Autoloading, Application Skeleton

Disclaimer: вы смотрите просто запись лекции, это **НЕ** специально подготовленный видеокурс!

Autoloading

The idea of autoloading is to provide a mechanism for automated inclusion of files with classes definition on the first attempt to instantiate a class.

No need for multiple include/require call (and no need to maintain the list).

In complex applications PHP saves a lot of time (ne need to parse many files).

The code becomes more clear, compact, maintainable.

Autoloader code

See complete detailed description and code in “PSR-4: Autoloader”: https://www.php-fig.org/psr/psr-4/

```php
<?php

spl_autoload_register(function ($className) {

    // Directory to store classes
    $baseDir = __DIR__ . '/src/';

    // Full fine name
    $fileName = $baseDir . $className . '.php';
    $fileName = str_replace(['\\', '//'], ['/', '/'], $fileName);

    // For debug only!
    echo "This is the class file: [" . $fileName ."]\n";

    // If the file exists, require it
    if (is_file($fileName)) {
        require $fileName;
    }
});

$testObject = new TestClass;
```

Here is a simplified yet usable code.

```
// This is the class file: [D:/PWD_Code_Samples/04 04 -
Autoloading, Application Skeleton/src/TestClass.php]
```

Application skeleton

So, finally, here's a general skeleton for a PHP application/package (see details here: https://github.com/php-pds/skeleton).

If a package has a root-level directory for ...	**... then it MUST be named:**
command-line executables	bin/
configuration files	config/
documentation files	docs/
web server files	public/
other resource files	resources/
PHP source code	src/
test code	tests/

Application skeleton

And now let’s look at a simple application…

(See handouts, “application” folder.)

Autoloading, Application Skeleton

Disclaimer: вы смотрите просто запись лекции, это **НЕ** специально подготовленный видеокурс!