Programmering i PHP

Marcus Rej<65533>s

marcus@rejas.se

Magnus M<65533><65533>tt<65533>

magnus@php.net

Programmering i PHP

av Marcus Rej<65533>s och Magnus M<65533><65533>tt<65533>

Publicerad Utkast

Denna bok <65533>r anpassad f<65533>r gymnasieskolans kurser Programmering A och B med kurskoderna DTR1207 och DTR1208 (PPHP1408). Den kan naturligtvis anv<65533>ndas <65533>ven i andra sammanhang, till exempel sj<65533>lvstudier, studiecirklar eller annan l<65533>rarledd utbildning.

Till boken kommer att finnas f<65533>rslag p<65533> <65533>vningar man kan utf<65533>ra f<65533>r att praktiskt ut<65533>va de f<65533>rdigheter man skaffar sig under studierna.

Tryckta b<65533>cker kan k<65533>pas av TriNix AB i Helsingborg, telefon 042-127800.

Revisionshistorik

Revision $Id: programmeringAB.xml,v 1.23 2006/06/11 10:27:10 rejas Exp $
*** Utvecklingsversion ***

Innehållsförteckning

Tabellförteckning

F<65533>rord

1. Tack till

Vi vill b<65533>rja denna bok med att tacka alla som hj<65533>lpt oss med den. Speciellt vill vi tacka Micke Karlsson som f<65533>reslagit f<65533>rb<65533>ttringar, Jerry Segerholm som hittat stavfel och Daniel Wahlgren som hittat b<65533>de stavfel och sakfel. Tack!

Vi vill ocks<65533> tacka dig som l<65533>ser denna bok. Hittar du n<65533>got som <65533>r fel eller som du tycker att vi kan g<65533>ra p<65533> n<65533>got b<65533>ttre s<65533>tt s<65533> tveka inte att h<65533>ra av dig till oss. V<65533>ra adresser st<65533>r i b<65533>rjan av boken. Vi har v<65533>ra begr<65533>nsningar som f<65533>rfattare men genom att sl<65533>ppa denna bok fri f<65533>rv<65533>ntar vi oss att du som l<65533>sare skall skicka kommentarer s<65533> att vi tillsammans kan g<65533>ra den mycket b<65533>ttre <65533>n vad vi och kanske du skulle ensamma.

Kapitel 1. Kort historik

1.1. F<65533>re 1900

Man b<65533>rjade faktiskt att utveckla den metodiken som anv<65533>nds i dagens datorer redan p<65533> 1800-talet. Den f<65533>rsta kalkylatorn som r<65533>knade digitalt uppfann och byggde engelsmannen Charles Babbage 1832! Den opererade p<65533> sexsiffriga nummer och kunde l<65533>sa andragradspolynom med sex siffrors noggrannhet. Efter denna maskin b<65533>rjade Babbage p<65533> en <65533>nnu st<65533>rre maskin, the analytical engine. Denna maskin var en enorm uppfinning som introducerade saker som vi fortfarande k<65533>nner igen i dagens datorer. Maskinen hade in- och utmatningsenhet, minne samt kalkylator och kontrollenhet (processor). Dessv<65533>rre fick han den aldrig att fungera. Idag vet vi varf<65533>r, det var en enorm uppgift som inte gick att l<65533>sa med d<65533>tidens kunskaper och verktyg. Babbage dog i 1871. D<65533> var det inte m<65533>nga som visste vem han var men nu n<65533>mns han i varenda kurs som innefattar datorhistoria v<65533>rlden <65533>ver.

Mycket mer saker h<65533>nder under 1800-talet. Men vi hoppar direkt till 1900-talet.

1.2. 1900-talet

1936 skrev matematikern Alan Turing en rapport som bland annat beskriver hur en digital dator skulle fungera. Rapporten beskriver en matematisk maskin som kunde utf<65533>ra logiska operationer och l<65533>sa, skriva och radera bin<65533>ra symboler (1:or och 0:or) p<65533> en o<65533>ndlig remsa. Dessa kom sedan att kallas Turing-maskiner.

1941 blev den f<65533>rsta fungerande turingmaskinen klar. Den skapades av Konrad Zuse. Den var fritt programmerbar och helautomatisk. Den hade en klockfrekvens p<65533> 5.33 Hz och vad byggd med rel<65533>er. Det dr<65533>jde n<65533>gra <65533>r till innan datorerna blev elektroniska.

Den f<65533>rsta moderna datorn blev en maskin som hette ENIAC. Den stod klar 1946. ENIAC <65533>r en f<65533>rkorting f<65533>r Electronic Numerical Integrator And Computer. Den var stor som ett hus (18000 elektronr<65533>r) och kunde mindre <65533>n den enklaste kalkylator kan idag. En som var inblandad i utvecklingen av ENIAC var John von Neumann. Von Neumann skapade en datorarkitektur, von Neumann-arkitekturen, som datorer <65533>n idag byggs efter. Det var den f<65533>rsta arkitekturen d<65533>r ber<65533>kningsenhet och minne var separerade. Mellan dem fanns det en buss <65533>ver vilken data och instruktioner transporterades. Von Neumann anses av m<65533>nga vara den moderna datorns fader. Den f<65533>rsta Von Neuman-datorn, som han sj<65533>lv var med och byggde, hette EDVAC.

1.3. Nutid

Utvecklingen st<65533>r p<65533> intet s<65533>tt still. Datorerna utvecklas st<65533>ndigt. Varje <65533>r blir datorerna snabbare och billigare. Det l<65533>r dock dr<65533>ja ett tag innan n<65533>got s<65533> revolutionerande som Babbages kalkylator eller ENIAC ser dagens ljus. Sen hur l<65533>ng tid det tar, det <65533>terst<65533>r att se.

1.4. Mer l<65533>sning

P<65533> adressen *http://www2.fht-esslingen.de/studentisches/Computer_Geschichte/fold1.html* (http://www2.fht-esslingen.de/studentisches/Computer_Geschichte/fold1.html) finns en bra tidslinje f<65533>r datorns historia.

Kapitel 2. Programmeringsspr<65533>k

Detta avsnitt ger en orientering om olika programmeringsspr<65533>k och varf<65533>r det finns s<65533> m<65533>nga och n<65533>gra grundl<65533>ggande egenskaper hos olika spr<65533>k och olika familjer av spr<65533>k.

2.1. Olika spr<65533>k till olika saker

Det finns idag hundratals olika programmeringsspr<65533>k. De har kommit till av olika anledningar och har olika syften. Vissa <65533>r besl<65533>ktade och har arv fr<65533>n varandra och andra har utvecklats f<65533>r att fylla ett speciellt <65533>ndam<65533>l.

Anledningen till att det finns s<65533> m<65533>nga spr<65533>k och alltid kommer nya <65533>r naturligtvis att de <65533>r bra p<65533> olika saker och att olika personer uppskattar spr<65533>ken olika mycket.

Vi kan b<65533>rja med att dela upp spr<65533>ken i tv<65533> grupper efter hur programmen kompileras och k<65533>rs. Alla program skrivs i n<65533>gon form av k<65533>llkod. Denna k<65533>llkod m<65533>ste sedan <65533>vers<65533>ttas till n<65533>got som en dator kan f<65533>rst<65533>. Datorn f<65533>rst<65533>r bara 1 och 0, p<65533> eller av. Detta kallas kompilering och kan g<65533>ras vid olika tillf<65533>llen.

2.2. Kompilerande spr<65533>k

Datorn kan ju inte l<65533>sa utan det som man skriver i sina program m<65533>ste <65533>vers<65533>ttas till n<65533>got som datorn f<65533>rst<65533>r. Det datorn f<65533>rst<65533>r kallas f<65533>r maskinkod eller bin<65533>rkod. Denna kod best<65533>r av maskininstruktioner som <65533>r, n<65533>st intill, om<65533>jliga f<65533>r en m<65533>nniska att f<65533>rst<65533>.

Ett kompilerande spr<65533>k <65533>r ett programmeringsspr<65533>k d<65533>r k<65533>llkoden med hj<65533>lp av olika verktyg <65533>vers<65533>tts till maskinkod. Maskinkoden blir d<65533> ett frist<65533>ende program som kan k<65533>ras direkt av datorn. Man s<65533>ger att man kompilerar koden. Koden kopileras allts<65533> i samband med utvecklingen och inte i samband med exekveringen av programmet. Mer om det i ett annat avsnitt.

Exempel p<65533> kompilerande spr<65533>k <65533>r: C, C++, Pascal och m<65533>nga fler.

2.3. Interpreterande spr<65533>k

Ett interpreterande programspr<65533>k, som ocks<65533> kallas skriptspr<65533>k, <65533>r ett programspr<65533>k som inte kompilerar det program som programmeraren har skrivit f<65533>rr<65533>n det k<65533>rs, och som g<65533>r det varje g<65533>ng det k<65533>rs. Ibland kompileras hela programmet innan det k<65533>rs och ibland kompileras det rad f<65533>r rad. Detta g<65533>r att utvecklingen g<65533>r snabbare eftersom programmet inte beh<65533>ver kompileras vid testk<65533>rning men det inneb<65533>r ocks<65533> att det f<65533>rdiga programmet blir l<65533>ngsammare eftersom det m<65533>ste kompileras varje g<65533>ng det k<65533>rs.

Motorn som k<65533>r ett interpreterande spr<65533>k kallas interpretator, programtolk eller tolkare.

Exempel p<65533> interpreterande spr<65533>k <65533>r: Perl, PHP, Python, TCL, Bash.

2.4. Andra typer av spr<65533>k

Det finns andra s<65533>tt att l<65533>sa uppgiften p<65533>. Till exempel att man kompilerar koden till en mellankod som sedan en interpreterare tolkar. S<65533> fungerar till exempel Java.

2.5. F<65533>r- och nackdelar

Tabell 2-1. F<65533>rdeler och nackdelar med kompilerande och interpreterande spr<65533>k.

F<65533>rdelar	**Nackdelar**
Kompilerande	
Snabbt att k<65533>ra	L<65533>ngsam programmering
L<65533>tt att distribuera	Sv<65533>rt att portera
Interpreterande	
Snabb programmering	L<65533>ngsamt att k<65533>ra
Mycket l<65533>tt att portera (om tolkare finns)	Sv<65533>rare att distribuera (tolkare m<65533>ste finnas)

2.6. Exempel p<65533> olika spr<65533>k som ni b<65533>r k<65533>nna till

Det finns tusentals, jo s<65533>kert, olika programmeringsspr<65533>k. H<65533>r listas de jag tycker ni b<65533>r k<65533>nna till och hur programmet "Hello World!" ser ut i dem.

2.6.1. C

C <65533>r ett av de mest utbredda spr<65533>ken. Det <65533>r en vidareutveckling av spr<65533>ket B. Utvecklades vid AT&T Bell Labs samtidigt med operativsystemet UNIX®. C anses som ett ganska maskinn<65533>ra spr<65533>k. Flera operativsystem <65533>r till stor det skrivna i C. Hello World i C ser ut s<65533> h<65533>r:

Exempel 2-1. Hello World i C

```
/*
 * Hello World i C
 */
#include <stdio.h>

int main() {
        printf("Hello, world!\n");
        return 0;
}
```

2.6.2. C++

C++ utveckladen av Bjarne Stroustrup vid AT&T Bell Labs och <65533>r en ut<65533>kning av programspr<65533>ket C. Till skillnad fr<65533>n C s<65533> <65533>r C++ objektorienterat (eller har b<65533>ttre st<65533>d f<65533>r det i alla fall).

Hello World i C++ ser ut s<65533> h<65533>r:

Exempel 2-2. Hello World i C++

```
/*
 * Hello World i C++
 */
#include <iostream>

int main () {
        cout << "Hello world" << endl;
}
```

2.6.3. C# (C-sharp eller Ciss)

Objektorienterat spr<65533>k utvecklat av Microsoft f<65533>r att m<65533>ta Java fr<65533>n Sun. Spr<65533>ket <65533>r, liksom Java, halvinterpreterande och har mycket influenser fr<65533>n C och C++.

Hello World i C# ser ut s<65533> h<65533>r:

Exempel 2-3. Hello World i C#

```
//
// Hello World i C#
//
using System;
class HelloWorld {
        static void Main() {
                Console.WriteLine("Hello, world!");
        }
}
```

2.6.4. Java

Objektorienterat spr<65533>k utvecklat av Sun Microsystems. Spr<65533>ket <65533>r halvinterpreterande och har mycket influenser fr<65533>n C och C++. En stor f<65533>rdel med Java <65533>r att program skrivna i Java <65533>r n<65533>stan platformsoberoende. Du kan allts<65533> k<65533>ra dem i alla milj<65533>er till vilka det finns en javamotor.

Hello World i Java ser ut s<65533> h<65533>r:

Exempel 2-4. Hello World i Java

```
//
// Hello World i Java
//
class HelloWorld {
        public static void main (String s[]) {
                System.out.println("Hello world");
        }
```

```
}
```

2.6.5. Mer l<65533>sning

En sida med j<65533>ttem<65533>nga exempel p<65533> "Hello World!" i olika spr<65533>k. (*http://www2.latech.edu/~acm/HelloWorld.shtml* (http://www2.latech.edu/~acm/HelloWorld.shtml))

Kapitel 3. Fr<65533>n k<65533>llkod till program

I detta avsnitt beskrivs vad en kompilator <65533>r och varf<65533>r den anv<65533>nds. Detta ing<65533>r i kursen men anv<65533>nds inte i PHP eftersom det <65533>r ett s<65533> kallat skriptspr<65533>k. I fallet med PHP tar en tolkare hand om allt detta p<65533> servern n<65533>r programmet k<65533>rs.

3.1. Kompilering

N<65533>r man "kompilerar" ett program s<65533> utf<65533>r man egentligen 3 steg. Ett fj<65533>rde steg utf<65533>rs precis innan programmet laddas f<65533>r att k<65533>ras. Dessa steg behandlar vi h<65533>r.

Stegen <65533>r f<65533>rbehandling av k<65533>llkoden, kompilering, assemblering och slutligen l<65533>nkning. Dessa steg behandlas h<65533>r.

3.1.1. F<65533>rbehandling av k<65533>llkoden (preprocessing)

Det f<65533>rsta som h<65533>nder med din k<65533>llkod <65533>r att den g<65533>r igenom n<65533>got som kallas en preprocessor. Den tar bort alla kommentarer, som ju i alla fall bara betyder n<65533>got f<65533>r m<65533>nniskor och formaterar din kod s<65533> att den passar kompilatorn.

3.1.2. Kompilering

Sedan <65533>r det kompilatorns tur. Det <65533>vers<65533>tter k<65533>llkoden till ett mellanspr<65533>k anpassat f<65533>r den m<65533>lmilj<65533> som man kompilerar f<65533>r. Detta mellanspr<65533>k kallas assembler och skall sedan behandlas av en assemblerare.

