=(array)$var1;
   $var5=(object)$var1;
   var_dump($var2, $var3, $var4, $var5);
?>
```

It will produce the following **output** –

```
bool(true)
string(3) "100"
array(1) {
  [0]=>
  int(100)
```

```
}
object(stdClass)#1 (1) {
  ["scalar"]=>
  int(100)
}
```

Example

Casting a variable to a string can also be done by enclosing in double quoted string −

</> Open Compiler

```php
<?php
   $var1=100.50;
   $var2=(string)$var1;
   $var3="$var1";
   var_dump($var2, $var3);
?>
```

Here, you will get the following **output** −

```
string(5) "100.5"
string(5) "100.5"
```