



**F8**  
**Webbteknologier 2**  
EDA095 Nätverksprogrammering

**Roger Henriksson**  
**Datavetenskap**  
**Lunds universitet**



## PHP – PHP: Hypertext Preprocessor

- Skriptspråk särskilt lämpligt för webbapplikationer.
- Öppen, gratis, programvara.
- HTML-kod med PHP-taggar inlagda där dynamiskt beteende behövs (jämför JSP).
- Vanlig och mycket spridd teknik:
  - CMS – Content Management Systems (ex LTH).
  - Wiki – Många wikiimplementationer i PHP.
  - Webbkopplingar till databassystem, t.ex. webbshopgränssnitt.
- URL: [www.php.net](http://www.php.net)



## Struktur/taggar

HTML-kod med PHP-taggar insprängda.

```
<?php
```

```
...
```

```
?>
```

All text utanför PHP-taggar skickas oförändrad till klienten.

Filändelse: .php



## Datatyper

Dynamiska typer, ingen variabeldeklaration behövs.

*boolean* Sant/falskt.

*integer* Heltal.

*float* Flyttal.

*string* Strängar.

*array* Vektorer.

*object* Objekt.

*resource* Referens till externa resurser, t.ex. filer.

*NULL* Tomt värde.

Variabler inleds med \$, exempel: `$value`.



## Språkkonstruktioner

Några språkkonstruktioner:

```
$a = $b*3; // Tilldelning, uttryck
echo "Svar: ".$a."\n"; // Utskrift,
                        strängkonkatenering
if ($a==0) { ... } else { ... } // If-sats
for($i=0;$i<10;$i++) { ... } // For-sats
```



## Vektorer

Vektorer är associativa! (Egentligen en "ordered map".)  
Index kan vara av godtycklig typ.

```
$arr = array(); // Tom array
$arr["roger"] = "duktig";
$arr[2] = 45;
$arr[] = "NP"; // Index: högsta numeriska
               // index + 1
```

Iterator över vektorn:

```
foreach($arr as $key => $value) {
    echo $key." : ".$value."\n";
}
```



## Funktioner

```
<?php
function sum($arg_1, $arg_2, $arg_3)
{
    echo "Example function\n".
    $retval = $arg_1+$arg_2+$arg_3;
    return $retval;
}
?>
```

Summan är: <?php echo sum(2,5,8); ?>



## Biblioteksfunktioner

Stor mängd inbyggda biblioteksfunktioner.  
Betoning på webbrelaterade uppgifter.

<i>Audio</i>	<i>Autentisering</i>	<i>Datum/tid</i>
<i>Komprimering</i>	<i>Kreditkorts- betalningar</i>	<i>Filhantering</i>
<i>Teckenkodning</i>	<i>Stränghantering</i>	<i>Bildmanipulering</i>
<i>Mail</i>	<i>Matematik</i>	<i>Nätverk</i>
<i>Kryptering</i>	<i>Databaser (SQL)</i>	<i>Processhantering</i>

Med mera...



## Koppling till webbservern

### Request

Parametrar accessas genom "superglobala" variabler, t.ex:

- `$_GET` Parametrar vid GET-anrop. Vektor.
- `$_PUT` Parametrar vid PUT-anrop. Vektor.
- `$_COOKIE` Cookies. Vektor.

### Response

HTML-kod skrivs till standard output.

Cookies: `setcookie(...);`



## En elektronisk shoppinglista

### Fallstudie:

Implementera en webbaserad tjänst för att hantera en familjs gemensamma shoppinglista.

- Lista över dagligvaror som saknas i hemmet.
- Enkel inläggning av varor från Internetanslutna enheter.
- Stöd både för att skapa listan och vid shoppingtillfället.

### DEMO – shoppinglist.php



## Design

### Lagring

- Textfil med en rad per artikel.

### Webbsida

Formulär med:

- Checkbox för varje artikel.
- Textinmatningsfält för ny artikel

### Implementation

PHP-skript:

1. Läs in artiklar från fil.
2. Om checkbox ikryssad: tag bort motsvarande artikel.
3. Om textinmatningsfält ifyllt: lägg till artikel.
4. Om ändringar gjorts: spara artiklar på fil.
5. Skapa nytt formulär med aktuella artiklar.



## JavaScript

### Skriptspråk för webbläsare

Syntax som påminner lite om Java. I övrigt har JavaScript och java inget med varandra att göra.

### Historia

Skapat av netscape. Ursprungligen kallat LiveScript. Fick namnet JavaScript i Netscape 2.0.

Microsofts variant, JScript, nästan kompatibelt.



