

PDO

PHP Data Object

PREPARE / EXECUTE

**KOLLA ALLA "STRÄNGAR"
OCH SEMIKOLON;**

KOLLA RÄTT

TABELLNAMN OCH

DATABASNAMN

PDO

```
$pdo = new PDO($dsn, $user, $pass, $options);
```

```
$pdo = new PDO(  
    "mysql:host=localhost;dbname=products;charset=utf8",  
    "root",  
    "root",  
    $options  
);
```

that's so fetch!

```
$statement = $pdo->prepare("SELECT * FROM pc");  
$statement->execute();  
$data = $statement->fetchAll(PDO::FETCH_ASSOC);  
foreach ($data as $row){  
    echo "<li>" . $row['model'] . "</li>";  
}
```

\$_SESSION

SUPERGLOBAL

\$_SESSION



SUPERGLOBALS

- \$GLOBALS
- \$_SERVER
- \$_GET
- \$_POST
- \$_FILES
- \$_COOKIE
- \$_SESSION
- \$_REQUEST
- \$_ENV

`$_GET` för att få URL-query

```
https://mysite.com?name=jesper&ok=whatever
```

```
var_dump($_GET) // { name: jesper, ok: whatever }
```

`$_POST` för att hämta data från forms

```
<form action="post.php" method="POST">
  <input type="text" name="name">
  <input type="text" name="ok">
  <input type="submit">
</form>
```

```
var_dump($_POST) // { name: jesper, ok: whatev }
```

form action, form method och name är viktiga

`$_SESSION`

I sessionsvariabeln kan vi lagra information som ska vara under en längre tid

En session håller data när du går till en annan sida.

Lagras på servern och i browsern men inte i databasen.

```
$_SESSION[ 'name' ] = "Jesper";
```

Varje gång du ska köra en session måste du se till så att sessionen är startad

```
session_start();
```

```
session_destroy();
```

En session kan t.ex. vara **en inloggning**

Vi ska inte starta sessionen om den redan är startad.

```
if (session_status() == PHP_SESSION_NONE) {  
    session_start();  
}
```

Lägg `session_start()` längst upp i dokumentet

LOGIN / LOGOUT

`$_SESSION` används främst i kombination med login / logout
men man kan lagra vad som helst i arrayen

Vi vet hur vi kan lagra en användare/vad som helst i SQL

```
INSERT INTO users (username, password)  
VALUES (zero_cool, strongpass2000)
```

Vad är felet?

```
INSERT INTO users (username, password)
VALUES (zero_cool, strongpass2000)
```

Tips: säkerhet

Vi måste kryptera våra lösenord
Det är inbyggt

```
password_hash("strongpass2000", PASSWORD_DEFAULT)
```

Men vi måste kunna "okryptera" vårt lösenord

Det är inbyggt

```
$hash = password_hash("strongpass2000", PASSWORD_DEFAULT)
```

```
if (password_verify('strongpass2000', $hash)) {  
    echo "Yes! It's YOU!";  
} else {  
    echo "Liar";  
}
```

IMPROVEMENTS!



Allting kan delas upp i funktioner och klasser.

FIRST STEP!

Includes! requires!

```
include 'Database.php';  
require 'myCoolFunctions.php';
```

Lägg logiken i separata filer

```
function print_pc($data){  
    $list = "<ul>";  
    foreach ($data as $row){  
        $list .= "<li>" . $row['model'] . "</li>";  
    }  
    $list .= "</ul>";  
    return $list;  
}
```

```
$data = $statement->fetchAll(PDO::FETCH_ASSOC);  
echo print_pc();
```


CLASSES

```
class Animal
{
    private $name;
    public function __construct($name)
    {
        $this->name = $name;
    }
}
```

```
class Dog extends Animal
{
    //I am almost the same as Animal!
}
```



I Am Devloper

@iamdevloper

Följer



manager: we need to design an admin system
for a veterinary centre

dev: ok, this is it, remember your training

```
class Dog extends Animal {}
```

Ni behöver kanske inte arv men ni behöver strukturera enligt klasser.

Allting som du kan göra som enskild funktion eller proceduell kod kan läggas i en klass.

Allting som berör användare kan få en egen klass

```
class Users
{
    public function login(){}
}
```