

SCOPE

+ include & require

HUR MAN SÖKER EFTER INFORMATION

STÄLLA EN FRÅGA TILL EN MÄNNISKA

om kod

- Det här vill jag göra
- Det här har jag gjort
- Det här var det som faktiskt hände

Förväntat resultat & Faktiskt resultat

Utgå alltid ifrån att personen inte har en aning om
vad du gör

Även om programmering är alltid densamma så
förändras **kontexten**

STÄLLA EN FRÅGA TILL EN MASKIN

Maskinerna är mycket bättre på att gissa vad du gjort, vad du vill göra och vad som har hänt

Maskinerna är starkare, smartare, bättre

Tillbe våra nya ledare, maskinerna

Hur man söker information

Google

Stack Overflow

Hitta information som är generisk: samma för alla

Ta bort information som gäller bara dig

Begränsa tidsspannet

Läs svaret, kopiera inte bara koden

Scope

För det mesta är vi i:

Global Scope

Detta menas **globalt** för hela filen. INTE alla filer på hela hemsidan

PHP är **stateless**: vet inte vad som hände på förra sidan

Hela filen utvärderas innan vi ser sidan.

```
<? $name = "Jesper"; ?>  
<div>  
    <?= $name ?>  
</div>
```

Olika **php**-taggar, samma **scope**

```
$a = "Hello!";  
  
function say_hello(){  
    echo $a;  
}  
  
say_hello();
```

Vad kommer att skrivas ut?

När vi skapar en function får vi även

`Function scope`

Funktionen kan inte komma åt variabler utanför

Utanför funktionen kan vi inte komma åt variabler
inuti funktionen

Vi måste i så fall säga att vi ska använda den globala variabeln

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

Eller använd superglobala variabeln: **\$GLOBALS**

```
$a = "Hello!";

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

say_hello();
```


Detta är en **Bra Grej**

Bara den kod som MÅSTE ha tillgång till variabel ska ha det.

Det är också därför vi ska använda **return**

Global scope

```
<?php
$products = array("A thing", "anotha", "one more!");
foreach($products as $product){ ?>

    <p> <?= $product ?> </p>

<?php } ?>
```

Stänger taggen på tredje raden, men loopen stängs inte förän sista raden. Loopen fortsätter

Vanligt mönster: vi har en alternativ syntax

```
<?php
$products = array("A thing", "anotha", "one more!");
foreach($products as $product):
?>

    <p> <?= $product ?> </p>

<?php endforeach; ?>
```

```
<?php
```

```
$age = 18;
```

```
if($age > 18):
```

```
?>
```

```
    <p> Välkommen! </p>
```

```
<?php endif; ?>
```

Båda sätten fungerar nästan likadant

`endif;` och `endforeach;` brukar ge lite snyggare
och tydligare kod

<http://php.net/manual/en/control-structures.alternative-syntax.php>

Whoops 🤡

Det ska egentligen vara `elseif`

INCLUDE

/

REQUIRE

Med `include` & `require` så kopierar vi kod till flera ställen

Vi länkar **INTE** samma kod till olika ställen


```
//code.php
```

```
<?php
```

```
    $name = "Jesper";
```

```
//index.php
```

```
<?php
```

```
    include 'code.php';
```

```
    echo $name;
```

`include` ger oss en `varning` om det är fel på filen
vi inkluderar

`require` ger oss ett `error` om det är fel på filen vi
inkluderar

```
//index.php
<?php
    include 'code.php';
    echo $name;
```

```
//index.php
<?php
    require 'code.php';
    echo $name;
```

Funkar på både html och php

KODEXEMPEL