AÇIKLAB PHP EĞİTİMİ

PARDUS İŞLETİM SİSTEMİ ÜZERİNDE PHP GELİŞTİRME

PHP GELİŞTİRME ORTAMININ KURULUMU

- Basit bir php geliştirme ortamı için;
- *sudo apt install php libapache2-mod-php*
- Komutu ile PHP ve Apache ikilisi kurulmuş olur.
- *sudo systemctl restart apache2*
- Komutu ile de apache php modüllerinin yeniden yüklenmesini sağlayabilirsiniz.
- PHP ile geliştirme yaparken herhangi bir editör kullanabilirsiniz.
- Visual Studio Code iyi bir seçim olabilir.

PHP MODÜLLERİNİN KURULUMU

- Php ile geliştirme yaparken bazı modüllere ihtiyacınız olabilir.
- Bu modüllerin sadece paketlerini kurarak hızlı bir şekilde kullanmaya başlayabilirsiniz.
- Bunları kurmak için;
- *sudo apt install php-MODUL_ADI*
- Şeklinde komut çalıştırarak kurabilirsiniz.
- Örneğin;
- *sudo apt install php-mysql php-gd php-json*

PHP GELİŞTİRME ORTAMINI TEST EDELİM

- */var/www/html/*
- Yoluna gidip *index.php* isimli yeni bir dosya oluşturalım.
- Bu dosyanın içine;
- *<?php*
  phpinfo();
- Satırlarını yazalım.
- Sonrasında tarayıcı ile http://localhost adresine gidelim.

PHP SYNTAX ÖRNEKLERİ

Değişken Tipleri

```php
$x = 1;
$y = "foo";
$z = True;
```

Operatörler

```php
$x = 1;
$y = 2;
$sum = $x + $y;
echo $sum;
```

String Formatlama

```php
$name = "Jake";
echo "Your name is $name";
```

Dizi İşlemleri

```php
$odd_numbers = [1,3,5,7,9];
$first_odd_number = $odd_numbers[0];
$second_odd_number = $odd_numbers[1];

echo "The first odd number is $first_odd_number\n";
echo "The second odd number is $second_odd_number\n";
```

```php
$odd_numbers = [1,3,5,7,9];
echo count($odd_numbers);
```

```php
$odd_numbers = [1,3,5,7,9];
$odd_numbers[5] = 11;
print_r($odd_numbers);
```

```php
$odd_numbers = [1,3,5,7,9];
unset($odd_numbers[2]);
print_r($odd_numbers);
```

```php
$phone_numbers = [
  "Alex" => "415-235-8573",
  "Jessica" => "415-492-4856",
];

$phone_numbers["Michael"] = "415-955-3857";

print_r($phone_numbers);
```

PHP SYNTAX ÖRNEKLERİ

For Döngüsü

```php
$odd_numbers = [1,3,5,7,9];
for ($i = 0; $i < count($odd_numbers); $i=$i+1) {
    $odd_number = $odd_numbers[$i];
    echo $odd_number . "\n";
}
```

While Döngüsü

```php
$counter = 0;

while ($counter < 10) {
    $counter += 1;
    echo "Executing - counter is $counter.\n";
}
```

Foreach

```php
$phone_numbers = [
  "Alex" => "415-235-8573",
  "Jessica" => "415-492-4856",
];

foreach ($phone_numbers as $name => $number) {
  echo "$name's number is $number.\n";
}
```

PHP OOP ÖRNEKLERİ

- Sınıf Tanımlama

```php
<?php
class BasitSınıf
{
    // özellik bildirimi
    public $öntanımlı = 'öntanımlı bir değer';

    // yöntem tanımı
    public function ÖntanımlıyıGöster() {
        echo $this->öntanımlı;
    }
}
?>
```

```php
<?php
$örnek = new BasitSınıf();

// Bu bir değişkenle de yapılabilir:
$sınıf = 'BasitSınıf';
$örnek = new $sınıf(); // new BasitSınıf()
?>
```

PHP OOP ÖRNEKLERİ

- Özellik Tanımlama

```php
<?php

class User
{
    public int $id;
    public ?string $name;

    public function __construct(int $id, ?string $name)
    {
        $this->id = $id;
        $this->name = $name;
    }
}

$user = new User(1234, null);

var_dump($user->id);
var_dump($user->name);

?>
```

