

Scope & partials

Variable scope

När vi börjar använda funktioner och flera separata filer upptäcker vi att allting är inte tillgängligt hela tiden. Ta det här exemplet, vad kommer att skrivas ut?

```
$a = "Hello!";

function say_hello(){
    echo $a;
}

say_hello();
```

Ingenting, varje funktion skapar sitt eget **scope**. Det som händer i funktionen stannar i funktionen. Vi kan inte heller komma åt värden som är utanför funktionen om vi inte skickar med dem. Inuti funktionen har vi **Function scope** och utanför, i själva **.php**-filen har vi **global scope**. Dessa kan inte prata med varandra hur som helst. Det är därför funktioner tar emot argument och returnerar nya värden.

```
$a = "Hello!";

function say_hello($argument){
    echo $argument;
}

/* We are sending $a to the function and echo it */
say_hello($a); // "Hello!"
```

Alternativt måste vi säga att vi ska använda den **globala** variabeln **\$a** genom nyckelordet **global**:

```
$a = "Hello!";

function say_hello()
{
    /* "Use the global defined $a" */
    global $a;
    echo $a;
}

say_hello();
```

Eller använda superglobalen **\$GLOBALS** som funger som koden ovan. (**\$_POST** och **\$_GET** är också superglobaler som är tillgängliga överallt)

```
$a = "Hello!";

function say_hello()
{
    /* "Use the global defined $a" */
    echo $GLOBALS['a'];
}

say_hello();
```

Allting som inte är inuti en funktion är i global scope.

File scope

Om värden är skapade i två olika filer kan de inte heller prata med varandra. Om vi har två filer som nedan så är bara variabeln `$a` tillgänglig i `index.php` och inte i `form.php`. Variablerna är bara globala i den filen man är för tillfället.

```
//index.php
<!DOCTYPE html>
<html>
<head>
  <title>index.php</title>
</head>
<body>
  <?php $a="Hello!"; ?>
</body>
</html>
```

```
//form.php
<?php
echo $a; //this won't work, separate file.
```

För att "kombinera" olika filer och inkludera samma kod i flera olika filer behöver vi `include` & `require`.

`include` & `require` - skapa partials

Vi kan dela upp våra `.php`-filer i mindre delar om samma kod ska användas på olika sidor. **Observera att filerna fortfarande INTE kan "prata" med varandra utan att vi i princip klistrar in samma kod på flera olika sidor**

Tänk t.ex. att vi ska ha `<head>` på varje enskild sida. Den ska alltid se likadan ut. Därför kan vi göra den till en **partial**, en delfil:

```
//head.php
<html>
<head>
  <title>coolsite</title>
</head>
```

```
//index.php
require 'head.php';
<body>
  <h1> Yo, where is my head? Ah, in "head.php" file</h1>
  <p>I still work as before :)</p>
</body>
</html>
```

```
//contact.php
require 'head.php';
<body>
  <h1> I am a different file, still the same head </h1>
</body>
</html>
```

Vi kan även enbart ha `php` i filerna och inkludera dessa. Detta kommer att resultera i att variabeln från `username.php` finns tillgänglig i `index.php`. Vi kan använda `include` och `require` varsomhelst på en `php`-sida.

```
//username.php
$username = "one_cool";
```

```
//index.php
<html>
<head>
  <title>include stuff </title>
</head>
<body>
```

```
<?php
    include 'username.php';
    echo $username;
?>
</body>
</html>
```










Övning

I grupper om 3

Du ska omstrukturera en färdig sida så att den, efter du är klar, är uppdelad i olika `.php`-filer. De delar som används på flera sidor och som ska se likadana ut på samtliga sidor ska ligga i olika `partials`.

Våra sidor som `index.php`, `about.php` och `contact.php` kan vi referera till som `views`, det vi ser utåt som användare. Våra `views` är ihopsatta av flera av våra `partials`. En del av våra `views` är likadana på varje sida, medan visst innehåll ändras beroende på vilken sida vi är på tillfället.

1 . Ladda ner filen `base_site.zip` från **zenit**. I den ligger två mappar. Mappen `base` är de filerna ni ska omvandla. Sidan ska se exakt likada ut i webbläsaren när ni är klara, det är bara filerna som ska delas upp. 2 . Ni ska använda `include` och `require` för att återanvända vissa delar av sida och dela upp er mapp i liknande struktur:

-  `base`
 -  `partials`
 -  `head.php`
 -  `nav.php`
 -  `header.php`
 -  `footer.php`
 -  `about.php`
 -  `contact.php`
 -  `index.php`

3 . När ni är klara så jämför först med en annan grupp och jämför om ni har kommit fram till samma lösning.

4 . Sitt kvar i grupperna och säg till mig så kollar vi igenom lösningen tillsammans.

5 . Till sist kan ni kolla på lösningsförlaget om ni vill.