3.1.3. Assemblering

Assemleraren <65533>vers<65533>tter assemblerkoden till maskinkod. Denna kod <65533>r relokerbar, det vill s<65533>ga den <65533>r inte bunden till fasta adresser i minnet. Dessa relokerbara

adresser m<65533>ste dock <65533>ndras innan programmet kan k<65533>ras, det g<65533>r laddaren (loader).

3.1.4. Laddning och l<65533>nkning

Detta <65533>r egentligen tv<65533> steg men de g<65533>rs n<65533>stan alltid tillsammans och de g<65533>rs varje g<65533>ng programmet skall k<65533>ras. De g<65533>rs oftast av samma rutin. Laddning inneb<65533>r att programmet flyttas till ett st<65533>lle i minnet d<65533>r det kan k<65533>ras och se till att adresserna i programmet anpassas efter det st<65533>lle det skall k<65533>ras. L<65533>nkaren beh<65533>ver bara jobba om det k<65533>rbara programmet beh<65533>ver delar fr<65533>n flera olika filer, s<65533> kallade delade bibliotek (shared libraries, dll:er). L<65533>nkaren ser d<65533> till att h<65533>nvisningarna till dessa i programmet blir riktiga och kontrollerar att de finns tillg<65533>ngliga.

3.1.5. Mer l<65533>sning

Om du <65533>r intresserad kan du l<65533>sa denna l<65533>nk *http://techpubs.sgi.com/library/dynaweb_docs/0620/SGI_Developer/books/Pascal_PG/sgi_html/ch02.html* (http://techpubs.sgi.com/library/dynaweb_docs/0620/SGI_Developer/books/Pascal_PG/sgi_html/ch02.html) d<65533>r kompileringen i Pascal beskrivs mer ing<65533>ende.

Kapitel 4. Hall<65533> V<65533>rlden!

Detta avsnitt beskriver hur ett enkelt PHP-script <65533>r uppbyggt.

4.1. Programmeringsmilj<65533>n

Som vi har sagt tidigare s<65533> skapar man ett program p<65533> f<65533>ljande s<65533>tt (n<65533>got f<65533>renklat):

- Skapa PHP-fil
- Kopiera till servern
- Provk<65533>r

Du kan anv<65533>nda vilken editor du vill f<65533>r att skapa php-filer. Notepad fungerar alldeles utm<65533>rkt men det finns de med fler finesser. Vim <65533>r en av dem. Den har st<65533>d bland annat f<65533>r f<65533>rgl<65533>ggning av kod och finns f<65533>r b<65533>de Linux® och Windows att h<65533>mta fr<65533>n deras hemsida: *http://vim.sf.net* (http://vim.sf.net)

4.2. Hall<65533> v<65533>rlden!

N<65533>stan alla programmeringskurser b<65533>rjar med att man skapar ett program som heter Hello World. Hello World <65533>r ett program som inte g<65533>r n<65533>got annat <65533>n att skriva ut "Hello World" p<65533> sk<65533>rmen.

Vi b<65533>rjar med att titta p<65533> hur det ser ut i PHP.

Exempel 4-1. Hello World i PHP

```
<html>
<head>
   <title>Hall<65533> V<65533>rlden!</title>
</head>
<body>
   <p>
      <?php
         echo "Hall<65533> V<65533>rlden!";
      ?>
   </p>
```

```
</body>
</html>
```

S<65533>, vad g<65533>r nu detta program? Som ni s<65533>kert kan se s<65533> <65533>r det HTML i b<65533>rjan och slutet av filen. Det enda av filen som <65533>r PHP <65533>r mellan tecknen <?php och ?>. Det <65533>r f<65533>r att tolkaren skall veta n<65533>r den skall exekvera raderna ist<65533>llet f<65533>r att skriva ut dem.

Den enda PHP-koden i detta lilla program <65533>r echo "Hall<65533> V<65533>rlden!";. Echo talar om att n<65533>got skall skrivas ut (Hall<65533> V<65533>rlden!) och semikolonet talar om att en sats i programmet slutar. En sats kan omfatta flera rader men avslutas alltid av ett semikolon.

Man kan sj<65533>lvklart skriva ett "Hello World!" program som enbart anv<65533>nder sig av PHP och ingen HTML (dock utan den formatering som kommer med HTML).

Exempel 4-2. Hello World i PHP utan HTML

```
<?php
echo "Hall<65533> V<65533>rlden!";
?>
```

4.3. <65533>vningsuppgifter

- <65533>ndra programmet s<65533> att det skriver ut n<65533>got annat <65533>n "Hall<65533> V<65533>rlden!". Testa att det fungerar som du vill.
- G<65533>r s<65533> att tv<65533> rader skrivs ut. (Anv<65533>nd HTML-koder).

Kapitel 5. Webbrowser, webbserver och program

N<65533>r man skriver och k<65533>r ett PHP-program <65533>r det m<65533>nga delar som skall samverka. Man skriver ett program som skall tolkas av en PHP-tolk. Sedan skall utdata av detta program hanteras av en webbl<65533>sare. Hur detta h<65533>nger ihop f<65533>rklaras i detta kapitel.

5.1. Webbl<65533>saren

Webbl<65533>saren <65533>r den del av kedjan som <65533>r n<65533>rmast anv<65533>ndaren. Webbl<65533>saren <65533>r det som du eller dina anv<65533>ndare kommer i kontakt med. F<65533>r att webbl<65533>saren skall visa vettiga saker f<65533>r anv<65533>ndaren g<65533>ller att den matas med vettiga saker fr<65533>n ditt program.

Webbl<65533>saren f<65533>rst<65533>r HTML [1] s<65533> det som skall komma till webbl<65533>saren <65533>r HTML. Webbl<65533>saren f<65533>rst<65533>r inte PHP-kod s<65533> PHP-koden m<65533>ste tolkas av en PHP-tolk p<65533> webbservern.

5.2. Webbservern

Webbservern har till uppgift att serva webbl<65533>saren med webbsidor. K<65533>r man PHP s<65533> <65533>r det <65533>ven p<65533> webbservern som PHP-koden tolkas. PHP-tolken g<65533>r inget f<65533>r att det som kommer ut ur PHP-koden skall bli giltig HTML utan det <65533>r upp till dig som programmerare att se till att den <65533>r det.

Webbservern har ofta en m<65533>ngd olika funktioner beroende p<65533> vad den f<65533>r tillf<65533>llet skall g<65533>ra. Du kan till exempel ha rena HTML-sidor parallellt med din PHP-filer. Webbservern vet vilka sidor som <65533>r PHP och vilka som <65533>r HTML och hanterar dem d<65533>refter.

5.3. Program

Programmet, eller skriptet, skall fungera s<65533> att det skriver ut HTML-kod som webbservern kan skicka till anv<65533>ndarens webbl<65533>sare.

Noter

1. Moderna webbl<65533>sare f<65533>rst<65533>r en massa annat ocks<65533>, men det som <65533>r relevant i denna kurs <65533>r HTML

Kapitel 6. Variabler

Detta avsnitt beskriver vad variabler <65533>r och hur man anv<65533>nder dem i ett PHP-script. Det tar <65533>ven upp skillnaden mellan variabler i PHP och i andra vanliga kompilerande spr<65533>k.

6.1. Vad <65533>r en variabel

En variabel <65533>r en platsh<65533>llare f<65533>r ett v<65533>rde som <65533>ndras under programmets g<65533>ng. Motsatsen till variabel <65533>r en konstant. En konstant kan eller f<65533>r inte <65533>ndras under programmets g<65533>ng.

Ett exempel p<65533> en variabel kan vara termer i en addition. T<65533>nk dig att du vill g<65533>ra ett program som skall r<65533>nka ut en summa. Man skulle kunna g<65533>ra programmet med bara konstanter, det skulle se ut s<65533> h<65533>r:

Exempel 6-1. Exempel med konstanter

```
<html>
 <head>
    <title>Addition</title>
 </head>
 <body>
    <?php
       echo "Summan av talen 1 och 2 <65533>r: ", 1 + 2;
    ?>
 </body>
</html>
```

Som du kan se s<65533> r<65533>knar PHP ut vad 1 + 2 blir. Men vad har man f<65533>r nytta av ett program som bara r<65533>knar ut 1 + 2. Vi vill ju kunna mata in vilka v<65533>rden som helst f<65533>r termerna.

Se nedanst<65533>ende exempel:

Exempel 6-2. Exempel med variabler

```
<html>
 <head>
  <title>Addition</title>
 </head>
```

```
 <body>
  <?php
   $tal1 = 2;
   $tal2 = 3;
   echo "Summan av talen $tal1 och $tal2 <65533>r: ", $tal1 + $tal2;
  ?>
 </body>
</html>
```

I exemplet ovan anv<65533>nder vi tv<65533> variabler, $tal1 och $tal2. I PHP b<65533>rjar alla variabelnamn med ett dollartecken. Programmet ovan <65533>r dock lika statiskt som det f<65533>sta programmet. V<65533>ra variabler f<65533>r ju samma v<65533>rde (2 och 3) varje g<65533>ng programmet k<65533>rs. De kan inte p<65533>verkas av n<65533>got.

Vi tar ytterligare ett exempel:

Exempel 6-3. Exempel med variabler och $_GET

```
<html>
 <head>
  <title>Addition</title>
 </head>
 <body>
  <?php
   $tal1 = $_GET['tal1'];
   $tal2 = $_GET['tal2'];
   echo "Summan av talen $tal1 och $tal2 <65533>r: ", $tal1 + $tal2;
  ?>
 </body>
</html>
```

I exemplet ovan h<65533>mtar vi v<65533>rdena till $tal1 och $tal2 fr<65533>n en speciell variabel som heter $_GET. I $_GET finns det data som ges till programmet via URL:en. Till exempel s<65533> kommer programmet om det anropas som http://dinserver/programnamnet.php?tal1=100&tal2=199 att skriva ut "Summan av talen 100 och 199 = 299".

Testa sedan andra tal och se vad som h<65533>nder.

Du kanske provade att s<65533>tta ett av talen till n<65533>gor annat <65533>n ett tal, till exempel bokst<65533>ver? I s<65533> fall s<65533>g du att PHP inte kan summera text. Observera att om du matar in ett decimaltal s<65533> m<65533>ste decimalpunkt och inte komma anv<65533>ndas.

6.2. Datatyper

Alla programmeringsspr<65533>k arbetar med olika datatyper. I PHP beh<65533>ver du inte ange vilken typ det <65533>r du jobbar med utan det listar programtolken ut fr<65533>n inneh<65533>llet. Det <65533>r dock v<65533>ldigt viktigt att man h<65533>ller reda p<65533> vilka typer ens variabler har eftersom det kan bli underliga fel annars.

F<65533>ljande typer finns:

6.2.1. Skal<65533>ra

- boolean, bool (true, false)
- integer, int (heltal)
- float, double, real (reella tal)
- string (Textstr<65533>ngar)

6.2.2. Sammansatta

- arrayer (En samling av v<65533>rden som indexeras antingen av en integer eller string).
- objekt (Kommer inte att ing<65533> i denna kurs, se manualen om du <65533>r intresserad).

6.2.3. <65533>vriga speciella

- NULL (Variabeln har inget v<65533>rde alls).
- resource (Inneh<65533>ller en referens till en extern tillg<65533>ng).

6.3. <65533>vningar

6.3.1. Addition

Skapa ett program som med hj<65533>lp av variabler summerar tre tal.

Kapitel 7. Operatorer

Detta avsnitt beskriver vad operatorer <65533>r och hur man anv<65533>nder dem i ett PHP-script.

7.1. Vad <65533>r en operator?

En operator <65533>r n<65533>got som verkar p<65533> en eller flera termer. Exempel p<65533> operatorer <65533>r +, -, * och / som g<65533>r precis vad ni tror att de g<65533>r.

Det finns olika typer av operatorer. Aritmetiska operatorer, som de ovan, opererar bara p<65533> tal. Sedan finns det tilldelningsoperatorer som g<65533>r tilldelningar och s<65533> vidare. H<65533>r kommer n<65533>gra av de vi kommer att jobba med:

7.1.1. Aritmetiska operatorer

Tabell 7-1. Aritmetiska operatorer

Exempel	Namn	Resultat
$a + $b	Addition	Summan av $a och $b
$a - $b	Subtraktion	Differensen av $a och $b
$a * $b	Multiplikation	Produkten av $a och $b
$a / $b	Division	Kvoten av $a och $b
$a % $b	Modulus	Resten av division mellan $a och $b

7.1.2. Tilldelningsoperatorer

Det finns bara en tilldelningsoperator och den heter helt enkelt "tilldelas". Den representeras av ett lika-med-tecken (=). S<65533> h<65533>r kan den anv<65533>ndas:

Exempel 7-1. Tilldelningsoperatorn

```
<?php
$c = $a + $b; // Utl<65533>ses, c tilldelas v<65533>rdet av a + b
?>
```

Det finns vissa andra tilldelningsoperatorer, men dessa beh<65533>ver ni inte kunna. Jag tar dem kort h<65533>r.

Exempel 7-2. Tilldelningsoperatorerna += och -=

```
<?php
$a += 5; // Samma sak som $a = $a + 5
$a -= 5; // Samma sak som $a = $a - 5
?>
```

7.1.3. J<65533>mf<65533>relseoperatorer

J<65533>mf<65533>relseoperatorer arbetar p<65533> tal och returnerar alltid ett v<65533>rde av typen boolean. Det vill s<65533>ga true eller false.

Tabell 7-2. J<65533>mf<65533>relseoperatorer

Exempel	Namn	Resultat
$a == $b	Lika med	Sant om $a <65533>r lika med $b.
$a != $b	Inte lika med	Sant om $a inte <65533>r lika med $b.
$a < $b	Mindre <65533>n	Sant om $a <65533>r mindre <65533>n $b.
$a > $b	St<65533>rre <65533>n	Sant om $a <65533>r st<65533>rre <65533>n $b.
$a <= $b	Mindre <65533>n eller lika med	Sant om $a <65533>r mindre <65533>n eller lika med $b.
$a >= $b	St<65533>rre <65533>n eller lika med	Sant om $a <65533>r st<65533>rre <65533>n eller lika med $b.
$a === $b	Identiska	Sant om $a <65533>r lika med $b och b<65533>da <65533>r av samma typ.

Exempel	Namn	Resultat
$a !== $b	Inte identiska	Sant om $a inte <65533>r lika med $b eller om de inte <65533>r av samma typ.

7.1.4. Logiska operatorer

Som jag sade ovan s<65533> returnerar alltid de j<65533>mf<65533>rande operatorerna av typen boolean och opererar p<65533> tal. Logiska operatorer returnerar alltid boolean, men opererar ocks<65533> bara p<65533> logiska termer.

