FUNCTIONS

Om ingenting funkar

Kolla semikolon

Kolla "skumma" mellanslag

```
echo "Hej" . $name . "!";
```

Kolla så att alla strängar är rätt

Computer Facts
@computerfact

Följer

restarting a computer fixes problems because its soul dies and is replaced by a new one which might have different opinions about your work

Översätt från engelska

16:41 - 9 sep. 2016

PHP standard code style

http://www.php-fig.org/psr/psr-1/

http://www.php-fig.org/psr/psr-2/

Googla på PSR-1 och PSR-2

```php
$numbers = array(5,6,2);
$sum = 0;
for($i = 0; $i < count($numbers); i++){
  $sum += $i;
}
echo $sum;
```

`count()` är en inbyggd funktion

Flertal problem

Koden körs direkt hela tiden

Koden körs en gång

Vi har ingen kontroll

Vi upprepar oss

Vi kan inte enkelt kalla på samma kod flera gånger med olika värden

DRY

DON'T REPEAT YOURSELF

Programmerare ska vara lata

Lathet

Skapa en första lösning som FUNKAR

Hitta grundproblemet och lös det

Sen, om det behövs, **gör bättre**

Refaktorera

Omstrukturering av kod

Tanken är att förbättra, utan att ändra grundfunktionaliteten

Få den mer lättläst

Minska errorrisken

Inkapsling

Eng: encapsulation

Att plocka ut kod som hör samman

Ge den koden ett namn

Allting handlar om struktur

I ett dokument har vi **headings** som ger oss en överblick t.ex.

FUNCTIONS

En funktion kapslar in och gör ett kodblock återanvändbart

En funktion kan kallas på när vi vill

En funktion kan kallas på flera gånger med olika värden

```
function say_hello(){
  echo "Hello";
}
```

En funktion start med nyckelordet `function`

Följt av namnet som du ger funktionen

Följt av paranteser

Följt av curly brackets där vår kod ska vara

En funktion körs inte förän du kallar på den

```php
function say_hello(){
  echo "Hello";
}
```

```php
say_hello(); //echo "Hello"
```

Där du kallar på funktionen skrivs koden ut

Inte där du skrev koden

ARGUMENT & PARAMETRAR

Vi vill ofta kalla på samma kod fast med olika värden

Perfekt tillfället för funktioner

Vi kan även skicka med värden till en funktion

Inom funktionen: parameter

```php
function say_hello($name){
  echo "Hello $name";
}
```

När vi skickar med värdet: argument

```php
say_hello("Jesper");
```

`"Jesper"` lagras i `$name`

Värdena vi skickar med när vi kallar på funktionen

(argument)

Lagras som variabler (parametrar) inuti funktionen

Variabeln finns enbart inuti funktionen

Vi kan i princip ha hur många argument som helst, håll koll på ordningen dock.

```php
function say_hello($name1, $name2){
  echo "Hello $name1 and $name2";
}
```

```php
say_hello("Jesper", "Grobb");
```

RETURN

Oftast vill vi spara värdet från en funktion istället för att skriva ut det

Med `echo` skrivs det ut, men sparas inte i koden

Med `return` säger vi att vi ska fortsätta använda värdet som skapas

```php
function say_hello($name){
  return "Hello $name";
}
```

Vi måste spara värdet som returneras, annars försvinner det

```php
say_hello("Jesper"); //Does "nothing"
```

```php
echo say_hello("Jesper");
```

```php
$hello = say_hello("Jesper");
echo $hello;
```

Mer värt exempel

```php
function add_two_numbers($a, $b){
  $sum = $a + $b;
  return $sum;
}
```

```php
$sum = add_two_numbers(5,6);
echo $sum * 5; //we can continue using the value
```

```php
$new_sum = add_two_numbers($sum, $sum);
echo $new_sum;
```

ÖVNINGAR PÅ ZENIT

Samt den första individuella