## Vad kan JavaScript göra?

### JavaScript kan t.ex:

- Göra beräkningar på tal och strängar.
- Modifiera utseende och innehåll i ett webbläsarfönster.
- Skapa nya fönster och ladda in nya sidor.
- Manipulera HTML-element på en webbsida, t.ex. innehåll i ett formulär.

### JavaScript kan inte:

- Kommuniera direkt via nätverket (via UDP/TCP).
- Använda filsystemet.



## JavaScript och webbsidor

JavaScript kan läggas in som en del av HTML-koden för en webbsida.

JavaScript kan antingen fås att exekvera när webbsidan laddas, eller när någon speciell händelse inträffar.

### JavaScript vid laddning av en webbsida

- Skriptet körs i samband med att webbsidan ritas upp.
- Utdata från skriptet kompletterar den övriga HTML-koden före uppritning.

### Händelsestyrda JavaScript

- Skriptet körs när en viss händelse inträffar, t.ex. att användaren klickar i webbläsarfönstret. Kan påverka webbläsarfönster eller någon HTML-komponent.



## <script>-tagen

HTML-kod för att definiera ett JavaScript

Syntax:

```
<script>
```

```
...
```

```
</script>
```

eller

```
<script language="JavaScript">
```

```
...
```

```
</script>
```

Generell HTML-tag för skript oavsett skriptspråk.

JavaScript är oftast standardskriptspråk.

Andra möjligheter: VBScript.



## Hello, world!

```
<html>
<head><title>HelloWorldScript</title></head>
<body>
<h1>A greeting from our JavaScript:</h1>
<script language="JavaScript">
document.write("Hello, World!");
</script>
</body>
</html>
```





## Datatyper

Variabeldeklaration – dynamiska typer!

```
var name;
```

### Typer

*tal* Numeriska tal, både heltal och flyttal.

*sträng* Texter ungefär som i Java.

*boolean* Sant eller falskt.

*undefined* Värde oinitialiserade variabler har.

*null* Tomt värde.

Dessutom finns vektorer och objekt.



## Några satskonstruktioner

Tilldelning

```
a = a + 3;
```

If-sats

```
if (a>b) {
    document.writeln("Tal A är störst!");
} else {
    document.writeln("Tal A är INTE störst!");
}
```

For-sats

```
for(i=1;i<=10;i++) {
    document.writeln(i+ " "+i*i);
}
```

Väldigt likt Java!



## Funktioner

Precis som man kan definiera metoder i Java kan man definiera funktioner i JavaScript.

```
<script language="JavaScript">
function printSquare(x) {
    document.writeln(x+ " "+x*x+"<br>");
}
var i;
for(i=1;i<=30;i++) {
    printSquare(i);
}
</script>
```

Kan till exempel användas för att beskriva vad som ska hända när någon viss händelse inträffar.

**DEMO - Kvadrattabell (square.html)**



## Standardobjekt

För att påverka omvärlden, t.ex. webbläsarfönstret, används olika s.k. standardobjekt.

Exempel:

<i>navigator</i>	Tillhandahåller information om webbläsaren.
<i>window</i>	Funktionalitet för att öppna/påverka fönster.
<i>document</i>	Motsvarar HTML-dokumentet i webbläsaren och ger tillgång till alla HTML-komponenter i detta, t.ex. formulär.

### Standardklasser

Date, Math, Array.



## Formulär och JavaScript

JavaScript är bland annat bra för att behandla inmatade data i ett HTML-formulär.

Låt användaren fylla i ett värde samt anropa sedan JavaScript-funktionen "compute()":

```
<form name="inputform">
<input name="field" type="text">
<input type="button" value="Enter!" onClick="compute()">
</form>
```

Ett JavaScript kan nå fälten i formuläret via "document"-objektet:

```
var userValue;
userValue = document.inputform.field.value;
```

**DEMO - Gissa talet (game.html)**



## Animeringar

Man kan ange att JavaScriptkod ska köras efter en viss fördröjning eller med ett regelbundet intervall. Kan t.ex. användas för enklare animeringar.

Exempel:

```
var buttonNotPressed =
    setTimeout("window.alert(
        'Tryck på OK för att fortsätta!')",10000);

clearTimeout(buttonNotPressed);

var change = setInterval("changeImage()",2000);

clearInterval(change);
```



## Ajax – Asynchronous JavaScript and XML

- Samling av relaterade tekniker för interaktiv webb.
- Förbättrar svarstider genom att HTML-sidor genereras lokalt mha JavaScript istället för på servern. Endast små datamängder överförs.
- Data/skript överförs asynkront i bakgrunden.
- Bygger på JavaScript och (ofta, men inte alltid) XML för överföring av data till/från servern.
- Sidans struktur kan manipuleras dynamiskt.

**Exempel:**

- Sidor med sökresultat. Endast själva sökresultatet behöver överföras/genereras och bytas ut på sidan.