



RECAP-ISH

Allting utanför `<?php ?>` tolkas som **HTML**

Allting innanför `<?php ?>` tolkas som **PHP**

```
<? ?> == <?php ?>
```

Varning: detta kan vara avstängt på många servrar -
`short_open_tag`

Single & double quotes matter

```
<?php
$first = "Jesper";
$last = "Orb";
$full = "$first $last";           //It works
$fullWithDouble = "{$first} {$last}"; //It works
$fullWithSingle = '{$first} {$last}'; //Does not work
```

CONSTANTS

```
<?php  
define( 'PI', 3.14159265359 );
```

Konstanter borde ha stora bokstäver

KEY/VALUE

```
<?php
$assoc = array(
    "key" => "value"
);
```

```
<?php
$person = array(
    "name" => "Jesper",
    "age" => "Infinite"
);
```

Separeras med **=>** istället för **:**

FOREACH ASSOCIATIVE

```
<?php
$panda = array(
    "name" => "Panda",
    "weight" => "3000"
);
foreach($panda as $key => $value){
    echo $key . $value;
}
```

```
<?php
$foo = 42;
function bar(){
    echo $foo;    //NOOO
}
```

Variabler finns enbart i det scope de är deklarerade i.


```
<?php
$foo = 42;
function bar($foo) {
    echo $foo;
}
```

Alltid en bättre idé att skicka vidare variabler.

Vi kan skicka med fler argument än parametrar

Vi ska skicka med lika många argument som vi har parametrar

Alla funktioner ska egentligen skrivas i `camel_case`

Förutom metoder som är `camelCase`

CLOSURE/ANONYMOUS FUNCTIONS

```
$name = 'Jesper';  
  
$say = function ($name) {  
    echo $name;  
};  
  
$say($name);
```

Får INTE variabler från det yttre scopet som i JS
Funktioner inuti funktioner är inte lika vanligt i PHP

Typ closure

```
$name = 'Jesper';  
  
$say = function () use ($name) {  
    echo $name;  
};  
  
$say();
```

USE CASE

```
//usort is like Array.prototype.sort()  
$array = usort($array, 'comparator');  
  
function comparator() {  
    return a - b;  
}
```

Callback function

CODING CONVENTIONS

- PHP The Right Way
- PHP Best Practices
- PHP-FIG

INCLUDE / REQUIRE

Vi har bara **function scope**

Alla våra sidor tolkas inte under samma **global scope**

Vi måste inkludera eller kräva filer

code.php

```
<?php  
$name = "Jesper";
```

index.php

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

Sökvägarna borde utgå från **root**: **/**

Generell vill man gå uppåt från root och inte backa som:

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

MAMP/WAMP/XAMPP borde vara inställt på detta från början
men **shit is hard** så man vet aldrig.

Absoluta sökvägar är generellt inte bra

Vad servern ska köras kan ändras och därmed slutar de att fungera

```
file:///Users/jesperorb/php/css/main.css
```

```
css/main.css
```

*require is identical to include except upon failure it will also produce a fatal **E_COMPILE_ERROR** level error. In other words, it will halt the script whereas include only emits a warning **(E_WARNING)** which allows the script to continue.*

REQUIRE_ONCE / INCLUDE_ONCE

Säkerställer att koden enbart inkluderas en gång.

```
include_once 'code.php';
```

Om man inkluderar samma kod flera gånger får man error
`include_once` kollar först upp så att filen inte har inkluderats
redan

DAGENS ÖVNING

- Strukturerar upp en hemsida
- Jämför med en annan grupp i klassen
- Pusha upp er lösning till GitHub

NOBEL PRIZE LIST