Tabell 7-3. Logiska operatorer

Exempel	Namn	Resultat
$a and $b	Och	Sant om $a och $b <65533>r sanna.
$a or $b	Eller	Sant om $a eller $b <65533>r sanna.
$a xor $b	Exklusivt eller	Sant om $a eller $b <65533>r sanna men inte b<65533>da tv<65533>.
!$a	Inte/Icke	Sant om $a inte <65533>r sant.

7.1.5. Str<65533>ngoperatorer

Det finns tre operatorer som opererar p<65533> str<65533>ngar, den f<65533>rsta k<65533>nner ni till sedan f<65533>rut och det <65533>r tilldelningsoperatorn "tilldelas". Den fungerar lika p<65533> str<65533>ngar som p<65533> tal. Sedan finns det tv<65533> till. Dessa beskrivs l<65533>ttast med ett exempel:

Exempel 7-3. Str<65533>ngoperatorer

```
<?php
$s = "Kalle ";    //$a tilldelas "Kalle "
$t = $a . "Anka"; //$t inneh<65533>ller nu "Kalle Anka"
$s = "Kalle ";
$s .= "Anka";     //Samma sak som $s = $s . "Anka"
?>
```

7.2. Mer l<65533>sning

Det st<65533>r mycket om operatorer i PHP-Manualen (
http://www.php.net/manual/en/language.operators.php
(http://www.php.net/manual/en/language.operators.php)).

Kapitel 8. Selektioner (Villkorssatser)

I de allra flesta programmeringsspr<65533>k finns det selektionssatster. Precis som namnet antyder handlar det om val. Programmet kan ta olika v<65533>gar beroende p<65533> olika villkor. Vanliga selektionssatser <65533>r if-satsen och if-else-satsen.

8.1. If-satsen

If-satser finns i de allra flesta spr<65533>k och ser n<65533>stan likadan ut i dem alla. If-satsen fungerar som s<65533> att om n<65533>got <65533>r sant s<65533> g<65533>r en sak, annars inte. Ett exempel <65533>r p<65533> sin plats.

Antag att jag vill att ett program skall tala om f<65533>r mig om ett tal <65533>r st<65533>rre <65533>n 100. Jag vill att programmet skall skriva ut det tal jag anger och om det <65533>r st<65533>rre <65533>n 100 s<65533> skall det ocks<65533> skrivas ut. S<65533> h<65533>r kan det se ut:

Exempel 8-1. St<65533>rre <65533>n 100

```
<html>
<head>
   <title>St<65533>rre <65533>n 100</title>
</head>
<body>
   <?php
      echo "Du angav tal: $tal";

      if ($tal > 100) {
         echo "<p><emphasis>$tal <65533>r st<65533>rre <65533>n 100</emphasis></p>";
      }
   ?>
</body>
</html>
```

If-satsen best<65533>r allts<65533> av ordet if f<65533>ljt av ett test inom parenteser. Satsen som f<65533>ljer efter utf<65533>rs om testet blir sant. Vill man att det skall vara flera satser som utf<65533>rs om testet blir sant kan man sl<65533> ihop dem till ett block med hj<65533>lp av { och } (m<65533>svingar). I exemplet ovan anv<65533>nder jag m<65533>svingarna fast de egentligen inte beh<65533>vs. Som regel <65533>r det b<65533>st att alltid s<65533>tta dit m<65533>svingarna ifall man vill stoppa in en rad till inom if-satsen sedan s<65533> gl<65533>mmer man dem inte.

Studera nu if-satsen ovan och skriv om programmet och testa olika tal.

8.2. If-else-satsen

Nu fungerar programmet s<65533> l<65533>ngt. Men det vore ju kul om programmet sade till <65533>ven om talet inte <65533>r st<65533>rre <65533>n 100. Allts<65533> om det <65533>r st<65533>rre <65533>n 100 skriv det annars skriv att det inte <65533>r st<65533>rre <65533>n 100.

Studera f<65533>ljande exempel

Exempel 8-2. If-else exempel

```
<html>
<head>
   <title>St<65533>rre <65533>n 100</title>
</head>
<body>
<?php
echo "Du angav tal: $tal";

if ($tal > 100) {
   echo "<p><emphasis>$tal <65533>r st<65533>rre <65533>n 100</emphasis></p>";
} else {
   echo "<p><emphasis>$tal <65533>r inte st<65533>rre <65533>n 100</emphasis></p>";
}
?>
</body>
</html>
```

I exemplet ovan ser vi hur en if-else sats fungerar. Om uttrycket inom parenteserna <65533>r sant utf<65533>rs det som kommer efter. Om inte s<65533> utf<65533>rs det som kommer efter else. *Det kan aldrig intr<65533>ffa att b<65533>da satserna utf<65533>rs!*

8.3. if-elseif

If-elseif <65533>r ocks<65533> en vanlig konstruktion. Den anv<65533>nds ofta tillsammans med else och blir d<65533> en if-elseif-else sats. Man kan ha flera elseif i en konstruktion men bara en else. Det som kommer efter det f<65533>rsta sanna uttrycket utf<65533>rs och inget annat. Om inget <65533>r sant kommer det som st<65533>r efter else (annars) att utf<65533>ras.

Vi tittar p<65533> v<65533>rt exempel igen. Talet man anger kan ju vara st<65533>rre eller mindre <65533>n 100. <65533>r det inte det s<65533> m<65533>ste det ju vara talet 100 som angetts. Vi testar igen.

Exempel 8-3. if-elseif-else

```
<html>
<head>
   <title>St<65533>rre <65533>n 100</title>
</head>
<body>
<?php
echo "Du angav tal: $tal";

if ($tal > 100) {
   echo "<p><emphasis>$tal <65533>r st<65533>rre <65533>n 100</emphasis></p>";
} elseif ($tal < 100) {
   echo "<p><emphasis>$tal <65533>r inte st<65533>rre <65533>n 100</emphasis></p>";
} else {
   echo "<p><emphasis>$tal <65533>r ju lika med 100</emphasis></p>";
}
?>

</body>
</html>
```

Som vi ser s<65533> <65533>r den inte helt olik de andra konstruktionerna med if. Den fungerar s<65533> att om det f<65533>rsta testet <65533>r sant s<65533> utf<65533>rs satsen efter det. Om inte g<65533>rs testen efter elseif. <65533>r denna sann s<65533> utf<65533>rs satsen efter den. Om inget test har varit sant utf<65533>rs det som kommer efter else. Det kan vara flera elseif och else kan utel<65533>mnas.

8.4. Mer l<65533>sning

Aktuellt avsnitt i PHP-manualen. *http://www.php.net/manual/en/control-structures.php* (http://www.php.net/manual/en/control-structures.php)

Kapitel 9. Iterationer (Upprepningar, loopar)

Datorprogram <65533>r extremt bra p<65533> att g<65533>ra saker om och om igen, utan att ledsna eller g<65533>ra fel. Till detta anv<65533>nder man n<65533>gon typ av iterationssats (iteration = upprepning).

9.1. While-satsen

While-satsen <65533>r en vanlig iterationssats. Den fungerar s<65533> att en sats (som kan vara ett block) k<65533>rs om och om igen s<65533> l<65533>nge som ett test <65533>r sant. Studera f<65533>ljande exempel som skriver ut tiotusen ettor.

Exempel 9-1. Exempel med while

```
<?php
// Exempel p<65533> while-loop

echo "<h1>Tiotusen ettor</h1>";

$a = 0;
while ($a < 10000) {
   echo "1 ";
   $a = $a + 1;
}
?>
```

Vi tittar p<65533> exemplet rad f<65533>r rad. Raden som b<65533>rjar med "//" <65533>r en kommentar, den kan ni ignorera, kommentarer <65533>r viktiga men vi kommer att g<65533> igenom dem lite senare.

Den andra raden <65533>r starten p<65533> while-loopen (kallas <65533>ven loop eftersom den loopar om och om igen). S<65533> l<65533>nge som testet ($a < 10000) <65533>r sant so kommer satsen efter att repeteras. N<65533>r $a <65533>r st<65533>rre <65533>n eller lika med 10000 kommer loopen att avbryts. Om man i loopen gl<65533>mmer att <65533>ka $a kommer testet alltid att vara sant och man kommer aldrig ur loopen. Detta kallas f<65533>r en o<65533>ndlig loop och <65533>r ett vanligt programmeringsfel som g<65533>r att programmet h<65533>nger sig eller kraschar.

Om testet inte <65533>r sant fr<65533>n b<65533>rjan s<65533> kommer aldrig det st<65533>r i satsen att k<65533>ras. Se d<65533>rf<65533>r till att testet <65533>r sant fr<65533>n b<65533>rjan.

9.2. do-while-satsen

Do-while liknar p<65533> m<65533>nga s<65533>tt den vanliga while-satsen. Den enda skillnaden <65533>r att det som st<65533>r i satsen alltid kommer att utf<65533>ras <65533>tminstone en g<65533>ng. Se f<65533>ljande exempel:

Exempel 9-2. Exempel med do-while

```
<?php
// Ett exempel p<65533> hur man anv<65533>nder do-while

$i = 0;

do {
   echo "$i ";
   $i = $i + 1;
} while ($i < 100);

?>
```

Tilldelningen till $i <65533>r viktig eftersom den s<65533>tter v<65533>rdet d<65533>r loopen skall b<65533>rja. I loopen skrivs f<65533>rst $i ut och sedan <65533>kas variabeln $i med ett. Detta sker s<65533> l<65533>nge som $i <65533>r mindre <65533>n 100. Allts<65533> fr<65533>n 0 till 99.

9.3. for-loopen

For <65533>r den vanligaste iterationen. Den <65533>r dock vid en f<65533>rsta anblick lite kr<65533>ngligare <65533>n de andra. Man kan om man vill anv<65533>nda while ist<65533>llet f<65533>r for om man vill, men n<65533>r man kan for <65533>r den mycket smidigare.

for-loopen skriver man med det reserverade ordet for f<65533>ljt av en parentes. Inom parentesen skall det st<65533> tre stycken uttryck. Dessa tre skall se ut enligt f<65533>ljande.

- Det f<65533>rsta kommer att exekveras en g<65533>ng inan loopen b<65533>rjar.
- Det andra skall vara ett boolskt uttryck. Loopen kommer att g<65533> s<65533> l<65533>nge detta <65533>r sant.
- Det tredje k<65533>rs efter varje g<65533>ng som loopen har g<65533>tt.

Nu k<65533>nner jag att exempel <65533>r p<65533> sin plats igen:

Exempel 9-3. Exempel med for-loop

```
<?php
//Exempel p<65533> for-loop

for ($i = 0; $i <= 10; $i++) {
   echo "$i<br>\n";
}
?>
```

Oftast anv<65533>nds de tre olika satserna p<65533> precis det s<65533>tt som visas ovan. N<65533>mligen att initiera en r<65533>knare, kolla ett gr<65533>nsv<65533>rde och r<65533>kna upp r<65533>knaren, men inget hindrar att man anv<65533>nder dem p<65533> andra s<65533>tt.

9.4. Mer l<65533>sning

Aktuellt avsnitt i PHP-manualen: *http://www.php.net/manual/en/control-structures.php* (http://www.php.net/manual/en/control-structures.php)

Kapitel 10. Pseudokod

Kommer ...

10.1. Mer l<65533>sning

L<65533>nkar till mera l<65533>sning f<65533>r den intresserade

Kapitel 11. Indentering

Att indentera sin kod <65533>r n<65533>got man g<65533>r f<65533>r att den skall bli l<65533>ttare att l<65533>sa och l<65533>ttare att hitta fel. Detta avsnitt beskriver hur man indenterar p<65533> ett bra s<65533>tt.

11.1. Vad <65533>r indentering?

Indentering g<65533>r ut p<65533> att man med hj<65533>lp av olika mycket blanksteg (space) till v<65533>nster om koden kan p<65533> ett logiskt s<65533>tt gruppera koden s<65533> att den g<65533>r l<65533>ttare att l<65533>sa. Det finns flera olika s<65533>tt att intendera p<65533> och varje programmerare har sin egen stil. F<65533>r att kod skall bli enhetliga s<65533> har m<65533>nga f<65533>retag en kodstandard i vilken det beskrivs hur kommentering och indentering skall g<65533>ra inom f<65533>retaget. Det g<65533>r att alla programmerare k<65533>nner sig hemma i varandras kod och att den totala kodmassan blir enhetlig och l<65533>ttare att granska.

Grundprincipen <65533>r att kod som h<65533>nger ihop skall ha samma indenteringsniv<65533>. Se f<65533>ljande exemepel:

Exempel 11-1. Indentering

```
<?php
if ($tal == 100) {
   echo "Talet <65533>r 100";
   $tal = $tal + 1;
}
?>
```

I exemplet ser vi att det som h<65533>r till if-satsen har flyttats in en niv<65533>. Det g<65533>r det l<65533>tt att se att det h<65533>r till if-satsen och att m<65533>svingarna <65533>r riktiga. Vissa indenterar if-satsen s<65533> h<65533>r:

Exempel 11-2. Indentering

```
<?php
if ($tal == 100)
{
   echo "Talet <65533>r 100";
   $tal = $tal + 1;
}
```

```
?>
```

Om vi s<65533>ger att vi har n<65533>stlade if-satser s<65533> syns det <65533>nnu tydligare vad bra det <65533>r att indentera.

Exempel 11-3. Indentering

```
<?php
if ($inloggad) {
   if ($tal == 100) {
      echo "Tal <65533>r hundra";
   } else {
      echo "Tal <65533>r inte hundra";
   }
} else {
   echo "Du <65533>r inte inloggad!"
}
?>
```

11.2. Mer l<65533>sning

L<65533>nkar till mera l<65533>sning f<65533>r den intresserade

Svensk text med massor av exempel p<65533> indentering och kommentering, samt l<65533>nkar till mer +info p<65533> engelska: http://www.phpsidan.nu/res_articles.php?view=art&id=48

Kapitel 12. Kommentering

Alla som n<65533>gon g<65533>ng jobbat i ett programmeringsprojekt vet att det <65533>r av yttersta vikt att man kommenterar sin kod. Detta avsnitt beskriver hur man kommenterar och vad man skall t<65533>nka p<65533> n<65533>r man kommenterar sin kod.

12.1. Hur ser en kommentar ut

I PHP finns det tv<65533> typer av kommentarer. De <65533>r // *Kommentar* och /* *Kommentar* */. Den f<65533>rsta fungerar s<65533> att allt som kommer efter // och fram till radens slut <65533>r en kommentar och kommer att ignoreras av PHP-tolkaren. Den andra typen av kommentar fungerar s<65533> att det som st<65533>r mellan /* och */ <65533>r kommentarer. Den andra varianten kan str<65533>cka sig <65533>ver flera rader.

