



Översikt

- Formulär.
- Validering med HTML5.
- Validering med JavaScript.
- Tekniken bakom.

.eerc

Lektion 1: Formulär

- Deklaration av formulär i HTML5.
- Fält för indata och element.
- Attribut.
- Validering.
- Se till så att fält inte är tomma.
- Validera numeriskt värde.

.eerc

Deklaration av formulär i HTML5

```
<html>
<body>
  <form name="userlogin" method="post" action="">
    <fieldset>
      <legend> Ange din inloggning: </legend>
      <div id="usernamefield" class="field">
        <label for="username"> Namn på användare: </label>
        <input id="username" name="username" type="text" placeholder="För- och efternamn" />
      </div>
      <div id="passwordfield" class="field">
        <label for="password"> Lösenord: </label>
        <input id="password" name="password" type="password" placeholder="Lösenord" />
      </div>
    </fieldset>
    <input type="submit" value="Logga in" />
  </form>
</body>
</html>
```

The screenshot shows a code editor with the above HTML code and a browser window displaying the rendered form. The form has two input fields: a text field for the username and a password field. A 'Logga in' button is at the bottom.

Fält för indata och element

- HTML5 definierar ett brett utval av nya typer för indata, några av dessa är inte implementerade av alla webbläsare.

```
<html>
<body>
  <div id="biltillverkare">
    <form name="val">
      <select id="biltillverkare" name="biltillverkare">
        <optgroup label="Europa">
          <option value="volvo">Volvo</option>
          <option value="audi">Audi</option>
        </optgroup>
        <optgroup label="USA">
          <option value="chrysler">Chrysler</option>
          <option value="ford">Ford</option>
        </optgroup>
      </select>
    </form>
  </div>
</body>
</html>
```

The screenshot shows a code editor with the above HTML code and a browser window displaying the rendered dropdown menu. The dropdown menu is titled 'Biltillverkare' and has four options: Europe, Audi, USA, and Ford.

Attribut

- Attribut modifierar beteendet för typer av indata och formulär, för att ge bättre respons till användare och öka användbarheten:
 - autofocus.
 - autocomplete.
 - required.
 - pattern.
 - placeholder.
 - m fl.

```
<html>
<body>
  <div id="biltillverkare">
    <form name="val">
      <select id="biltillverkare" name="biltillverkare">
        <optgroup label="Europa">
          <option value="volvo">Volvo</option>
          <option value="audi">Audi</option>
        </optgroup>
        <optgroup label="USA">
          <option value="chrysler">Chrysler</option>
          <option value="ford">Ford</option>
        </optgroup>
      </select>
    </form>
  </div>
</body>
</html>
```

The screenshot shows a code editor with the above HTML code and a browser window displaying the rendered dropdown menu. The dropdown menu is titled 'Biltillverkare' and has four options: Europe, Audi, USA, and Ford.

Övning Arbeta med formulär



Lektion 2: Validering

- Validering.
- Se till att fält inte är tomma.
- Validera numeriskt värde.
- Validera text värde.
- Liknande beteende i alla webbläsare.

.eeec

Validering

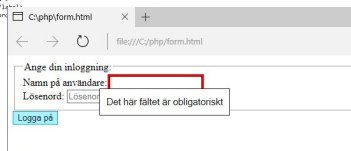
- Vad användare ger in, kan variera i korrekthet, relevans och kvalité.
- Validering på klientsidan kan förbättra användarvänligheten.
- Finns fortfarande behov av validering på serversidan.

.eeec

Se till så att fält inte är tomma

- Kontrollera så att fält inte är tomt, genom att lägga till attributet required.

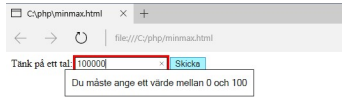
```
<form name="anloggnit" method="post" action="">
  <fieldset>
    <legend> Ange din inloggning </legend>
    <div id="användare" class="field">
      <input type="text" name="användare" />
    </div>
    <div id="password" class="field">
      <input type="password" name="password" />
    </div>
    </fieldset>
    <input type="submit" value="Logga på" />
  </form>
```



Validera numeriskt värde

- Använd attribut min och max, för att definiera minimumvärde och maxvärde.

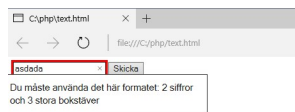
Tänk på ett tal: `<input id="percentage" type="number" min="0" max="100" />`



Validera text värde

- Använd regular expressions för att validera text som användare har gett in.

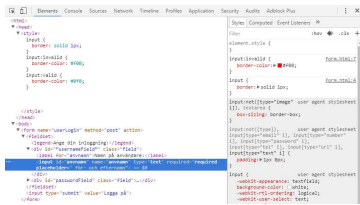
```
<input id="orderRef" name="orderReference" type="text"
  pattern="[0-9]{2}[A-Z]{3}" title="2 siffror och 3 stora bokstäver" />
```



Liknande beteende i alla webbläsare

- CSS kan användas för att få liknade beteende i alla webbläsare.

```
<html>
<head>
<style>
input {
border: solid 1px;
}
input:invalid {
border-color: #f00;
}
input:valid {
border-color: #0f0;
}
</style>
</head>
<body>
```



Lektion 3: Validering med JavaScript

- Validering med JavaScript.
- Trigga på `submit()`.
- Validera numeriskt värde.
- Validera text värde.
- Liknande beteende i alla webbläsare.



