Bestanden en Mappen

Voorkennis en uitwerking

Deze opgave gaat enerzijds over het werken met klassen en objecten in PHP, en anderzijds over het werken met bestanden en mappen in PHP. Voorts raken we ook het uploaden van bestanden en de superglobal $_FILES aan. Gebruik SPL zo veel als mogelijk.

Voor het uitwerken van de labo's maken we gebruik van MAMP of Wampserver. Het is in MAMP en Wampserver niet nodig om de nodige schrijfpermissies in te stellen.

De opdrachten hoeven niet per se in volgorde opgelost worden. Het huishoudelijk reglement is zoals steeds van toepassing.

Timing en upload

Dit labo loopt over één sessie (3u) en dient aan het einde van dit labo via Toledo geüpload te worden. Thuis werk je het labo verder af. Upload één .zip bestand met daarin je .php bestanden.

Doelstellingen

- Enkele ingebouwde PHP klassen kunnen gebruiken
- Bestanden en mappen op schijf kunnen manipuleren
- Bestanden kunnen uploaden via PHP

Op schijf

- De locatie waar dit project opgeslagen wordt is <webroot>/voornaam.achternaam/labo03
- Sla elk bestand als opgave_x.php op.

Opgaves

1. In labo01 werkten we een PHP-script uit dat met behulp van de date() functie een datum omzette in verschillende weergaves. Pas het aangeleverde script opgave_1.php aan zodanig dat de in PHP ingebakken DateTime klasse gebruikt wordt. Info over deze klasse vind je op http://php.net/manual/en/class.datetime.php .
Hoe kunnen we deze keer een gepaste foutboodschap geven in het geval van een niet-interpreteerbare datum?

2. Werk het aangeleverde script opgave_2.php verder uit. Het is de bedoeling dat dit script een reeks .jpg bestanden die op schijf staan inleest en vervolgens op het scherm weergeeft (<img>). Gebruik de aangeleverde bestanden uit de images map. De bestanden overloop je met de SPL DirectoryIterator klasse.

Structuur. Merk op dat we de typische codestructuur met een PHP-blok bovenaan en een HTML-blok onderaan (met hier en daar wat PHP enkel voor weergave) wensen te behouden. Om dit te verwezenlijken sla je de (namen van de) ingelezen bestanden op in een array in het HTML-blok. In het PHP-blok doorloop je vervolgens die array en output je één voor één de gestockeerde namen.

Vermijd absolute paden. Let op het feit dat je script eender waar moet kunnen werken: indien je het script en de afbeeldingen verplaatst van map (of van server), dan moet het script blijven werken. Gebruik daarom een **magische constante** om het huidige pad op te halen.

Over de directory separator. Bemerk verder dat je tijdens het inlezen met paden op schijf zal werken (viz. `/Applications/MAMP/htdocs/...` `C:\wamp64\www\...`), maar dat je bij weergave met paden relatief t.o.v. http://localhost/ zal moeten werken. Paden op schijf gebruiken de DIRECTORY_SEPARATOR als scheiding tussen mappen. In URLs wordt echter steeds – los van het besturingssysteem – de / gebruikt.

3. Breid de vorige opgave uit zodat je uit het bijgevoegde tekstbestand captions.txt de omschrijving van de afbeelding kan ophalen en weergeven. Elke regel van het bestand vormt de omschrijving van een afbeelding (regel 1 = omschrijving 1.jpg, regel 2 = omschrijving 2.jpg, regel 3 = ..). Gebruik een SplFileObject om de regels van het bestand te overlopen.

Gebruik geen aparte lus om de captions uit het bestand te halen. Terwijl je de bestanden uit de images-map overloopt, lees je en verplaats je manueel de pointer van het captions.txt

bestand, m.b.v. de methoden current() en next().

Plaats je afbeelding-links en captions in één grote array. Het is gebruikelijk in PHP om arrays te nesten en data-labels gebruiken:
array(0 => array('url' => … , 'caption' => …), 1 => array('url' => … , 'caption' => …), …

4. Maak een (aparte) PHP-pagina waarmee je de set afbeeldingen van de eerste twee opgaven kan uitbreiden door **nieuwe bestanden** (om het even welke naam, maar enkel de extensie .jpg) **te uploaden**. Vraag eveneens om een omschrijving. Deze omschrijving komt natuurlijk ook terecht onderaan in het tekstbestand captions.txt . Voorzie de nodige controles qua extensie/omschrijving en zorg ervoor dat de nummering van de bestanden gevolgd wordt.

5. Maak een simpele filebrowser. Het script lijst alle children van de huidige map op.

- Indien het een bestand betreft toon je naast de bestandsnaam ook de grootte van het bestand. Link de bestandsnaam door naar het bestand an sich (de webserver zal het dan ter download aanbieden).

- Indien het een map betreft kan je doorklikken en wordt het subpath via een GET parameter aan de URL gekoppeld. Uiteraard worden vervolgens de bestanden van die submap opgelijst. Voorzie eveneens een link terug naar de bovenliggende map.
  VoorbeeldURL: http://localhost/vn.an/labo03/opgave_5.php?path=images%2Ficons

Voorzie een controle zodat een path parameter waar ../ in voorkomt niet toegelaten wordt. Indien je dit niet zou doen zou dit immers een groot beveiligingslek met zich meebrengen!

Gebruik eventueel de bijgevoegde HTML template als basis om van te vertrekken. Een eigen (simpele) layout maken mag ook uiteraard.

Uitbreiding 1. Voorzie iconen op basis van de extensie. Indien er geen icoon voorzien is, gebruik dan default.gif als afbeelding. Gebruik folder.gif om mappen aan te duiden.

Uitbreiding 2. Breid de opgave verder uit zodat je bestanden en mappen kan toevoegen/bewerken/verwijderen.