

ITK:P2 F7

Datavalidering med JavaScript

DSV Peter Mozelius

1

JavaScript

- Ett interpreterat språk /skriptspråk
- Objektbaserat men INTE objektorienterat
- Används på klientsidan för t ex:
 - Animeringar
 - Mindre beräkningar
 - Datavalidering
 - "Browser sniffing"

2

JavaScript - Java

- Syntaxen är ofta lik Javas
- MEN, det finns mycket som skiljer
- T ex *dynamisk implicit typning*
- Java har **int, long, float, String** ...
- JavaScript har enbart **var**

3

Ett objektbaserat språk

- Objekt och funktioner/metoder
 - document (objekt)
 - write() (funktion)

Används i ett HTML-dokument enligt:
document.write("Hej ITKP2!");

4

Javascript inne i XHTML

```
<html>
  <head>
    <title>ITKP2 - Exempel1</title>
  </head>
  <body>
    <script type="text/javascript">
      <!-- <![CDATA[
        document.write("Hej ITKP2!");
      // ]] -->
    </script>

  </body>
</html>
```

5

DOM (Document Object Model)

- En modell för att komma åt alla element i ett HTML-dokument
- Dokumentet i sig är också ett objekt: **document**
- Ett annat användbart objekt är **window**
- **window.navigator** rör att komma åt själva webbläsaren

6

Browser sniffing

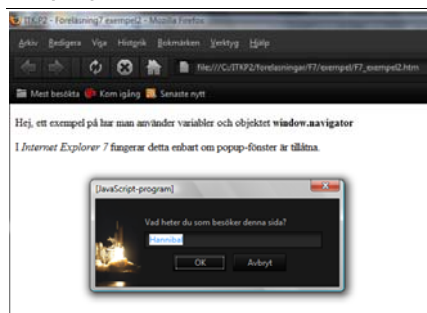
```
//Kräver att popup-fönster är på i IE7
var surfare =
  prompt("Vad heter du som besöker
denna sida?", "Hannibal");

var lasare = window.navigator.appName;

document.write("<h3>Hej ", surfare, ",
jag ser att du använder " , lasare,
".</h3>");
```

7

Popup-fönster i webbläsare



8

Javascript inlänkat från fil

- Det blir lätt grötigt och svårläst om allt ligger i samma fil. Lägg Javascriptet i en egen fil och länka in i XHTML-filen

```
<script type="text/javascript"
src="exempel2.js"> </script>
```

Skriptet ligger i filen: exempel1.js

9

Model - View - Controller

- MVC en vanlig bra princip som vi tidigare har pratat om för javaprogrammering
 - Ett annat exempel på MVC är
- **Model** = innehållet/XHTML-filen
- **View** = stilsättningen/CSS-filen
- **Controller** = javascriptet
- Håll sakerna separerade i olika filer

10

Nackdelar med JavaScript

- Webbläsare kan på vissa företag och organisationer ha JavaScript avstängt av säkerhetsskäl
- Webbläsare kan ha möjligheten till popup-fönster avstängt
- JavaScript-koden ligger i klartext och vem som helst kan knycka den

11

Paus



12

Händelsehantering

○ JavaScript kan hantera olika typer av händelser t ex:

- **onclick:** När användaren klickar
- **onload:** När dokumentet laddas
- **onmouseover:** Muspekaren är över
- **onmouseout:** Muspekaren lämnar
- **onunload:** Surfaren lämnar sidan

...

13

DOM – "cross-browser patch"

- Hur får vi händelsehanteringen att fungera i olika (äldre) webbläsare?
- Jo, javascriptet inleds med ett litet tillägg som gör att dokumentets olika delar kan nås på olika sätt
- Här i det följande exemplet triggas en funktion både för Mozilla och IE

14

DOM - för flera webbläsare

```
IE=(document.all) ? true:false; //IE4+
DOM2=((document.getElementById)&&
      (!IE))?true:false;//Mozilla
setEventByID(id,ev,fu) {
  if(DOM2) {
    document.getElementById(id).addEventListener
      (ev,fu,false); }
  if(IE) {
    document.getElementById(id).attachEvent("on"
      + ev,fu); }
}
+ setEventByID("knapp", "click", validera);
```

15

Datavalidering i JavaScript

- Att kontrollera inmatade värden redan innan de skickas vidare
- Som i de följande exemplen för
 - Ett telefonnummer
- Som i **UppgiftC** för
 - Ett Visa-kort

16

Datavalidering – exempel 3

Du vill kontrollera att ett telefonnummer enbart innehåller siffror

```
function validera() {  
  var indata =  
    document.getElementById("ruta").value;  
  if(isNaN(telnr))  
    window.alert(telnr + "  
    är inte ett giltigt telefonnummer!");  
  }  
}
```

```
setEventByID("knapp", "click", validera);
```

17

Data från XHTML-element

- Hur kan man i ett Javascript hämta data från olika enskilda element i ett XHTML-dokument?
- Jo, kombinationen av:
`document.getElementById("ruta").value`

OCH

```
<input type="text" value="" id="ruta" />
```

18

XHTML inmatningsformulär

- Inmatningsformulär för Javascript och annan interaktion bygger man inne i taggen **<form>** med **<input>**
- Typen av inmatning bestäms sedan man har attributet `type="typ"` t ex
 - **<input type="text" ...**
 - **<input type="radio" ...**
 - **<input type="submit" ...**

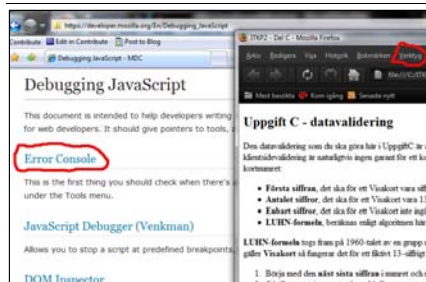
19

Attributen id och class

- Anledningen till att det finns två olika attribut är att de har olika syften:
- **<p id="UppgiftA" >**
 - Unik identifierare för ett enda element
- **<p class="uppgifter" >**
 - Gemensam identifierare för flera element

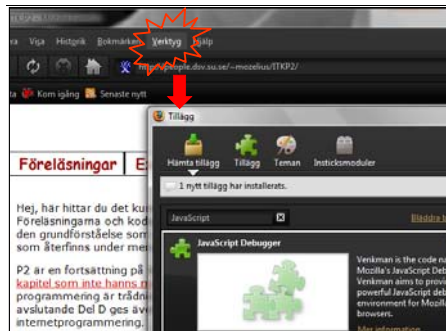
20

Mozillas felkonsol



21

JavaScript Debugger



22

Uppgift C

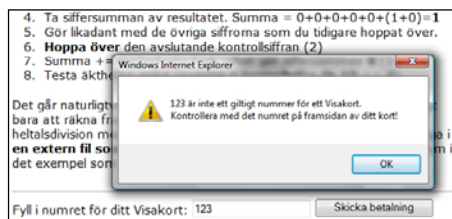
- Exemplet är krypterat

```
<script language = JavaScript
type=text/javascript>
  alert("OBS, exemplet längst
ner fungerar endast i Internet
Explorer");
</script>
```

23

Uppgift C

- Klientsidevalidering av ett Visa-kort



24



Tack för den här kursen!

- Det var allt för årets ITP2
- Den som i sommar vill läsa mer programmering kan fr om 22/2 anmäla sig till sommarkursen *Multimediaprogrammering i Python*
<http://dsv.su.se/courses/python/>

Tack för mig!

25
