

FORMS

& Associative arrays

LOOPAR

```
for($i = 0; $i < 10; $i++){  
    echo $i;  
}
```

Ser nästan alltid ut likadan oavsett språk

```
while(true){  
    echo "What?";  
}
```

Körs så länge **condition** stämmer

```
do{  
    echo "What?";  
} while(false);
```

Körs en gång även om **condition** inte stämmer

```
foreach($numbers as $number){  
    echo $number;  
}
```

Specifik för PHP, finns varianter i andra språk

start

stopp

ökning

Har vi en array, måste vi ofta använda en loop

Vi kan kombinera hur som helst

```
for($i = 0; $i < 10; $i++){  
  for($j = 0; $j < 10; $j++){  
    echo $i;  
    echo $j;  
  }  
}
```

while i for-loop, for i while? If i for? if i if? for i else?

ARRAYS

Indexerad array: automatisk nyckel

```
$numbers = array(5, 23, 4);
```

```
echo $numbers[0]; //5
```

```
echo $numbers[1]; //23
```

```
echo $numbers[2]; //4
```

key: value

Som i css ~ property: value

Associative Array

Nycklarna behöver inte vara **index**

```
$info = array("username" => "zero_cool");
```

Istället för **[0]** ger vi platsen ett namn

```
echo $info["username"]; //zero_cool
```

```
$numbers = array(5,6);
```

```
$numbers = array("first" => 5, "second" => 6);
```

Blanda inte ihop, använd antingen enbart index, eller
använd enbart namn på nycklarna

```
key => value
```

Det finns massvis av hjälpfunktioner för arrayer

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

```
$numbers = array(5,6,2);  
$sum = array_sum($numbers);
```

FORMS

Alla formulär i HTML fungerar by default med PHP
Formulär är inställda på att skicka iväg innehållet till
en **php-fil**

Vi behöver dock berätta **var** & **hur** informationen
ska skickas

```
<form action="form.php" method="POST">  
  <input type="text" name="username" />  
  <input type="submit" value="Skicka" />  
</form>
```

Varje formulär vill ha en **action** och **method**

Samt **<input>** för varje värde som ska skickas

`action` är viktigt

`method` är viktigt

`name` på inputfält är viktigt

Har vi detta kan vi lagra informationen på rätt sätt

SUPERGLOBALS

`$_POST` är en såkallad superglobal

Informationen du skickar med formuläret lagras

automatiskt i `$_POST`

Men i den filen du skickar informationen till

Vi kan även skicka med metoden GET

Skillnaden är att informationen lagras i \$_GET

```
<form action="form.php" method="GET">
  <input type="text" name="username" />
  <input type="submit" value="Skicka" />
</form>
```

```
echo $_GET[ "username" ];
```

ÖVNINGAR UPPE PÅ ZENIT

Finns extraövningar gällande loopar

Inlämningsuppgiften kommer upp imorgon

Kolla närmare på Arrayfunktioner & datumfunktioner

Imorgon kommer vi in på vad en funktion egentligen
är