Exempel 12-1. Exempel med kommentering

```
<?php
/* Detta <65533>r en kommentar */
// Detta <65533>r en kommentar
$i = 1000; // Detta <65533>r ocks<65533> en kommetar
?>
```

12.2. Att kommentera sin kod

Att kommentera i sin kod <65533>r en konst. Det <65533>r mycket att t<65533>nka p<65533>. Det som <65533>r sv<65533>rast <65533>r att veta hur mycket man skall kommentera. Det <65533>r lika illa att kommentera f<65533>r mycket som f<65533>r lite. H<65533>r kommer n<65533>gra riktlinjer.

Skriv i kommentaren VAD som g<65533>rs och inte HUR det g<65533>rs. Hur det g<65533>rs skall koden i sig sj<65533>lv f<65533>rklara.

Kommentera i en sammanh<65533>ngande l<65533>ngre kommentar f<65533>re ett avancerat block vad som g<65533>rs. Ett litet exempel:

Exempel 12-2. L<65533>ngre kommentar f<65533>re block

```
<?php
//
// Nedanst<65533>ende r<65533>knar ut summan av alla tal mellan tal1 och
// tal2.
//
// Summan skrivs ut och tal2 m<65533>ste vara st<65533>rre <65533>n tal1
//

$summa = 0;
for ($i = $tal1; $i <= $tal2; $i++) {
   $summa = summa + $i;
}
echo $summa;
?>
```

J<65533>mf<65533>r detta med nedanst<65533>ende kod som <65533>r full av "Papegojkommentarer" (En papegoja brukar bara l<65533>ra sig att upprepa det den h<65533>r).

Exempel 12-3. Papegojkommentarer

```
<?php
$summa = 0; // Summan s<65533>tts till 0

for ($i = $tal1; $i <= $tal2; $i++) { // R<65533>kna upp i fr<65533>n $tal1 till $tal2
   $summa = summa + $i; // Aktuellt tal l<65533>ggs till summan
}
echo $summa; // Skriv ut summan
?>
```

Observera att det <65533>r sv<65533>rare att f<65533>rst<65533> vad den h<65533>r koden g<65533>r <65533>n den ovanf<65533>r. Trots att den <65533>r full av kommantarer. Den nedre har bara kommentarer som beskriver vad koden i sig beskriver och tillf<65533>r inget. Radkommentarer <65533>r till<65533>tet om de g<65533>r n<65533>gon nytta. Till exempel d<65533> variabler deklareras <65533>r det bra att ha radkommentarer efter varje variabel d<65533>r man beskriver vad man t<65533>nkt att variabeln skall g<65533>ra.

12.3. Liten sammanfattning

- Kommentera inte f<65533>r mycket och inte f<65533>r litet.

- Koden i sig skall visa vad programmet g<65533>r.
- Beskriv g<65533>rna i en (l<65533>ngre) kommantar f<65533>re en funktion eller avancerat block i en funktion vad det g<65533>r ist<65533>llet f<65533>r att kommantera p<65533> varje rad.
- Kommentarer p<65533> samma rad som koden blir l<65533>tt "Papegojkommentarer" s<65533>dana <65533>r fula och skall inte g<65533>ras.

Ett mer sammanh<65533>ngande exempel finns nedan:

Exempel 12-4. Kommentering sammanh<65533>ngade exempel

```
<?php
//
// kommentering.php
//
// Detta <65533>r ett litet skript som bara demonstrerar kommentering.
// I b<65533>rjan av varje fil <65533>r det v<65533>ldigt bra om man har ett block
// som detta d<65533>r det st<65533>r vad som finns i filen. Och hur man f<65533>r
// tag i programmeraren.
//
// Av: Marcus Rej<65533>s <marcus@rejas.se>
// Ver: 1.002
//

//
// F<65533>ljande visar hur anv<65533>ndarens browser presenterar sig. Det
// <65533>r bra att f<65533>re avancerade block eller funktioner i koden
// beskriva vad koden g<65533>r.
//

echo "Din browser presenterar sig som:<br>";
echo $HTTP_USER_AGENT;


//
// Man kan <65533>ven visa vilket IP de kommer fr<65533>n
//
echo "<p>Du har IP-nummer:<br>";
echo $REMOTE_ADDR;

//
// Skriver ut alla tal mellan 1 och 10
//
echo "<p>Alla tal mellan 1 och 10 ";
$tal = 1;
while ($tal <= 10) {
   echo "$tal ";
   $tal++;
}
```

```
// Nedan visas samma kod med "Papegojkommentarer"
echo "<p>Alla tal mellan 1 och 10 ";
$tal = 1;               // Tal tilldelas 1
while ($tal <= 10) {    // S<65533> l<65533>nge som tal <= 10
   echo "$tal ";       // Skriv ut tal
   $tal++;             // <65533>ka tal med ett
}

// Vilket g<65533>r l<65533>ttast att f<65533>rst<65533>?

// Exempel p<65533> en block-kommentar. Nedanst<65533>ende <65533>r helt
// bortkommenterat

/*
  Nedan visas samma kod med "Papegojkommentarer"
$tal = 1;               // Tal tilldelas 1
while ($tal <= 10) {    // S<65533> l<65533>nge som tal <= 10
   echo "$tal ";       // Skriv ut tal
   $tal++;             // <65533>ka tal med ett
}
*/

echo "<hr>Detta <65533>r bara ett skript som demonstrerar kommentering.
Titta p<65533> k<65533>llkoden ist<65533>llet.";

?>
```

12.4. Mer l<65533>sning

L<65533>nkar till mera l<65533>sning f<65533>r den intresserade

Svensk text med massor av exempel p<65533> indentering och kommentering, samt l<65533>nkar till mer info p<65533> engelska: http://www.phpsidan.nu/res_articles.php?view=art&id=48

Kapitel 13. Mer om str<65533>ngar

Str<65533>ngar <65533>r en typ som best<65533>r av f<65533>ljder av tecken. Till exempel s<65533> <65533>r "Jag heter Marcus" en str<65533>ng. I detta avsnitt tittar vi lite mer p<65533> vad man kan g<65533>ra med str<65533>ngar.

13.1. Vad <65533>r en str<65533>ng

En str<65533>ng <65533>r en grupp av tecken. Str<65533>ngar f<65533>rekommer, i stort sett, i alla program. PHP <65533>r ett spr<65533>k som <65533>r v<65533>ldigt rikt p<65533> funktioner f<65533>r att hantera str<65533>ngar. Mycket beroende p<65533> att det <65533>r ett spr<65533>k f<65533>r web-programmering d<65533>r i princip allt som kommer fr<65533>n programmet <65533>r str<65533>ngar. Det uppmuntrar <65533>ven till att man l<65533>ter ok<65533>nda anv<65533>ndare mata in str<65533>ngar till programmen vilket g<65533>r att man av s<65533>kerhetssk<65533>l m<65533>ste vara f<65533>rsiktig med str<65533>ngarna.

I PHP markeras en str<65533>ng av att den innesluts av enkla eller dubbla citationstecken. Skillnaden <65533>r den att inom dubbla citationstecken kommer alla variabler i str<65533>ngen att bytas ut mot sitt v<65533>rde. Se f<65533>ljande exempel:

Exempel 13-1. Exempel med str<65533>ngar

```
<?php
$summa = 1 + 6;
echo "Summan <65533>r $summa"; // Skriver ut: Summan <65533>r 7
echo 'Summan <65533>r $summa'; // Skriver ut: Summan <65533>r $summa
?>
```

Som du ser s<65533> sker ingen variabelsubstitution i den andra raden eftersom den omges av enkla citationstecken.

13.2. Escape-tecken

Som vi s<65533>g i f<65533>rra stycket s<65533> omges en str<65533>ng av citationstecken. En naturlig fr<65533>ga man d<65533> st<65533>ller sig <65533>r vad som h<65533>nder om jag vill ha

citationstecken i en str<65533>ng. Se f<65533>ljande exempel:

Exempel 13-2. Citationstecken i str<65533>ngar

```
<?php
echo "Tjenare din gamle "hacker"";
?>
```

Man ser direkt att det inte kommer att bli bra. Hur skall tolkaren kunna veta var str<65533>ngen slutar? Det som kommer att ske <65533>r att str<65533>ngen b<65533>rjar vid det f<65533>rsta citationstecknet och slutar vid den andra. Den bosktav (h) som kommer efter kommer att orsaka ett "parse error". Hur g<65533>r man d<65533>? Jo om man vill infoga specialtecken i en str<65533>ng m<65533>ste dessa f<65533>reg<65533>s av specialtecknet \ (bakv<65533>nt snedstreck eller backslash). Str<65533>ngen ovan blir d<65533>:

Exempel 13-3. Exempel p<65533> escape-tecken

```
<?php
echo "Tjenare din gamle \"hacker\"";
?>
```

Nu blir utskrifter som vi t<65533>nkt oss. Det finns <65533>ven andra specialtecken:

Tabell 13-1. Specialtecken i str<65533>ngar

Teckenkombination	Skrivs ut som
\"	"
\’	’
\\	\
\$	$
\n	Ny rad
\t	Tab

I str<65533>ngar inom enkla citationstecken (’) s<65533> substitueras bara "\’". Alla andra representerar sig sj<65533>lva.

13.3. L<65533>gga ihop str<65533>ngar

Man kan inte l<65533>gga ihop str<65533>ngar med hj<65533>lp av additionsoperatorn (+). Den <65533>r ju till f<65533>r aritmetiska termer. Som tur <65533>r s<65533> finns det speciella

opteratorer f<65533>r just str<65533>ngar. Den som l<65533>gger ihop tv<65533> str<65533>ngar kallas f<65533>r concatenationsoperatorn. Den representeras av tecknet "." (punkt). Se nedanst<65533>ende exempel.

Exempel 13-4. Exempel med str<65533>ngar

```
<?php
$a = "Hello ";
$b = $a . "World!"; // Vi l<65533>gger till str<65533>ngen "World!" efter $a
echo $b; // Skriver ut "Hello World!"
?>
```

13.4. J<65533>mf<65533>ra str<65533>ngar

I PHP kan man j<65533>mf<65533>ra str<65533>ngar med de operaorer som vi l<65533>rt oss f<65533>r numeriska v<65533>rden. Det <65533>r ganska specifikt f<65533>r PHP. R<65533>kna inte med att du kan g<65533>ra s<65533> i andra spr<65533>k du kommer i kontakt med. <65533>ven i PHP finns det funktioner f<65533>r att j<65533>mf<65533>ra str<65533>ngar.

13.4.1. Strcmp och strcasecmp

Strcmp (STRing CoMPare) <65533>r en funktion som j<65533>mf<65533>r tv<65533> str<65533>ngar med varandra. Om de <65533>r exakt likadana returneras v<65533>rdet 0. Om den f<65533>rsta <65533>r st<65533>rre returneras 1 och om den andra <65533>r st<65533>rre returneras -1. Syntaxen och ett exempel p<65533> hur den kan anv<65533>ndas visas i nedanst<65533>ende exempel.

Exempel 13-5. Exempel med strcmp

```
<?php
if (strcmp($password, "Hemligt") == 0) {
   echo "R<65533>tt l<65533>senord";
} else {
   echo "Fel l<65533>senord!";
}
?>
```

T<65533>nk p<65533> att strcmp g<65533>r skillnad p<65533> stora och sm<65533> bokst<65533>var. Texten "R<65533>tt l<65533>senord" ovan kommer bara att skrivas ut om $password

inneh<65533>ller exakt "Hemligt". Vill du j<65533>mf<65533>ra str<65533>ngen utan att versaler/gemener skall ha n<65533>gon betydelse kan du prova strcasecmp som fungerar p<65533> samma s<65533>tt fast "case insensitive".

13.5. <65533>ndra str<65533>ngar

Ofta vill man <65533>ndra p<65533> str<65533>ngar s<65533> att de ser lite annorlunda ut. Det kan vara att man vill g<65533>ra om alla bokst<65533>ver till versaler eller gemener. Eller att man vill byta n<65533>got ord mot ett annat. PHP har massor av funktioner f<65533>r detta. Vanliga saker man vill g<65533>ra med str<65533>ngar <65533>r att byta ut en f<65533>rekomst av ett ord mot ett annat eller att g<65533>ra alla tecken till sm<65533> eller stora bokst<65533>ver eller ta bort tomma tecken.

13.5.1. strtoupper och strtolower

Dessa tv<65533> funktioner (STRing TO UPPERcase och LOWERcase) tar en str<65533>ng som argument och returnerar samma str<65533>ng med alla bokst<65533>ver konverterade till antingen stora bokst<65533>ver (versaler) eller sm<65533> bokst<65533>ver (gemener).

Se f<65533>ljande lilla exempel:

Exempel 13-6. Exempel med strtoupper och strtolower

```
<?php
$str1 = "Kalle Anka";
$str2 = strtolower($str1);  // str2 blir "kalle anka"
$str3 = strtoupper($str1);  // str3 blir "KALLE ANKA"
echo "\$str1: ". $str1;
echo "\n\$str2: ". $str2;
echo "\n\$str3: ". $str3 ."\n";
?>
```

13.5.2. Ucfirst och ucwords

Dessa tv<65533> funtioner (UpperCase FIRST och WORDS) <65533>r kanske inte lika anv<65533>ndbara som strtoupper och strtolower men kan vara bra att k<65533>nna till. De kan anv<65533>ndas till exempel om man vill snygga till anv<65533>ndarinmatad data. Vad de g<65533>r <65533>r att g<65533>ra den f<65533>rsta bokstaven i en str<65533>ng (ucfirst) eller f<65533>rsta

bokstaven i varje ord i str<65533>ngen (ucwords) till versal. Observera att dessa funktioner bara verkar p<65533> de tecken det g<65533>ller och inte de <65533>vriga. Se f<65533>ljande exempel:

Exempel 13-7. Exempel med ucfirst och ucword

```
<?php
$str1 = "kalle anka";
$str2 = ucwords($str1); // str2 blir "Kalle Anka"

$str1 = 'KALLE ANKA';
$str2 = ucwords($str1); // str2 blir KALLE ANKA (inga tecken blir gemener)

$str3 = ucwords(strtolower($str1)); // str3 blir Kalle Anka
?>
```

I exemplet ser vi att "KALLE ANKA" blir samma sak efter ucwords. Det beror p<65533> att denna funktion bara verkar p<65533> det f<65533>rsta tecknet i varje ord. Detta g<65533>rs versalt. Inget annat g<65533>rs. <65533>r alla tecken versaler s<65533> kommer inget att ske. I den sista satsen l<65533>gger vi in ett anrop till strtolower vilket g<65533>r att ucwords matas med str<65533>ngen "kalle anka" ist<65533>llet.

