# PHP – Filtered unserialize()

In PHP, the built-in function unserialize() is available from PHP version 4 onwards. With PHP 7, a provision to pass a list of allowed classes has been added. This allows the untrusted source to be filtered out. The unserialze() function unserializes the data from only the trusted classes.

In PHP, serialization means generation of a storable representation of a value. This is useful for storing or passing PHP values around without losing their type and structure. The built-in serialize() function is used for this purpose.

## serialize(mixed $value): string

The unserialze() function gives a PHP value from the serialized representation. From PHP 7 onwards, the unserialize() function follows the format below −

## unserialize(string $data, array $options = [ ]): mixed

The **$data** parameter is the serialized string which you want to unserialize.

The **$options** parameter has been newly introduced. It is an associative array of following keys −

| Sr.No | Name & Description |
|---|---|
| 1 | **allowed_classes**<br>an array of class names which should be accepted,<br>or<br>false to accept no classes,<br>or<br>true to accept all classes.<br>Omitting this option is the same as defining it as true |
| 2 | **max_depth**<br>The maximum depth of structures permitted during unserialization. |

## Example

Take a look at the following example −

</> Open Compiler

```php
<?php
   class MyClass {
      var int $x;
      function __construct(int $x) {
         $this->x = $x;
      }
   }
   class NewClass {
      var int $y;
      function __construct(int $y) {
         $this->y = $y;
      }
   }

   $obj1 = new MyClass(10);
   $obj2 = new NewClass(20);

   $sob1 = serialize($obj1);
   $sob2 = serialize($obj2);

   // default behaviour that accepts all classes
   // second argument can be ommited.
   // if allowed_classes is passed as false, unserialize converts all objects into
   $usob1 = unserialize($sob1 , ["allowed_classes" => true]);

   // converts all objects into __PHP_Incomplete_Class object except those of MyCl
   $usob2 = unserialize($sob2 , ["allowed_classes" => ["MyClass", "NewClass"]]);

   echo $usob1->x . PHP_EOL;
   echo $usob2->y . PHP_EOL;
?>
```

It will produce the following **output** –

```
10
20
```