PHP OOP ÖRNEKLERİ

- Kalıtım

```php
class Foo
{
    public function printItem($string)
    {
        echo 'Foo: ' . $string . PHP_EOL;
    }

    public function printPHP()
    {
        echo 'PHP is great.' . PHP_EOL;
    }
}
```

```php
class Bar extends Foo
{
    public function printItem($string)
    {
        echo 'Bar: ' . $string . PHP_EOL;
    }
}
```

```php
$foo = new Foo();
$bar = new Bar();
$foo->printItem('baz'); // Çıktısı: 'Foo: baz'
$foo->printPHP();       // Çıktısı: 'PHP is great'
$bar->printItem('baz'); // Çıktısı: 'Bar: baz'
$bar->printPHP();       // Çıktısı: 'PHP is great'
```

PHP OOP ÖRNEKLERİ

- Diğer tüm örnekler için resmi php dökümanını inceleyebilirsiniz.
- https://www.php.net/manual/tr/oop5.intro.php

COMPOSER

- Composer php için geliştirilmiş bir paket yönetim sistemidir.
- Projelerde ihtiyaç halinde kütüphaneler kullanılabilir.
- Bu kütüphanelere projenin bağımlılıkları denir.
- Composer projenin bağımlılıklarını composer.json dosyasında tutar.
- Composer aynı zamanda auto-load yapısı sunar ve farklı dosyalarda yazdığınız php sınıflarını uğraşmadan projenizde kullanmanıza olanak sağlar.

COMPOSER KURULUMU

- https://getcomposer.org/download/ adresinden kurulum yapılabilir.
- *php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');*
- Komutu ile kurulum betiği indirilir.
- *php composer-setup.php --install-dir=/usr/local/bin --filename=composer*
- Komutu ile de kurulum global olarak tamamlanır.

COMPOSER INIT

- Composer ile yeni bir proje oluşturmak için öncelikle bir klasör oluşturulur ve klasörün içerisinde terminal açılır.
- *composer init*
- Komutu ile composer.json dosyası otomatik olarak oluşturulur.
- Bu komut aynı zamanda vendor klasörünü de oluşturmuş olur.
- Projenizin ana klasöründe *index.php* adında bir dosya oluşturup ilk satırına;
- *require __DIR__ . '/vendor/autoload.php';*
- Bu kodu yazarak otomatik üretilmiş olan autoload.php dosyasını php betiğinize eklemiş olursunuz.

COMPOSER REQUIRE

- Projenize bir kütüphane bağımlılığı eklemek için
- *composer require PAKET_ADI*
- Komutunu kullanabilirsiniz.
- Örneğin: *composer require mervick/aes-everywhere*
- Composer paketlerini keşfetmek için Github'ı ve Packagist'i kullanabilirsiniz.
- https://packagist.org/

COMPOSER INSTALL

- Eğer bir projedeki mevcut bağımlılıkları kurmak isterseniz:
- *composer install*
- Komutunu çalıştırabilirsiniz.
- NOT: Composer'ın paketleri indirdiği *vendor* klasörü genelde Git gibi versiyonlama sistemlerine pushlanmaz. Github'tan bir proje indirdiğinizde *composer install* komutu ile önce bağımlılıklarını kurmalısınız.

COMPOSER AUTOLOAD

- Composer'ın class autoload yapısını kullanmak için projenizde bir klasör oluşturmalısınız.
- Standartlara uyması açısından bu klasörün adı *src* veya *app* olabilir.
- Daha sonrasında örneğin *app* klasörü için *composer.json* dosyanızda sağ taraftaki düzenlemeleri yapmalısınız:
- Ardından;
- *composer dump-autoload*
- *Komutunu çalıştırıp auto-load'un tetiklenmesini sağlayabilirsiniz.*

```
"autoload": {
        "psr-4": {
                "App\\": "app"
        },
        "classmap": [
                "app"
        ]
}
```

ÖRNEK COMPOSER.JSON DOSYASI

- https://github.com/limanmys/liman-ansible/blob/master/composer.json

```
{
        "name": "liman/ansible",
        "type": "project",
        "authors": [
                {
                        "name": "Mustafa AKBEL",
                        "email": "mustafaakbell@gmail.com"
                }
        ],
        "require": {
                "illuminate/events": "^8.0",
                "doctrine/dbal": "^2.10",
                "illuminate/validation": "^8.0",
                "illuminate/translation": "^8.0",
                "illuminate/filesystem": "^8.0",
                "vlucas/phpdotenv": "^5.1",
                "illuminate/pagination": "^8.5"
        },
        "autoload": {
                "files": [
                        "app/Helpers/Helpers.php"
                ],
                "psr-4": {
                        "App\\": "app"
                },
                "classmap": [
                        "app"
                ]
        }
}
```