

Formulär, `$_GET` & `$_POST`

Om du är lat som din lärare och inte orkar med att styla `input`-fält så använder du t.ex. [Bootstrap 4 - Forms](#) för att styla och positionera dina formulär.

Två bra länkar om du inte har helt koll på alla typer av `input`:

- [HTML5 Input types @ HTML5 Doctor](#)
- `<input>` @ [MDN](#)

`<form>`

Varje formulär på varje hemsida har alltid en `action` samt en `method`.

När vi klickar på `submit`-knappen så kommer vi att skickas till den sidan vi har skrivit in i `action` samt värdena i våra `input`-fält skickas med till den sidan.

```
<!--
-   `action` is the file we are sending the info to
-   `method` is telling the form to save the values
-   to the $_POST array
-->
<form action="form.php" method="POST">
  <!-- ALL INPUTS MUST HAVE A NAME TO BE STORED -->
  <input type="text" name="username" />
  <input type="submit" value="Skicka" />
</form>
```

```
//form.php
<?php

/*
 * Arrays kan either have index as keys: 0,1,2,3,4
 * or they can have names as keys, $_POST have names as keys.
 * DO NOT MIX
 */

echo $_POST["username"]; /* value from input */

/*
 * If there is only php in a file and no HTML like this on, we can omit the end
 * tag if we want, or you can choose to leave it hanging like here.
 */
?>
```

Ibland kan vi för att undvika error behöva kolla om värdet finns eller inte. Detta gör vi oftast med `isset()`-funktionen:

```
/* If the value is set, exists, echo it */
if(isset($_POST["username"])){
    echo $_POST["username"];
}
```

Övningar

1. Skapa ett formulär i din `index.php`. I formuläret ska man kunna fylla i information i texttrutor. Närmare bestämt: *användarnamn*, *födelsedatum* och *favorit css-egenskap*. Det ska också finnas en `submit`-knapp. Skapa även en `php`-fil som heter `form.php` som ska ta emot denna information. Gör som exemplet ovan och få det att fungera så att du kan skicka informationen till `form.php` och att det skrivs ut på sidan i stil med: *"Hej användarnamn, du är född den 1975-03-22. Din favoritegenskap är box-shadow. Nice."*

2.

Lägg även till en extra input som kollar om användaren har skrivit in rätt lösenord. Om fel lösen skrivs in så ska denna bild visas: <https://media.giphy.com/media/3ohzdQ1lynzclJldUQ/giphy.gif>. Annars ska all information visas upp som tidigare.

3.

Lägg till en till koll om både lösenordet är rätt samt om användaren är minst 25 år så får användaren se sidan. Annars ska bilden visas. Här kan man t.ex. använda hjälpfunktionen `strtotime()`

4.

Testa att ändra metoden på något av dina formulär till `GET`. Vad händer? Ser du någon skillnad någonstans? Varför/Varför inte ska man använda `GET` istället för `POST`?

5.

Använd de mindre vanliga `input`-typerna. använd t.ex. `type="color"` och skicka med en färg i ditt formulär. Färgen ska sedan användas på nästa sida: `form.php`. Sätt bakgrundsfärgen till det som användaren har skrivit in t.ex.

6.

Vi kan även direkt skicka tillbaka informationen till `index.php` med `action="index.php"`. Detta är inget man brukar göra utan man brukar skilja på `php`-filer som hanterar formulär och filer som visar upp formulär t.ex. Men vi gör det bara för övningens skull. Skicka tillbaka informationen till samma sida som du skrev in den på. T.ex. att det ska visas upp *"Du skrev in för kort lösenord"* om vi har skrivit in ett lösenord som är kortare än 6 karaktärer.

7.

Återanvänd de övningarna från tidigare. T.ex. att man kan i ett formulär skicka in information till *fårräknaren* om hur många månader man ska räkna på, att man kan ange hur många gånger mjaumaskinen ska mjaua eller skicka in hur många kilo finska pinnar som finns på Nannas tea party.

8.

Om du har gjort allt ovan så kan det vara en bra idé att kolla närmare på de inbyggda arrayfunktionerna samt datumfunktionerna som finns i PHP. De är rätt många och man får själv lära sig de man måste använda för att lösa sitt problem. Det är även bra att ha lite extra koll på datumformatering, det finns många inbyggda funktioner för att formatera om datum så vi slipper göra det själva. Full lista på alla funktioner koppla till detta finns nedan:

- [PHP Array Function @ W3schools.com](#)
- [PHP Date/Time functions @ W3Schools.com](#)

Lösningsförslag

1.

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

```
//form.php
<?php
```

```
echo 'Hej ' . $_POST["username"] . 'du är född ' . $_POST["birthday"] . 'Din favoritenskap är' . $_POST['css_proper
```

2.

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

```
//form.php
```

```
<?php
if($_POST['password'] == "magic word"){
    echo 'Hej ' . $_POST["username"] . 'du är född ' . $_POST["birthday"] . 'Din favoritenskap är' . $_POST['css_pr
} else{
    echo "<img src='https://media.giphy.com/media/3ohzdQ1Iynzc1JldUQ/giphy.gif' alt='magic word' />";
}
```

3.

```
//form.php
```

```
<?php
```

```
/*
```

```
 * strtotime takes a string and returns a date variable. You could just
 * have a text field where you enter 25 and check that value but this is
 * more fun
 */
```

```
$birthday = strtotime($_POST['birthday']);
$now = strtotime('now');
```

```
//Difference is in seconds, so we need to break it down to years
```

```
$age = ($now - $birthday)/60/60/24/356;
```

```
if($_POST['password'] == "magic word" && $age > 25){
    echo 'Hej ' . $_POST["username"] . 'du är född ' . $_POST["birthday"] . 'Din favoritenskap är' . $_POST['css_pr
} else{
    echo "<img src='https://media.giphy.com/media/3ohzdQ1Iynzc1JldUQ/giphy.gif' alt='magic word' />";
}
```

4.

När man använder `GET` så skickas all information öppet i URLen, i en `POST` så är det dolt från URLen. Ingen av dem är säkrare än den andra egentligen men när man t.ex. vill skicka lösenord så är det bäst att köra via `POST` då lösenordet inte syns i URLen. De flesta formulär som skickas är via `POST` och det är det vi kommer att använda mest. Båda fungerar nästan på samma sätt, man kan göra exakt likadana saker med dem i PHP.

5.

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

```
<head>
  <style>
    body{
      background-color: <?php echo $_POST['color'] ?>
    }
  </style>
</head>
<body>
  <h1>Wooh!</h1>
</body>
```

6.

```
//index.php
<html>
<head>
  <title>Sending to index!</title>
</head>
<body>
  <?php

    /*
     * Getting error if we display it without it being set
     * If we submit a form to a page, the content of that
     * form is always available in $_POST regardless of which
     * php-file we send it to
     */
    if(isset($_POST["username"])){
      echo $_POST["username"];
    }
    if(isset($_POST["css_property"])){
      echo $_POST["css_property"];
    }
    if(isset($_POST["birthday"])){
      echo $_POST["birthday"];
    }
  ?>

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