Ucfirst fungerar p<65533> exakt samma s<65533>tt fast bara p<65533> det allra f<65533>rsta tecknet i str<65533>ngen.

13.5.3. strrev

Detta <65533>r en v<65533>ldigt trevlig lite funktion (STRing REVerse). Det enda den g<65533>r <65533>r att den tar en str<65533>ng som argument och returnerar samma str<65533>ng fast reverserad.

Exempel 13-8. Exempel med strrev

```
<?php
$str1 = "Kalle Anka";
$str2 = strrev($str1); // str2 blir "aknA ellaK"
?>
```

13.5.4. strlen

En betydligt mer anv<65533>ndbar funktion <65533>n strrev <65533>r strlen (STRing LENgth). Denna returnerar antalet tecken i en str<65533>ng.

Som vanligt tar vi ett litet exempel:

Exempel 13-9. Exemepel med strrev

```
<?php
$str1 = "Kalle Anka";
echo strlen($str1);          // skriver ut 10
echo strlen(" Kalle Anka "); // skriver ut 12
?>
```

13.5.5. str_replace

Detta <65533>r en mycket anv<65533>ndbar funktion. Den byter ut en teckenf<65533>ljd i en mening mot en annan. Funktionen tar tre argument och returnerar en str<65533>ng. Det f<65533>rsta argumentet <65533>r den teckenf<65533>ljd som skall ers<65533>ttas, det andra <65533>r det som det skall ers<65533>ttas med och det tredje argumentet <65533>r den str<65533>ng som det hela ber<65533>r. Det som returneras <65533>r str<65533>ngen i det tredje argumentet d<65533>r alla f<65533>rekomster av teckenf<65533>ljden i det f<65533>rsta argumentet ersatts med tecknen i det andra argumentet.

F<65533>rvirrad? Se nedanst<65533>ende exempel:

Exempel 13-10. Exempel med str_replace

```
<?php
$str1 = "Kalle Anka <65533>r b<65533>st";
$str2 = str_replace("Kalle", "Kajsa", $str1);

echo $str2;  // str2 <65533>r "Kajsa Anka <65533>r b<65533>st"
?>
```

Detta <65533>r anv<65533>ndbart till v<65533>ldigt mycket. Bara fantasin s<65533>tter gr<65533>nserna.

13.6. Hantera o<65533>nskad HTML i str<65533>ngar

I PHP hanterar man av naturliga sk<65533>l ofta str<65533>ngar som skall presenteras i en webl<65533>sare. Flera av dessa str<65533>ngar kommer vanligtvis fr<65533>n anv<65533>ndare eller andra os<65533>kra k<65533>llor. D<65533> vill man g<65533>rna kontrollera s<65533> att inte anv<65533>ndaren kan mata in data som f<65533>rst<65533>r resten av sidan. Till exempel s<65533> skall man inte i ett web-forum kunna skriva in HTML-kod hur som helst. Man skulle d<65533> kunna l<65533>nka in fula bilder eller andra typsnittet p<65533> hela sidan. Man kan <65533>stadkomma detta genom att anv<65533>nda massor av anrop p<65533> str_replace. Till exempel:

Exempel 13-11. Med str_replace

```
<?php
//
// F<65533>ljande tv<65533> rader tar bort alla f<65533>rekomster av st<65533>rre <65533>
// mindre <65533>n och ers<65533>tter dem med deras HTML-motsvarighet. Detta
// eliminerar alla HTML-taggar.
//
$html_string = str_replace("&lt;", "&amp;lt;", $html_string);
$html_string = str_replace("&gt;", "&amp;gt;", $html_string);
?>
```

Men det finns b<65533>ttre s<65533>tt att g<65533>ra det p<65533>. Det finns (naturligtvis) f<65533>rdiga funktioner som g<65533>r HTML av str<65533>ngar, till exempel htmlentities. htmlentities g<65533>r om en str<65533>ng s<65533> att alla tecken som har en motsvarighet i HTML kod blir just, HTML kod. Funktionen tar ett argument och har tv<65533> stycken valfria argument som du inte beh<65533>ver ange, mer <65533>n om du t.ex. beh<65533>ver anv<65533>nda en annan teckenupps<65533>ttning. Se f<65533>ljande exempel:

Exempel 13-12. Exempel med htmlentities

```
<?php
$str1 = "<h1>Kalle Anka</h1>"; // Blir "Kalle Anka" (Rubrik 1) i browsern
$str2 = htmlentities($str1);   // Blir <h1>Kalle Anka</h1>; i browsern
?>
```

Detta g<65533>r att allt man skickar till htmlentities kommer att synas i browsern precis som det ser ut i klartext.

13.7. S<65533>ka i str<65533>ngar

Ofta vill man hitta eller anv<65533>nda bara vissa delar av en str<65533>ng. Det finns flera funktioner f<65533>r detta. Vi skall titta p<65533> tv<65533> av dem.

13.7.1. strstr och stristr

Dessa tv<65533> funktioner fungerar s<65533> att de tar tv<65533> argument, b<65533>da <65533>r str<65533>ngar (STRing in STRing och STRing case-Insensitive in STRing). Den f<65533>rsta <65533>r den str<65533>ng som det skall s<65533>kas i och det andra <65533>r det s<65533>kta. Det som returneras <65533>r det som <65533>r kvar av det f<65533>rsta argumentet efter den funna str<65533>ngen. F<65533>rvirrad?

Exempel 13-13. Exemempel med strstr()

```
<?php
$namn = "Kalle Anka";
$enamn = strstr($namn, " ");    // enamn blir " Anka"

$adress = "kalle@ankeborg.net";
$domain = strstr($adress, "@"); // domain blir "@ankeborg.net"
?>
```

Funktionen stristr fungerar p<65533> samma s<65533>tt men den bryr sig inte om om den efters<65533>kta str<65533>ngen har stora eller sm<65533> bokst<65533>ver (Case-Insensitive).

13.8. Mer l<65533>sning

Aktuellt avsnitt i Manualen. *http://www.php.net/manual/en/ref.strings.php* (http://www.php.net/manual/en/ref.strings.php)

Kapitel 14. Funktioner

Funktioner anv<65533>nds f<65533>r att dela upp programmet i mindre sm<65533> delar. Det g<65533>r att programmeraren kan koncentrera sig p<65533> en del i taget och l<65533>per mindre risk att g<65533>ra fel. Man anv<65533>nder <65533>ven funktioner till kod som kan <65533>terfinnas p<65533> flera st<65533>llen f<65533>r att minska duplikation av kod.

14.1. Vad <65533>r en funktion?

En funktion <65533>r ett antal instruktioner som frist<65533>ende utf<65533>r en sak. Denna snutt kan sedan k<65533>ras om och om i samma program eller delas med andra program. Ett exempel:

Exempel 14-1. Funktioner

```
<?php
/*
 * print_html_header_start
 *
 * Skriver ut en html-header
 */
function print_html_header_start () {
   echo "<html><head></head><body>";
}

/*
 * print_html_header_stop
 *
 * Skriver ut slut-html-taggar
 */
function print_html_header_stop () {
   echo "</body></html>";
}


// H<65533>r b<65533>rjar programmet
print_html_header_start();

// Andra utskrifter h<65533>r.
print_html_header_stop();
?>
```

I exemplet ovan deklareras tv<65533> funktioner som skriver ut en html-header och en html-footer. Dessa funktioner anropas l<65533>ngre ned i programmet.

14.2. N<65533>r skall man anv<65533>nda funktioner?

Man skall anv<65533>nda funktioner s<65533> snart man kan. Sj<65533>lva programmet brukar ofta vara bara anrop till olika funktioner. F<65533>ljande kan vara bra grundregler.

Anv<65533>nd funktioner till:

- Alla uppgifter som kan avgr<65533>nsas
- Alla uppgifter som <65533>r repititiva
- All kod som du kan t<65533>nkas <65533>teranv<65533>nda i andra program

F<65533>rs<65533>k att t<65533>nka ett program i funktioner.

Vi funderar p<65533> programmet password.php och f<65533>rs<65533>ker identifiera olika funktioner.

En funktion <65533>r att skriva ut html-formul<65533>ret. Det <65533>r ju en avgr<65533>nsad uppgift. En annan solklar funktion <65533>r autentiseringen.

14.3. Argument till funktioner

En funktion kan ta noll eller flera argument. Ett argument <65533>r ett v<65533>rde som man skickar till funktionen. Till exempel om man vill vill g<65533>ra en funktion som summerar tv<65533> tal s<65533> <65533>r det bra om den kan ta de tv<65533> talen som argument. Se f<65533>ljande exempel:

Exempel 14-2. Argument till funktioner

```
<?php
function summa($tal1, $tal2) {
        return ($tal1 + $tal2);
}

echo summa(5,6);
?>
```

Observera att ordningen p<65533> argumenten spelar roll. I exemplet kommer tal1 att bli 5 och tal2 6. Variablerna $tal1 och $tal2 existerar bara inom funktionen och inte i resten av programmet. Vilken som f<65533>r vilket v<65533>rde best<65533>ms av ordningen i funktionsanropet.

14.4. Returv<65533>rden

I funderingen ovan vore det ju bra om autentiseringen kunde returnera ett v<65533>rde (ett boolskt v<65533>rde till exempel). Det kan se ut s<65533> h<65533>r:

Exempel 14-3. Retur fr<65533>n funktioner

```
<?php
function is_logged_in ($name, $pass) {
        if ($name == "kalle") && ($pass == "ankeborg")) {
                return true;
        } else {
                return false;
        }
}
?>
```

Denna funktion <65533>r ganska kompakt och g<65533>r sig f<65533>tj<65533>nt av en f<65533>rklaring. Den tar tv<65533> argument, $namn och $pass. Funktionen returnerar v<65533>rdet av en boolsk operation.

14.5. Mer l<65533>sning

Aktuellt avsnitt i manualen. *http://www.php.net/manual/en/functions.php* (http://www.php.net/manual/en/functions.php)

Kapitel 15. Filer och filhantering

N<65533>r man skriver datorprogram vill man ofta spara information mellan olika k<65533>rningar program. Ett s<65533>tt att g<65533>ra detta <65533>r att lagra informationen i en eller flera filer. I detta kapitel behandlas hur man g<65533>r.

15.1. Filer

Innan vi b<65533>rjar anv<65533>nda oss av filer i programmeringen skall vi titta lite kort p<65533> vad en fil <65533>r. En fil <65533>r en samling sammanh<65533>ngande information p<65533> ett medium, oftast en h<65533>rddisk, som man namngett.

F<65533>r att l<65533>sa eller skriva i filen anv<65533>nder man ett filhandtag (eng. file handle). N<65533>r man programmerar kan man inte hoppa runt i filen hur som helst lika enkelt som man g<65533>r i till exempel ett ordbehandlingsprogram.

15.2. Arbetsmetod vid arbete med filer

N<65533>r man jobbar med filer i PHP anv<65533>nder man f<65533>ljande metodik.

Man <65533>ppnar en fil med ett anrop till funktionen *fopen* . Den funktionen returnerar ett filhandtag. Detta filhandtag kan man sedan anv<65533>nda f<65533>r att skriva till eller l<65533>sa fr<65533>n filen. Slutligen skall man st<65533>nga sin fil med funktionen *fclose*. N<65533>r man har en fil <65533>ppen finns det ocks<65533>, <65533>ven om man inte m<65533>rker det s<65533> ofta, en s<65533> kallad filpekare som h<65533>ller reda p<65533> var i filen man <65533>r.

15.3. Funktionen fopen

Funktionen fopen tar tv<65533> argument och returnerar ett filhandtag. S<65533> h<65533>r kan ett anrop till fopen se ut:

Exempel 15-1. Funktionen fopen

```
<?php
$filhandtag = fopen("/home/rejas/data/testfil", "a");
?>
```

Vad som sker <65533>r att $filhandtag tilldelas ett handtag till filen testfil i katalogen /home/rejas/data/. Observera att detta <65533>r s<65533>kv<65533>gen till filen p<65533> servern och att den anv<65533>ndare som k<65533>r webservern m<65533>sta ha r<65533>tt att l<65533>sa och eventuellt skriva till filen. Det f<65533>rsta argumentet <65533>r s<65533>ledes filnamnet, men vad <65533>r det andra? Jo det andra talar om p<65533> vilket s<65533>tt filen skall <65533>ppnas. F<65533>ljande <65533>r de vanligaste v<65533>rdena p<65533> det andra argumentet och deras inneb<65533>rder:

Tabell 15-1. Andra argumentet till fopen

V<65533>rde	Inneb<65533>rd
r	<65533>ppnar en fil endast f<65533>r l<65533>sning, filpekaren placeras f<65533>rst i filen.
r+	<65533>ppnar en fil f<65533>r l<65533>sning och skrivning, filpekaren placeras i b<65533>rjan av filen.
w	<65533>ppnar en fil endast f<65533>r skrivning, filpekaren st<65533>lls st<65533>lls f<65533>rst i filen. Om filen inte finns skapas den och om den finns s<65533> blir den <65533>verskriven.
w+	Samma som w men <65533>ppnar <65533>ven f<65533>r l<65533>sning
a	<65533>ppnar endast f<65533>r skrivning. Skapar filen om den inte finns. St<65533>ller filpekaren i slutet av filen.
a+	Samma som a men <65533>ven f<65533>r l<65533>sning.
x	<65533>ppnar en fil f<65533>r skrivning och placerar filpekaren i b<65533>rjan av filen. Om filen redan existerar returnerar funktionen FALSE och ett varnings meddelande kan komma att skrivas ut. Annars skapas filen.
x+	Samma som x men till<65533>ter <65533>ven l<65533>sning av filen.

Argumenten med + till fopen verkar bra att anv<65533>nda men anv<65533>nds faktiskt inte s<65533> ofta som man kan tro.

15.4. Funktionen fwrite

Funktionen fwrite anv<65533>nda f<65533>r att skriva till en fil. Den tar tv<65533> argument, ett

filhandtag och s<65533> det som skall skrivas till filen.

Ett exmpel:

Exempel 15-2. Funktionen fwrite

```
<?php
$fh = fopen("/home/rejas/data/testfil", "a");
fwrite($fh, "Hej p<65533> dig\n");
fclose($fh);
?>
```

Funktionen fwrite returnerar false om det skulle vara s<65533> att den inte kan skriva till filen s<65533> det kan vara bra att kolla att det g<65533>r bra.

Exempel 15-3. Funktionen fwrite med felkontrol

```
<?php
$fh = fopen("/home/rejas/data/testfil", "a");
if (! fwrite($fh, "Hej p<65533> dig\n")) {
   echo "Ooops, fel vid skrivning till fil";
   exit; // Avbryter k<65533>rningen
}

fclose($fh);
?>
```

<65533>ven fopen returnerar false om den inte kan <65533>ppna filen, att kolla detta l<65533>mnas som <65533>vning till l<65533>saren :).

