**In PHP, what is the \_\_autoload function? Can you provide an example of how it’s used?**

PHP functions that start with a double underscore – a “\_\_” – are called magic functions in PHP. The \_\_autoload function is also a magic function because it has a double underscore in front of it as well. If you want to read a little bit more about magic functions in general, you can go here: [Magic Functions in PHP](http://www.programmerinterview.com/index.php/php-questions/php-what-are-magic-functions/).

## Why is the \_\_autoload function used?

In PHP, the \_\_autoload function is used to simplify the job of the programmer by including classes automatically without the programmer having to add a very large number of include statements. An example will help clarify. Suppose we have the following code:

include "class/class.Foo.php";

include "class/class.AB.php";

include "class/class.XZ.php";

include "class/class.YZ.php";

$foo = new Foo;

$ab = new AB;

$xz = new XZ;

$yz = new YZ;

Note in the code above that we have to include each of the 4 different class files separately – because we are creating an instance of each class, we absolutely must have each class file. Of course, we are assuming that developers are defining only one class per source file – which is good practice when writing object oriented programs, even though you are allowed to have multiple classes in one source file.

## The \_\_autoload function simplifies inclusion of class files in PHP

Imagine if we need to use 20 or even 30 different classes within this one file – writing out each include statement can become a huge pain. And this is exactly the problem that the PHP \_\_autoload function solves – it allows PHP to load the classes for us automatically! So, instead of the code above, we can use the \_\_autoload function as shown below:

function \_\_autoload($class\_name)

{

require\_once “./class/class.”.$class\_name.“.php”;

}

$foo = new Foo;

$ab = new AB;

$xz = new XZ;

$yz = new YZ;

## How does the \_\_autoload function work?

Because the \_\_autoload function is a magic function, it will **not** be called directly by you, the programmer. Instead, it is called behind the scenes by PHP – that’s what makes it magical. But, when does the \_\_autoload function actually get called? Well, in the code above, the \_\_autoload function will be called 4 times, because PHP will not recognize the Foo, AB, XZ, and YZ classes so PHP will make a call to the \_\_autoload function each time it does not recognize a class name.  
  
Also, in the code above, we can see that the autoload function takes the class name as a parameter (the $class\_name variable). PHP passes the $class\_name variable behind the scenes to the \_\_autoload function whenever it finds that it doesn’t recognize the class name that is being used in a given statement. For instance, when PHP sees the “$foo = new Foo;” line, it does not recognize the Foo class because the Foo class was never included or “required” as part of the current file. So, PHP then (behind the scenes) passes the “Foo” class to the \_\_autoload function, and if the class file is found by the autoload function then it is included by the “require\_once” statement.

One last thing worth noticing in the code above is how we use the $class\_name variable in the class.”.$class\_name.“.php”; piece of the code. Basically, this allows us to point to the correct file dynamically. The assumption here is also that the class folder is in the same directory as the current file.

## When is the \_\_autoload function called?

The \_\_autoload function is called anytime a reference to an ***unknown*** class is made in your code. Of course, the \_\_autoload function must be defined by you in order to actually be called.

## Does the \_\_autoload function work with static function calls?

Yes, it does. Remember that a static function is a function that can be called by just using the class name in which it is defined – and there is no need to create an object of the class. The code below, which has a call to a static function, will still run the \_\_autoload function:

function \_\_autoload($class\_name)

{

require\_once “./class/class.”.$class\_name.“.php”;

}

//this is a call to a static function

SampleClass::staticFunctionCall($param);

In the code above, the class SampleClass is not recognized because it is not explicitly included anywhere in the code. This means that PHP will make a call to the \_\_autoload function when it realizes that the SampleClass definition is not provided anywhere. Once the \_\_autoload function is called, the class.SampleClass.php file will be included in order to have the definition of the SampleClass class. Of course, the SampleClass is needed because a call is being made to a static function that belongs to the SampleClass class.

## When else would the \_\_autoload function be called automatically by PHP?

One last thing that is interesting and worth noting is that even calling the **class\_exists**(which just checks to see if a given class is defined) PHP function will call the \_\_autoload function by default. There is an extra parameter in the class\_exists function that would allow you to disable the automatic call to the \_\_autoload function.