Validering med JavaScript

- Fånga händelsen `submit`, för att validera hela formuläret
 - returnerar `true` om information är korrekt, annars `false`.

OBS! Formulär skickas bara om `submit` event handler returnerar `true`!



Validering med JavaScript (forts.)

- Använd JavaScript för att emulera funktioner som det inte finns stöd för i HTML5.
- Formulär skickas bara om true!

```
is@homead
File Edit Format View Help
<form id="scoreForm" onsubmit="return validateForm();" >
  <div id="scoreField" class="field" >
    <input id="score" name="score" type="number" />
  </div>
</form>

<script>
function isAnInteger( text ){
var intTestRegex = /^[0-9-]+$/;
return String(text).search(intTestRegex) != -1;
}

function validateForm()
{
if( ! isAnInteger(document.getElementById('score').value))
return false; /* Nej, det är inte några siffror! returnerar false, formulär skickas bara om true! */
return true;
}
</script>
```

.eeec

Trigga på submit ()

- Bygg din egna funktion som lyssnar på när användaren klickar på submit.

```
<form action="" method="post" onsubmit="return validateForm();" >
...
<script>
function isAnInteger( text ){
var intTestRegex = /^[0-9-]+$/;
return String(text).search(intTestRegex) != -1;
}

function validateForm()
{
if( ! isAnInteger(document.getElementById('id').value))
return false; /* inget numeriskt värde, valideringen fallerar. */
return true;
}
</script>
```

.eeec

Övning Arbeta med validering



.eeec

Lektion 4: Tekniken bakom

- Formulär.
- Client/Server arkitektur.
- Klientsidan.
- Action.
- Method.
- GET.
- POST.
- enctype.
- Skicka filer.
- Säkerhet.

.eeec

Formulär

- Används i de flesta fallen för att skicka information till server.
- Server bearbetar informationen och skickar tillbaks ett svar.

.eeec

Client/Server arkitektur

- Webben är baserat på ett väldigt enkelt protokoll.
- Klient skickar begäran till server, genom att använda HTTP-protokollet.
- Server svarar på denna begäran med samma protokoll.
- På klientsidan är formulär (FORM), bara ett bekvämt sett att konfigurera HTTP-begäran som skickar information till server.

.eeec

Klientsidan

- Elementet `<form>` definierar hur informationen skall skickas.
- Alla attribut är designade för att du skall kunna konfigurera en begäran, som kommer att skickas när du klickar på knappen skicka.
- De två viktigaste attributen är `action` och `method`.

.easesec

Action

- Definierar var informationen skall skickas, måste vara en fungerande URL.
- Om informationen inte tillhandahålls, kommer informationen att skickas till sidan som tillhandahåller formuläret.
- Exempel:

```
<form action="http://course.easesec.se/process.php">
```

.easesec

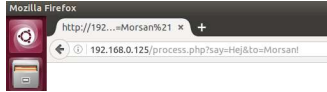
Method

- Detta attribut definierar hur informationen skickas.
- HTTP-protokollet tillhandahåller ett antal olika sätt att skicka begäran på, information i ett HTML formulär kan skickas i åtminstone två av dessa, nämligen GET och POST.
- HTTP-begäran består av två delar, header och body.
- Header består av ett set av global metadata om kapacitet för webbläsaren.
- Body består av information som är nödvändig för att server skall kunna bearbeta specifik begäran.

.easesec

GET

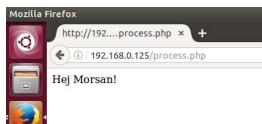
- Metoden GET används av webbläsare för att fråga om server kan skicka tillbaka en given resurs.
- I detta fallet skickar webbläsaren en tom body. Eftersom body är tom, kommer formulär som skickas med denna metod att lägga till informationen till URL.



.eeec

POST

- Metoden POST är lite annorlunda.
- Webbläsaren skickar begäran till server, frågar efter ett svar beroende på den information som skickas i body.
- Om ett formulär skickas med denna metod, kommer informationen att läggas till i body på HTTP-begäran.



.eeec

enctype

- Detta attribut ger möjligt att specificera ett värde för Content-Type i HTTP header.
- Headern är väldigt viktig, för den talar om för server vad för typ av information som skickas.
- Som standard är värdet application/x-www-form-urlencoded.



.eeec

Skicka filer

- För att kunna skicka filer, måste du göra två saker:
 - sätt metoden till POST.
 - sätt värden för enctype till multipart/form-data, detta kommer att splittra informationen i olika delar.

.eeec

Säkerhet

- Cross-Site Scripting (XSS) och Cross-Site Request Forgey (CSRF) är vanliga typer av attacker som kan förekomma när du visar information som har skickats av användare, tillbaka till användaren eller till annan användare.
- SQL injections, där attacken försöker utföra händelser på databasen.
- HTTP header injection och e-mail injection.

.eeec

Repetitionsfrågor

.eeec