15.5. Readfile

Readfile <65533>r en av m<65533>nga funktioner som kan anv<65533>ndas f<65533>r att l<65533>sa fr<65533>n en fil. Den l<65533>ser en hel fil och skriver ut den p<65533> utskiftsbufferten.

Se f<65533>ljande exempel:

Exempel 15-4. Exempel med readfile

```
<?php
readfile("/home/rejas/data/testfil");
?>
```

15.6. Sammanh<65533>ngande exempel

Nu kan vi skriva ett litet program som varje g<65533>ng det k<65533>rs l<65533>gger till en rad i en fil och skriver ut filen.

Exempel 15-5. Sammanh<65533>ngande exempel p<65533> filanv<65533>ndning

```
<?php
$filename = "/home/rejas/data/testfil";

$fh = fopen($filename, "a");

if (! fwrite($fh, "Hej p<65533> dig\n")) {
   echo "Ooops, fel vid skrivning till fil";
   exit; // Avbryter k<65533>rningen
}

fclose($fh);

readfile($filename);
?>
```

15.7. Mer l<65533>sning

Mer information om funktioner f<65533>r att hantera filer och filsystem finns h<65533>r:
http://www.php.net/manual/en/ref.filesystem.php (http://se.php.net/manual/en/ref.filesystem.php)

Kapitel 16. Arrayer (vektorer)

Vi har tidigare tittat p<65533> variabler. Nu skall vi titta p<65533> en speciell typ av variabler n<65533>mligen arrayer. En array <65533>r en variabel som kan inneh<65533>lla flera olika v<65533>rden. En array kallas iband <65533>ven f<65533>r en vektor.

16.1. Arrayer

Ibland vill man spara flera v<65533>rden i en variabel. Till exempel om de m<65533>ste h<65533>ra ihop och inte f<65533>r komma is<65533>r eller om man vill returnera flera v<65533>rlden fr<65533>n en funktion. Det <65533>r precis var arrayer <65533>r, en typ av variabel som kan h<65533>lla flera v<65533>rden.

Vi kastar oss direkt in p<65533> ett exempel:

Exempel 16-1. Arrayer

```
<?php
// Vi skapar en array som heter arr och inneh<65533>ller tre v<65533>rden.
$arr[0] = 10;
$arr[1] = 20;
$arr[2] = 30;

echo $arr[2]; // Skriver ut 30
?>
```

16.2. Array-funktioner

En annan f<65533>rdel med arrayer <65533>r att det finns massor av funktioner som verkar p<65533> arrayer. Till exempel s<65533> finns det funktioner f<65533>r att sortera, reversera eller blanda arrayer.

Vi skall titta p<65533> ett par anv<65533>ndbara funktioner:

16.2.1. Funktionen array_reverse

Den h<65533>r funktionen tar en array som argument och returnerar samma array fast i omv<65533>nd ordning. Det vill s<65533>ga det som var sist i arrayen tidigare liger f<65533>rst i den array som array_reverse returnerar. Ett litet exempel:

Exempel 16-2. Exempel med array_reverse()

```
<?php
$arr[0] = 1;
$arr[1] = 2;
$arr[2] = 3;

$arr2 = array_reverse($arr);

echo $arr2[0] ."\n"; // Skriver ut 3
echo $arr2[1] ."\n"; // Skriver ut 2
echo $arr2[2] ."\n"; // Skriver ut 1
?>
```

16.2.2. Funktionen sort

Funktionen sort tar en array som argument och sorterar den. Observera att denna funktion inte returnerar n<65533>got utan sorterar den array den f<65533>r som argument.

Exempel 16-3. Exempel med sort()

```
$arr2[0] = "Kalle";
$arr2[1] = "Fnatte";
$arr2[2] = "Knatte";
$arr2[3] = "Kajsa";
$arr2[4] = "Joakim";
$arr2[5] = "Alexander";
$arr2[6] = "Tjatte";


sort($arr2); // Observera att inget returneras, utan att arrayen blir sorterad.

for ($i = 0; $i <= 6; $i++) {
        echo "$arr2[$i]<br>";
}
```

16.2.3. Funktionen file

File <65533>r en funktion som kanske b<65533>ttre pasar bland filfunktionerna (d<65533>r finns den till exempel i PHP-manualen) men jag har valt att l<65533>gga den h<65533>r eftersom den inbjuder till att anv<65533>nda andra array-funktioner.

File tar ett argument som skall vara ett filnamn, den returnerar en array med varje rad i filen i ett element.

16.3. Mer l<65533>sning

http://www.php.net/manual/en/language.types.array.php http://www.php.net/manual/en/ref.array.php

Kapitel 17. Mer om funktioner

Tidigare har vi l<65533>rt oss hur funktioner fungerar. I detta kapitel l<65533>r vi oss lite mer om dem.

17.1. Call by reference, call by value

N<65533>r man anropar en funktion skickar man vanligtvis med ett eller flera argument. Vi har tidigare sett att om man <65533>ndrar dessa argument i funktionen s<65533> <65533>ndras de inte utanf<65533>r funktionen. Det <65533>r f<65533>r att man l<65533>ttare skall kunna flytta funktionen mellan olika program utan att beh<65533>va vara r<65533>dd f<65533>r att den skall <65533>ndra n<65533>got utanf<65533>r sj<65533>lva funktionen. Detta kallas f<65533>r *call by value*. Det vill s<65533>ga att v<65533>rdet i variabeln skickas till funktionen, inte sj<65533>lva variabeln.

Ibland kan man vilja l<65533>ta funktionen <65533>ndra variabler <65533>ven i v<65533>rlden utanf<65533>r. Allts<65533> om jag skickar en variabel till en funktion och den <65533>ndras i funktionen s<65533> skall den <65533>ndras <65533>ven utanf<65533>r funktionen. Detta f<65533>rfarande kallas f<65533>r *call by reference*. Det vill s<65533>ga att man ist<65533>llet f<65533>r att skicka en kopia p<65533> v<65533>rdet i en variabel s<65533> skickar man en pekare till variablen s<65533> att det man g<65533>r men variabeln i funktionen <65533>ven h<65533>nder utanf<65533>r funktionen.

Detta kan i flera fall vara v<65533>ldigt smidigt men skall normalt undvikas. Ett litet exempel.

Exempel 17-1. Call by reference

```
<?php
function swap (&$var1, &$var2) {
   $tmp = $var2;
   $var2 = $var1;
   $var1 = $tmp;
}

$text1 = "Text1";
$text2 = "Text2";

echo "F<65533>re swap: text1: $text1, text2: $text2 <br>\n";

swap($text1, $text2);

echo "Efter swap: text1: $text1, text2: $text2 <br>\n";
?>
```

Kapitel 18. Inmatning

Kommer ...

Appendix A. Kurs DTR1207 - Programmering A

50 po<65533>ng, inr<65533>ttad 2000-07 SKOLFS: 2000:28

A.1. M<65533>l

A.1.1. M<65533>l f<65533>r kursen

Kursen skall ge grundl<65533>ggande teoretiska och praktiska kunskaper i programmering. Kursen skall <65533>ven ge kunskaper om vanliga anv<65533>ndningsomr<65533>den f<65533>r olika programmeringsspr<65533>k. Kursen skall ocks<65533> ge grundl<65533>ggande f<65533>rdigheter i systemering och struktureringsteknik.

A.1.2. M<65533>l som eleverna skall ha uppn<65533>tt efter avslutad kurs

Eleven skall

kunna n<65533>got programmeringsspr<65533>ks grundl<65533>ggande datatyper, f<65533>rdefinierade strukturer och funktioner samt deras regler och syntax

kunna analysera programmeringsuppgifter och formulera strukturerad pseudokod samt konstruera enkla algoritmer

kunna systemera och strukturera programmeringsarbetet samt skriva enkla program och fels<65533>ka k<65533>llkod

k<65533>nna till kompilatorns/l<65533>nkarens arbete fr<65533>n k<65533>llkod till f<65533>rdigt program

k<65533>nna till viktiga operativsystemstandarder f<65533>r bl.a. teckenkoder och utmatningsrutiner

k<65533>nna till spr<65533>kens allm<65533>nna prestanda och egenskaper samt vilka programmeringsuppgifter de <65533>r b<65533>st l<65533>mpade f<65533>r.

A.2. Betygskriterier

A.2.1. Kriterier f<65533>r betyget Godk<65533>nd

Eleven analyserar enkla programmeringsuppgifter och skapar med viss handledning k<65533>rbara v<65533>ldokumenterade program.

Eleven s<65533>ker med viss handledning upp de fakta som beh<65533>vs f<65533>r programmeringsuppgifterna.

Eleven beskriver det anv<65533>nda programspr<65533>kets uppbyggnad, viktigaste funktioner, egenskaper och prestanda.

A.2.2. Kriterier f<65533>r betyget V<65533>l godk<65533>nd

Eleven utf<65533>r sina programmeringsuppgifter p<65533> egen hand och inom rimlig tid.

Eleven h<65533>mtar p<65533> egen hand fakta fr<65533>n olika k<65533>llor och till<65533>mpar dessa i uppgifterna.

A.2.3. Kriterier f<65533>r betyget Mycket v<65533>l godk<65533>nd

Eleven utf<65533>r sj<65533>lvst<65533>ndigt sina programmeringsuppgifter med noggrannhet och n<65533>r snabbt avsett resultat.

Eleven anpassar sin arbetsinsats till situationen, analyserar resultat samt <65533>tg<65533>rdar kvalitetsavvikelser.

Eleven beskriver samband och ser helheter i komplicerade programmeringssituationer.

Skolverket 2002-04-09

Appendix B. Kurs DTR1208 - Programmering B

50 po<65533>ng, inr<65533>ttad 2000-07 SKOLFS: 2000:28

B.1. M<65533>l

B.1.1. M<65533>l f<65533>r kursen

Kursen skall ge f<65533>rdjupade teoretiska och praktiska kunskaper i ett strukturerat programmeringsspr<65533>k. Kursen skall ocks<65533> ge kunskaper om spr<65533>kets viktigaste datastrukturer. Dessutom skall kursen ge f<65533>rdigheter i algoritmkonstruktion.

B.1.2. M<65533>l som eleverna skall ha uppn<65533>tt efter avslutad kurs

Eleven skall

f<65533>rst<65533> och kunna anv<65533>nda spr<65533>kets vanliga datastrukturer s<65533>som f<65533>lt, lista, stack och filsystem

kunna inf<65533>ra och anv<65533>nda array, l<65533>nkade listor och tr<65533>dstrukturer i datastrukturer

f<65533>rst<65533> och implementera vanliga sorteringsalgoritmer

f<65533>rst<65533> och implementera vanliga s<65533>kalgoritmer

kunna analysera programmeringsuppgifter och formulera strukturerad pseudokod

kunna skriva program och fels<65533>ka k<65533>llkod

k<65533>nna till kompilatorns/l<65533>nkarens uppgift vid arbete fr<65533>n k<65533>llkod till f<65533>rdigt program

k<65533>nna till viktiga operativsystemstandarder f<65533>r bl.a. teckenkoder och utmatningsrutiner

k<65533>nna till spr<65533>kets allm<65533>nna prestanda och egenskaper samt vilka programmeringsuppgifter det <65533>r l<65533>mpligast f<65533>r.

B.2. Betygskriterier

B.2.1. Kriterier f<65533>r betyget Godk<65533>nd

Eleven analyserar programmeringsuppgifter och skapar med viss handledning enkla k<65533>rbara v<65533>ldokumenterade program.

Eleven s<65533>ker med viss handledning upp de fakta som beh<65533>vs f<65533>r programmeringsuppgifterna.

Eleven beskriver det anv<65533>nda programspr<65533>kets uppbyggnad, viktigaste funktioner, egenskaper och prestanda.

B.2.2. Kriterier f<65533>r betyget V<65533>l godk<65533>nd

Eleven utf<65533>r sina programmeringsuppgifter p<65533> egen hand och inom rimlig tid.

Eleven h<65533>mtar p<65533> egen hand fakta fr<65533>n olika k<65533>llor och till<65533>mpar dessa i uppgifterna.

B.2.3. Kriterier f<65533>r betyget Mycket v<65533>l godk<65533>nd

Eleven utf<65533>r sj<65533>lvst<65533>ndigt sina programmeringsuppgifter med noggrannhet och n<65533>r snabbt avsett resultat.

Eleven anpassar sin arbetsinsats till situationen, analyserar resultat samt <65533>tg<65533>rdar kvalitetsavvikelser.

Eleven beskriver samband och ser helheter i komplicerade programmeringssituationer

B.3. Specialisering mot enligt lista nedan. F<65533>r betyg mm. anv<65533>nd angivna koder.

Kod	Namn
PCOB1408	Cobol
PDEL1408	Delphi
PJAV1408	Java
PPAS1408	Pascal
PPER1408	Perl
PPHP1408	PHP
PROC1408	C++
PVBA1408	Visual Basic

Skolverket 2002-04-09

Appendix C. Reserverade ord i PHP

Reserverade ord har en speciell inneb<65533>rd i PHP. Du kan inte anv<65533>nda dem som namn p<65533> konstanter, namn p<65533> klasser eller namn p<65533> funktioner. Du kan naturligtvis anv<65533>nda dem i str<65533>ngar.

De reserverade orden <65533>r: *and, array() , as , break , case, cfunction , __CLASS__, class, const, continue, declare, default, die(), do, echo(), else, elseif, empty(), enddeclare, endfor, endforeach, endif, endswitch, endwhile, eval, exception, exit(), extends, __FILE__ , for, foreach, function, __FUNCTION__, global, if, include(), include_once(), isset(), __LINE__ , list(), __METHOD__, new, old_function, or , php_user_filter, print(), require(), require_once(), return(), static, switch, unset(), use, var, while, xor*

Appendix D. GNU Free Documentation License

Version 1.2, November 2002. Svensk <65533>vers<65533>ttning av Marcus Rej<65533>s och Alexander Nordstr<65533>m, Januari 2004.

This is an unofficial translation of the GNU Free Documentation License into Swedish. It was not published by the Free Software Foundation, and does not legally state the distribution terms for documentation that uses the GNU FDL -- only the original English text of the GNU FDL does that. However, we hope that this translation will help Swedish speakers understand the GNU FDL better.

Detta <65533>r en inofficiell <65533>vers<65533>ttning av GNU Free Documentation License till svenska. Den har inte publicerats av Free Software Foundation och <65533>r inte juridiskt g<65533>llande f<65533>r spridning av dokumentation som anv<65533>nder GNU FDL -- bara den engelska originaltexten i GNU FDL g<65533>ller. Vi hoppas att denna <65533>vers<65533>ttning skall hj<65533>lpa svensktalande att f<65533>rst<65533> GNU FDL b<65533>ttre.

D.1. 0. BAKGRUND

Syftet med denna licens <65533>r att g<65533>ra en handbok, bok, eller annat praktiskt och anv<65533>ndbart dokument "fritt" som i frihet: att f<65533>rs<65533>kra var och en den faktiska friheten att kopiera och sprida det vidare, med eller utan f<65533>r<65533>ndringar, antingen kommersiellt eller ideellt. Sekund<65533>rt bevarar denna licens ett s<65533>tt f<65533>r f<65533>rfattaren och f<65533>rl<65533>ggaren att f<65533> <65533>ra f<65533>r deras arbete utan att de anses vara ansvariga f<65533>r f<65533>r<65533>ndringar gjorda av andra.

Denna Licens <65533>r en sorts "copyleft", vilket betyder att derivativa verk av detta dokument sj<65533>lva m<65533>ste vara fria p<65533> samma s<65533>tt. Den kompletterar GNU General Public License, som <65533>r en copyleft-licens utformad f<65533>r fri programvara.

Vi har utformat denna licens f<65533>r att den skall anv<65533>ndas f<65533>r handb<65533>cker till fri programvara, eftersom fri programvara beh<65533>ver fri dokumentation: ett fritt program b<65533>r ha en handbok som erbjuder samma friheter som programmet g<65533>r. Men denna licens <65533>r inte begr<65533>nsad till programvaruhandb<65533>cker; den kan anv<65533>ndas f<65533>r vilket textverk som helst oavsett <65533>mne eller huruvida det <65533>r en utgiven, tryckt bok. Vi rekommenderar denna licens huvudsakligen f<65533>r alla verk vars syfte <65533>r instruktion eller referens.

D.2. 1. TILL<65533>MPNINGSOMR<65533>DE OCH DEFINITIONER

Denna licens [det engelska originalet] g<65533>ller f<65533>r varje handbok eller annat verk, oavsett uttrycksform, som inneh<65533>ller ett meddelande d<65533>r upphovsr<65533>ttsinnehavaren stadgat att verket kan spridas enligt villkoren i GNU Free Documentation License. Ett s<65533>dant meddelande ger en internationell frihet utan krav p<65533> ers<65533>ttning och utan tidsbegr<65533>nsning att anv<65533>nda verket under villkoren i denna licens [det engelska originalet]. "Dokument" nedan syftar p<65533> godtycklig handbok eller verk. Var och en <65533>r licenstagare och ben<65533>mns som "du". Du accepterar villkoren i GNU Free Documentation License om du kopierar, modifierar eller sprider verket p<65533> ett s<65533>dant s<65533>tt att det kr<65533>ver tillst<65533>nd enligt g<65533>llande upphovsr<65533>ttslagstiftning.

En "f<65533>r<65533>ndrad version" av dokumentet avser varje verk som inneh<65533>ller dokumentet eller en del av det, antingen ordagranna kopior, eller med <65533>ndringar och/eller <65533>versatt till ett annat spr<65533>k.

Ett "sekund<65533>rt avsnitt" <65533>r en m<65533>rkt bilaga eller f<65533>rord till dokumentet som exklusivt behandlar f<65533>rh<65533>llandet mellan dokumentets f<65533>rl<65533>ggare eller f<65533>rfattare och dokumentets huvudsakliga <65533>mne (eller till relaterade <65533>mnen) och som inte inneh<65533>ller n<65533>got som direkt faller under det huvudsakliga <65533>mnet. (S<65533>ledes, om dokumentet delvis <65533>r en l<65533>robok i matematik s<65533> f<65533>r ett sekund<65533>rt avsnitt inte f<65533>rklara n<65533>gon matematik.) F<65533>rh<65533>llandet kan vara en historisk koppling till <65533>mnet eller n<65533>got relaterat, eller en juridisk, kommersiell, filosofisk, etisk eller politisk st<65533>llning till det.

De "of<65533>r<65533>nderliga avsnitten" <65533>r sekund<65533>ra avsnitt vars titlar <65533>r angivna som of<65533>r<65533>nderliga avsnitt i meddelandet som stadgar att dokumentet <65533>r utgivet under denna licens [det engelska originalet]. Om ett avsnitt inte innefattas av den ovanst<65533>ende definitionen av sekund<65533>rt s<65533> <65533>r det inte till<65533>tet att ange det som of<65533>r<65533>nderligt. Dokumentet beh<65533>ver inte inneh<65533>lla n<65533>gra of<65533>r<65533>nderliga avsnitt. Om dokumentet inte anger n<65533>gra of<65533>r<65533>nderliga avsnitt s<65533> finns det inga.

"Omslagstexterna" <65533>r speciella korta ordf<65533>ljder som <65533>r listade som framsidestexter eller baksidestexter i meddelandet som stadgar att dokumentet <65533>r utgivet under denna licens [det engelska originalet]. En framsidestext kan vara som mest 5 ord och en baksidestext kan vara som mest 25 ord.

En "transparent" kopia av dokumentet <65533>r en maskinl<65533>sbar kopia, representerad i ett format vars specifikation finns tillg<65533>nglig f<65533>r allm<65533>nheten, som l<65533>mpar sig f<65533>r att revidera dokumentet p<65533> ett enkelt s<65533>tt med generella textredigeringsprogram eller (f<65533>r pixelbaserade bilder) generella grafikprogram eller (f<65533>r ritningar) n<65533>got v<65533>l tillg<65533>ngligt ritprogram, och som <65533>r passande som

indata till textfomaterare eller f<65533>r automatisk konvertering till en m<65533>ngd format som passar som indata till textformaterare. En kopia i ett f<65533>r <65533>vrigt transparent filformat vars markeringar, eller avsaknad av markeringar, har ordnats f<65533>r att hindra eller motverka att vidare f<65533>r<65533>ndring vidtas av l<65533>sare <65533>r inte transparent. Ett bildformat <65533>r inte transparent om det anv<65533>nds f<65533>r n<65533>gon betydande del text. En kopia som inte <65533>r "transparent" kallas "opak".

Exempel p<65533> passande format f<65533>r transparenta kopior innefattar ren ASCII utan markeringar, Texinfo indataformat, LaTeX indataformat, SGML eller XML som anv<65533>nder en publikt tillg<65533>nglig DTD, och standardenlig HTML, PostScript eller PDF utformat f<65533>r m<65533>nsklig f<65533>r<65533>ndring. Exempel p<65533> transparenta bildformat innefattar PNG, XCF och JPG. Opaka format innefattar leverant<65533>rsspecifika format som bara kan l<65533>sas och editeras med leverant<65533>rsspecifika ordbehandlare, SGML eller XML f<65533>r vilket DTD och/eller verktyg f<65533>r behandling inte finns allm<65533>nt tillg<65533>ngliga, och den maskingenererade HTML, PostScript eller PDF som produceras av vissa ordbehandlare enbart avsett som utdata.

"Titelsidan" inneb<65533>r, f<65533>r en tryckt bok, titelsidan sj<65533>lv, och s<65533>dana d<65533>rp<65533> f<65533>ljande sidor som kr<65533>vs f<65533>r att g<65533>ra det material som enligt denna licens skall synas p<65533> titelsidan l<65533>sbart. F<65533>r verk i s<65533>dana format som inte har n<65533>gon egentlig titelsida, avses med "titelsida" den text som <65533>r n<65533>rmast den mest framst<65533>ende f<65533>rekomsten av verkets titel, f<65533>reg<65533>ende den huvudsakliga textmassan.

Ett avsnitt "med titeln <65533><65533><65533> (XYZ)" avser en namngiven del av dokumentet vars titel <65533>r exakt XYZ eller inneh<65533>ller XYZ inom parentes efterf<65533>ljande text som <65533>vers<65533>tter XYZ till ett annat spr<65533>k. (H<65533>r st<65533>r XYZ f<65533>r ett speciellt namn p<65533> ett avsnitt nedan, som till exempel "Acknowledgements", "Dedications", "Endorsements" eller "History" [och <65533><65533><65533> f<65533>r l<65533>mplig <65533>vers<65533>ttning, till exempel "tillk<65533>nnagivanden", "dedikationer", "endossering" respektive "historik"].) Att "bevara titeln" p<65533> ett s<65533>dant avsnitt n<65533>r du <65533>ndrar dokumentet inneb<65533>r att det f<65533>rblir ett avsnitt "med titeln <65533><65533><65533> (XYZ)" enligt denna definition.

Dokumentet f<65533>r inneh<65533>lla garantiavs<65533>gelser invid meddelandet om att denna licens [det engelska originalet] g<65533>ller f<65533>r dokumentet. Dessa garantiavs<65533>gelser skall anses vara inkluderade per referens i denna licens, men bara f<65533>r att friskriva fr<65533>n garantier. All annan inneb<65533>rd dessa garantiavs<65533>gelser kan ha <65533>r ogiltiga och p<65533>verkar inte p<65533> n<65533>got s<65533>tt inneb<65533>rden i denna licens.

D.3. 2. ORDAGRANN KOPIERING

Du <65533>ger kopiera och sprida dokumentet p<65533> valfritt medium, antingen kommersiellt eller

ideellt, f<65533>rutsatt att denna licens [det engelska originalet], upphovsr<65533>ttsklausul, och meddelandet som stadgar att GNU Free Documentation License g<65533>ller f<65533>r dokumentet finns med p<65533> alla kopior, och att du inte l<65533>gger till n<65533>gra som helst andra villkor <65533>n de som ing<65533>r i denna licens. Du <65533>ger inte vidta tekniska <65533>tg<65533>rder f<65533>r att begr<65533>nsa eller kontrollera l<65533>sande eller vidare kopiering av de kopior du skapar eller sprider. Dock <65533>ger du ta emot kompensation i utbyte mot kopior. Om du sprider tillr<65533>ckligt m<65533>nga kopior m<65533>ste du ocks<65533> f<65533>lja villkoren i paragraf 3.

Du <65533>ger ocks<65533> l<65533>na ut kopior, under samma villkor som ovan, och du <65533>ger visa kopior offentligt.

D.4. 3. OMFATTANDE KOPIERING

Om du publicerar tryckta kopior (eller kopior i medier som normalt har tryckta omslag) av dokumentet, i en upplaga <65533>verstigande 100 exemplar, och dokumentets licensmeddelande kr<65533>ver omslagstexter, s<65533> m<65533>ste du f<65533>rse kopiorna med omslag som, klart och tydligt, visar alla omslagstexter: framsidestexter p<65533> framsidan och baksidestexter p<65533> baksidan. B<65533>da omslagen m<65533>ste klart och tydligt identifiera dig som utgivare av dessa kopior. Framsidan m<65533>ste presentera dokumentets hela titel, med alla ord i titeln lika framtr<65533>dande och synliga. Du <65533>ger l<65533>gga till ytterligare stoff p<65533> omslagen. Kopiering med f<65533>r<65533>ndringar gjorda bara p<65533> omslaget, s<65533> l<65533>nge som de bevarar dokumentets titel och i <65533>vrigt uppfyller dessa krav kan anses vara ordagrann kopiering i andra avseenden.

Om de obligatoriska texterna f<65533>r n<65533>got omslag <65533>r f<65533>r omfattande f<65533>r att rymmas i l<65533>sbart skick skall du placera de f<65533>rsta (s<65533> m<65533>nga som f<65533>r plats) p<65533> det egentliga omslaget, och forts<65533>tta med resten p<65533> de direkt intilliggande sidorna.

Om du publicerar opaka kopior av dokumentet i upplagor om mer <65533>n 100, m<65533>ste du antingen bifoga en maskinl<65533>sbar transparent kopia med varje opak kopia, eller ange i eller med varje opak kopia en n<65533>tverksadress som <65533>r tillg<65533>nglig f<65533>r den allm<65533>nna n<65533>tverksanv<65533>ndande massan d<65533>r man, med <65533>ppet standardiserade protokoll, kan ladda ner en komplett transparent kopia av dokumentet, utan extra material. Om du v<65533>ljer det senare alternativet, m<65533>ste du vidta sk<65533>liga <65533>tg<65533>rder, n<65533>r du b<65533>rjar sprida opaka kopior i kvantitet, f<65533>r att denna transparenta kopia skall f<65533>rbli tillg<65533>nglig p<65533> angivna platsen till <65533>tminstone ett <65533>r efter den sista g<65533>ngen du spred en opak kopia (direkt eller via ombud eller <65533>terf<65533>rs<65533>ljare) av den utg<65533>van till allm<65533>nheten.

Det <65533>r <65533>nskv<65533>rt, men inte ett krav, att du kontaktar f<65533>rfattarna till dokumentet i god tid innan du sprider n<65533>got st<65533>rre antal kopior, f<65533>r att ge dem en

chans att f<65533>rse dig med en uppdaterad version av dokumentet.

D.5. 4. F<65533>R<65533>NDRINGAR

Du <65533>ger kopiera och sprida en f<65533>r<65533>ndrad version av dokumentet under de villkor som beskrivs i paragraf 2 och 3 av GNU Free Documentation License, f<65533>rutsatt att du sl<65533>pper den f<65533>r<65533>ndrade versionen under exakt denna licens, och att den f<65533>r<65533>ndrade versionen antar dokumentets roll, och s<65533>ledes medger spridning och f<65533>r<65533>ndring av den f<65533>r<65533>ndrade versionen till envar som erh<65533>ller en kopia av den. Ut<65533>ver detta m<65533>ste du g<65533>ra f<65533>ljande med den <65533>ndrade versionen:

A. P<65533> titelsidan (och omslagen om det finns n<65533>gra) anv<65533>nda en titel skild fr<65533>n den som [original]dokumentet har, och skild fr<65533>n tidigare versioners titel (som skall, om det finns n<65533>gra, finnas listade i historikavsnittet i dokumentet). Du <65533>ger anv<65533>nda samma titel som det f<65533>reg<65533>ende dokumentet om den ursprungliga utgivaren ger sitt tillst<65533>nd.

B. Lista, som f<65533>rfattare, en eller flera personer eller juridiska personer som ansvarat f<65533>r f<65533>r<65533>ndringarna i den <65533>ndrade versionen, tillsammans med minst fem av de huvudsakliga f<65533>rfattarna av dokumentet (alla dess huvudsakliga f<65533>rfattare, om det har mindre <65533>n fem), s<65533>vida de inte ger dig tillst<65533>nd att bortse fr<65533>n detta krav.

C. Ange namnet p<65533> utgivaren av den <65533>ndrade versionen, som utgivare, p<65533> titelsidan.

D. Bibeh<65533>lla dokumentets alla upphovsr<65533>ttsklausuler.

E. L<65533>gga till en upphovsr<65533>ttsklausul f<65533>r dina f<65533>r<65533>ndringar angr<65533>nsande till de andra upphovsr<65533>ttsklausulerna.

F. Direkt efter upphovsr<65533>ttsklausulerna innefatta ett meddelande som ger allm<65533>nheten tillst<65533>nd att anv<65533>nda den <65533>ndrade versionen under villkoren i denna licens [det engelska originalet] i den form som visas i Till<65533>gg nedan.

G. I meddelandet om licensen bevara den fullst<65533>ndiga listan <65533>ver of<65533>r<65533>nderliga avsnitt och obligatoriska omslagstexter som finns i dokumentets meddelande om licensen.

H. Inkludera en of<65533>r<65533>ndrad kopia av denna licens [Det <65533>r den engelska originalversionen som avses].

I. Bevara avsnittet med titeln "historik (History)", bevara dess titel och l<65533>gg i avsnittet till en post med <65533>tminstone titeln, <65533>ret, nya f<65533>rfattare och utgivaren av den modifierade versionen s<65533> som angivet p<65533> titelsidan. Om det inte finns n<65533>got avsnitt med titeln "historik (History)" i dokumentet s<65533> skapa en med titeln, <65533>ret, f<65533>rfattare och utgivaren av dokumentet s<65533> som det st<65533>r p<65533> [original]dokumentets titelsida. L<65533>gg sedan till en post som beskriver den f<65533>r<65533>ndrade versionen s<65533> som beskrivits ovan.

J. Bevara den n<65533>tverksadress, om det finns n<65533>gon, angiven i dokumentet till den allm<65533>nt tillg<65533>ngliga transparenta kopian av dokumentet, och likas<65533>n<65533>tverksadresserna till de f<65533>reg<65533>ende versioner som dokumentet baseras p<65533>. Dessa f<65533>r placeras i avsnittet "historik (History)". Du <65533>ger utel<65533>mna en n<65533>tverksadress f<65533>r ett verk som <65533>r publicerat mer <65533>n fyra <65533>r f<65533>re dokumentet sj<65533>lvt, eller om den ursprunglige utgivaren vars verk n<65533>tverksadressen h<65533>nvisar till ger sitt tillst<65533>nd.

K. F<65533>r alla avsnitt med titlarna "tillk<65533>nnagivanden (Acknowledgements)" eller "dedikationer (Dedications)", bevara titeln p<65533> avsnittet, och bevara allt inneh<65533>ll och pr<65533>gel p<65533> alla tillk<65533>nnagivanden och/eller dedikationer gjorda av varje bidragare.

L. Bevara alla of<65533>r<65533>nderliga avsnitt i dokumentet of<65533>r<65533>ndrade till text och titel. Avsnittsnummer eller motsvarande anses inte tillh<65533>ra avsnittets titel.

M. Radera varje avsnitt med titeln "endossering (Endorsements)". Ett s<65533>dant avsnitt f<65533>r inte inkluderas i en modifierad version.

N. Inte byta titel p<65533> n<65533>got existerande avsnitt s<65533> att det blir "endossering (Endorsements)" eller s<65533> att titeln kan f<65533>rv<65533>xlas med n<65533>got of<65533>r<65533>nderligt avsnitt.

O. Bevara varje garantiavs<65533>gelseklausul.

Om den f<65533>r<65533>ndrade versionen inneh<65533>ller nya framsidestexter eller bilagor som <65533>r att anses som sekund<65533>ra avsnitt och inte inneh<65533>ller n<65533>got material kopierat fr<65533>n dokumentet, s<65533> <65533>ger du, om du vill, ben<65533>mna n<65533>gra eller samtliga av dessa som of<65533>r<65533>nderliga. F<65533>r att g<65533>ra detta, l<65533>gg deras titlar till listan <65533>ver of<65533>r<65533>nderliga avsnitt i den f<65533>r<65533>ndrade versionens licensmeddelande. Dessa titlar m<65533>ste vara skilda fr<65533>n alla andra avsnitts titlar.

Du <65533>ger l<65533>gga till ett avsnitt med titeln "endossering (Endorsements)", f<65533>rutsatt att det inte inneh<65533>ller n<65533>got annat <65533>n endosseringar f<65533>r din modifierade version fr<65533>n olika akt<65533>rer -- till exempel, meddelanden om utf<65533>rd korrekturl<65533>sning eller att texten har godk<65533>nts av en organisation som en officiell definition av en standard.

Du <65533>ger l<65533>gga till ett textavsnitt p<65533> upp till fem ord som framsidestext, och ett textavsnitt p<65533> upp till 25 ord som baksidestext i listan <65533>ver omslagstexter i den modifierade versionen. Bara ett textavsnitt med framsidestexter och ett med baksidestexter f<65533>r l<65533>ggas till av (eller genom f<65533>rsorg av) en enda juridisk person. Om dokumentet redan inneh<65533>ller en omlagstext f<65533>r n<65533>got av omslagen, tidigare tillagd av dig eller genom f<65533>rsorg av samma juridiska person som du f<65533>retr<65533>der, <65533>ger du inte l<65533>gga till en till, men du <65533>ger <65533>ndra den gamla med tillst<65533>nd fr<65533>n den tidigare utgivaren som lade till den f<65533>rra.

F<65533>rfattaren (f<65533>rfattarna) och utgivaren (utgivarna) av dokumentet ger inte via denna licens sitt tillst<65533>nd att anv<65533>nda sina namn f<65533>r publicitet eller f<65533>r att

l<65533>gga till eller antyda endossering av n<65533>gon modifierad version.

D.6. 5. KOMBINERA DOKUMENT

Du <65533>ger kombinera dokumentet med andra dokument som <65533>r utgivna under denna licens, under de villkor som definieras i paragraf 4 av GNU Free Documentation License f<65533>r modifierade versioner, f<65533>rutsatt att du, i det kombinerade dokumentet, innefattar alla of<65533>r<65533>nderliga avsnitt fr<65533>n originaldokumenten, omodifierade, och listar dem som of<65533>r<65533>nderliga avsnitt i ditt kombinerade verk i dess licensklausul, och att du bevarar alla deras garantiavs<65533>gelseklausuler.

Det kombinerade verket beh<65533>ver bara inneh<65533>lla en enstaka kopia av denna licens [engelska originalversionen], och flera identiska of<65533>r<65533>nderliga stycken kan ers<65533>ttas med en kopia. Om det finns flera of<65533>r<65533>nderliga stycken med samma namn men olika inneh<65533>ll, se till att titeln p<65533> varje s<65533>dant avsnitt <65533>r unik genom att i slutet p<65533> den, inom parentes, l<65533>gga till namnet p<65533> den ursprunglige f<65533>rfattaren eller utgivaren av det avsnittet om dessa <65533>r k<65533>nda, annars ett unikt nummer. G<65533>r samma justeringar av titlarna i listan <65533>ver of<65533>r<65533>nderliga avsnitt i licensklausulen i det kombinerade verket.

I det kombinerade verket m<65533>ste du kombinera alla avsnitt med titlarna "historik (History)" i de ursprungliga dokumenten, till ett avsnitt med titeln "historik (History)"; p<65533> samma s<65533>tt skall alla avsnitt med titlarna "tillk<65533>nnagivanden (Acknowledgements)", alla avsnitt med titlarna "dedikationer (Dedications)" kombineras. Du m<65533>ste ta bort alla avsnitt med titlarna "endossering (Endorsements)".

D.7. 6. SAMLINGAR AV DOKUMENT

Du <65533>ger skapa en samling best<65533>ende av dokumentet och andra dokument som <65533>r sl<65533>ppta under GNU Free Documentation License, och ers<65533>tta individuella kopior i dokumenten av denna licens med en enda kopia [av den engelska originalversionen] som inkluderas i samlingen, f<65533>rutsatt att du f<65533>ljer villkoren f<65533>r ordagrann kopiering i denna licens f<65533>r varje inkluderat dokument i alla andra avseenden.

Du <65533>ger lyfta ut ett dokument fr<65533>n en s<65533>dan samling, och sprida det enskilt under GNU Free Documentation License, f<65533>rutsatt att du l<65533>gger till en kopia av denna licens [den engelska originalversionen] i det utlyfta dokumentet, och f<65533>ljer villkoren f<65533>r ordagrann kopiering i denna licens f<65533>r det utlyfta dokumentet i alla andra avseenden.

D.8. 7. SAMMANSLAGNING MED OBEROENDE VERK

En samling av dokumentet eller av dess derivat med andra separata och oberoende dokument eller verk, p<65533> eller i en lagringsvolym eller ett spridningsmedium, kallas f<65533>r en "sammanslagning" om den sammanslagna upphovsr<65533>tten inte anv<65533>nds f<65533>r att begr<65533>nsa samlingens anv<65533>ndares r<65533>ttigheter som de enskilda dokumenten medger. N<65533>r dokumentet ing<65533>r i en s<65533>dan sammanslagning, g<65533>ller inte denna licens de andra verken i samlingen som inte sj<65533>lva <65533>r deriverat av dokumentet.

Om kravet p<65533> omslagstexter enligt paragraf 3 <65533>r till<65533>mpligt p<65533> dessa kopior av dokumentet, s<65533> kan dokumentets omslagstexter, om dokumentet utg<65533>r mindre <65533>n h<65533>lften av hela samlingen, placeras p<65533> det omslag som omger dokumentet inuti samlingen, eller den elektroniska motsvarigheten till omslag om dokumentet <65533>r i elektronisk form. Annars m<65533>ste de synas p<65533> det omslag som omger hela samlingen.

D.9. 8. <65533>VERS<65533>TTNING

<65533>vers<65533>ttning anses vara en sorts f<65533>r<65533>ndring, s<65533> du <65533>ger sprida <65533>vers<65533>ttningar av dokumentet enligt de villkor som s<65533>tts i paragraf 4. Of<65533>r<65533>nderliga avsnitt som ers<65533>tts med <65533>vers<65533>ttningar kr<65533>ver tillst<65533>nd fr<65533>n deras upphovsr<65533>ttsinnehavare, men du <65533>ger inkludera <65533>vers<65533>ttningar av alla eller vissa av dessa of<65533>r<65533>nderliga avsnitt tillsammans med originalversionerna av dessa of<65533>r<65533>nderliga avsnitt. Du <65533>ger inkludera en <65533>vers<65533>ttning av denna licens, och alla licensklausuler i dokumentet, och alla garantiavs<65533>gelser, f<65533>rutsatt att du ocks<65533> innefattar den engelska originalversionen av denna licens och originalversionerna av dessa klausuler. Skulle det finnas skillnader mellan <65533>vers<65533>ttningen och originalversionen av denna licens eller n<65533>gon klausul s<65533> g<65533>ller originalversionen.

Om ett avsnitt i dokumentet har titeln "tillk<65533>nnagivanden (Acknowledgements)", "dedikationer (Dedications)", eller "historik (History)", kommer kravet (paragraf 4) att bevara dess titel (paragraf 1) vanligtvis att kr<65533>va att sj<65533>lva titeln <65533>ndras.

D.10. 9. UPPH<65533>RANDE

Du <65533>ger inte kopiera, f<65533>r<65533>ndra, omlicensiera eller sprida dokumentet annat <65533>n enligt villkoren i GNU Free Documentation License. Alla <65533>vriga f<65533>rs<65533>k att kopiera, modifiera, omlicensiera, eller sprida dokumentet <65533>r ogiltiga och kommer automatiskt medf<65533>ra att du f<65533>rlorar dina r<65533>ttigheter enligt denna licens. Tredje man som har mottagit kopior eller r<65533>ttigheter fr<65533>n dig enligt dessa licensvillkor kommer dock inte att f<65533>rlora sina r<65533>ttigheter s<65533> l<65533>nge de f<65533>ljer licensvillkoren.

D.11. 10. FRAMTIDA VERSIONER AV DENNA LICENS

Free Software Foundation kan publicera nya, reviderade versioner av GNU Free Documentation License d<65533> och d<65533>. S<65533>dana nya versioner kommer att vara likadana i andemening som den nuvarande versionen, men kan skilja i detalj f<65533>r att behandla nya problem eller angel<65533>genheter. Se http://www.gnu.org/copyleft/.

Varje version av licensen ges ett unikt versionsnummer. Om dokumentet stadgar att en specifik numrerad version av denna licens "eller valfri senare version" g<65533>ller f<65533>r det, s<65533> <65533>ger du r<65533>tten att f<65533>lja villkoren enligt antingen den angivna versionen eller vilken senare version som helst som publicerats (inte som utkast) av Free Software Foundation. Om dokumentet inte anger en version av denna licens, <65533>ger du v<65533>lja vilken version som helst som publicerats (inte som utkast) av Free Software Foundation.

D.12. TILL<65533>GG: Hur du anv<65533>nder denna licens f<65533>r dina dokument

F<65533>r att anv<65533>nda GNU Free Documentation License f<65533>r ett dokument du har skrivit, inkludera en kopia av licensen [det engelska originalet] i dokumentet och placera f<65533>ljande copyrightklausul omedelbart efter titelsidan:

> Copyright (c) <65533>RTAL DITT NAMN. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".
>
> [<65533>vers<65533>ttning:] Copyright (c) <65533>RTAL DITT NAMN. Var och en <65533>ger r<65533>tt att kopiera, sprida och/eller f<65533>r<65533>ndra detta dokument under villkoren i licensen "GNU Free Documentation License", version 1.2 eller senare publicerad av Free Software Foundation, utan of<65533>r<65533>nderliga avsnitt, utan framsidestexter och utan baksidestexter. En kopia av denna licens finns med i avsnittet med titeln "GNU Free Documentation License".

Om du har of<65533>r<65533>nderliga avsnitt, framsidestexter och baksidestexter, ers<65533>tt "with...Texts." ("utan...texter.") med f<65533>ljande:

> with the Invariant Sections being LISTA DERAS TITLAR, with the Front-Cover Texts being LISTA, and with the Back-Cover Texts being LISTA.
>
> [<65533>vers<65533>ttning:] med de of<65533>r<65533>nderliga avsnitten LISTA DERAS TITLAR, med framsidestexterna LISTA, och med baksidestexterna LISTA.

Om du har of<65533>r<65533>nderliga avsnitt utan omslagstexter, eller n<65533>gon annan kombination av de tre, sl<65533> samman dessa tv<65533> alternativ s<65533> att det passar den

uppkomna situationen.

Om ditt dokument inneh<65533>ller icke-triviala exempel med programkod, s<65533> rekommenderar vi att du sl<65533>pper dessa exempel parallellt under en, av dig vald, fri programvarulicens, som till exempel GNU General Public License, f<65533>r att m<65533>jligg<65533>ra deras anv<65533>ndning i fri programvara.