

Systemutveckling PHP

Föreläsning 01 - Grunderna i PHP

Dagens ämnen

- Formalia
- Variabler
- Utskrift
- Kontrollstrukturer
- Utvecklingsmiljö

Kursplan



11

Systemutveckling PHP

Utbildning som kursen ingår i:
Webbutvecklare inom e-handel

Kursens omfattning:
50 yrkeshögskolepoäng

Undervisningsspråk:
Svenska

Förkunskapskrav: inga

Valbar kurs:
Nej

Syfte och mål:

Kursens syfte är att behandla grundläggande tekniker, arkitektur och systemdesign inom backend-webbutveckling för att skapa e-handelsdrivna webbplatser där information kan lagras, hanteras, analyseras och presenteras.

Målet med kursen är att den studerande ska utveckla en helhetsförståelse för information- och användarflöden i e-handelsapplikationer och förmåga att i yrkesrollen som e-handelsutvecklare ta fram fungerande, databasdrivna e-handelsplattformar med relevanta tekniker och funktioner.

Kursplan

Kunskaper:

- Om grundläggande programmering i ett backend-webbutvecklingsspråk.
- Om databaser.
- Om arkitektur och systemdesign för e-handelsapplikationer.
- Om programstruktur för hantering av användare och användardata.
- Om programstruktur för hantering av information och informationsflöde.
- Om säkerhet, integritet och lagstiftning.
- I ett backend-programmeringsspråk för webben.
- I planering av att skapa dynamiska e-handelsapplikationer.
- I hantering av personuppgifter och säkerhet.

Färdigheter:

- I att ta fram en e-handelsplattform.
- I att planera och genomföra större databasdrivna programmeringsprojekt.
- I att välja ut, designa och anpassa databaser utifrån den givna uppgiften.
- I att designa system och kodbaser utifrån arkitektoniska principer.

Kompetenser:

- Att planera och genomföra framtagandet av användar- och databasdrivna e-handelsapplikationer.
- Att optimera och anpassa databaser utifrån e-handelsplattformar.

Kursplan

- Att använda agila arbetsmetoder för att genomföra programmeringsprojekt inom utsatta tidsramar.

Kursens innehåll och delmoment

Kursen syftar till att ge den studerande färdigheten att arbeta som webbutvecklare på serversidan. För att möjliggöra detta måste studenten få kunskap om server-mjukvara, säkerhet, sessions/state, olika ramverk samt SQL databaser.

Efter att den ovanstående grundläggande kunskapen har tillgodogjorts, kommer studenten att introduceras till OOP (Objekt-orienterad programmering).

Den studerande skall vid avslutad kurs kunna bygga en objekt-orienterad back-end applikation i PHP som är kopplad till SQL databas.

Delmoment

- **PHP**
Studenten ska få djup kunskap och färdigheter i programmering och med särskilt fokus på objekt-orienterad programmering i PHP för webben.
- **Säkerhet**
Studenterna ska få god kunskap och vikten av att filtrera och validera data för att undvika intrång och skadlig kod.
- **Databashantering**
Studenten ska efter avslutad kurs ha kunskaper och kompetens att använda inbyggda funktioner i programmeringsspråk för att göra säkra anrop till en databas. Exempelvis PDO i PHP med prepared statements.

Kursplan

Former för kunskapskontroller

- 2 projektarbeten ett individuellt och 1 i grupp (innefattande skriftligt och muntlig rapport/presentation)
- 1 skriftligt prov/tentamen

Kursen kommer att betygsättas med hjälp av inlämningsuppgifter, projektarbeten, muntlig presentation och skriftligt prov/tentamen.

De kunskapskontroller (inlämningsuppgifter, projektarbeten och tentamen) som görs i utsatt omfång och i enlighet med kursledningens instruktioner har möjlighet att få bedömningen IG, G och i vissa fall VG. Dessa viktas och ligger sedan sammantaget till grund för hela kursens betygsättning kombinerat med lärarens/utbildningsledarens övergripande erfarenheter av studentens insatser på kursen.

Betygskriterier

IG – G – VG

För att erhålla följande betyg krävs att den studerande;

Kursplan

Godkänt

- Den studerande har nått samtliga mål för kursen

Väl godkänt

- Den studerande har nått samtliga mål för kursen
- Den studerande kan dessutom:
 - Skapa en mind map för arbete med objektorienterad programmering
 - Diskutera och implementera Objektorienterad programmering
 - Diskutera och implementera MVC- ramverk och CMS
 - Diskutera och analysera olika nya relevanta tekniker

Schema

Mån		Tis	Ons	Tors	Fre
8:30 - 12:00	Föreläsning Övningar	Egna studier	Föreläsning Övningar	Föreläsning Övningar	Egna studier
12:00 - 13:00	Lunch		Lunch	Lunch	
13:00 - 15:30	Projekt Genomgång Eget arbete		Öppet forum Hjälp Repetition	Projekt Genomgång Eget arbete	

Kommunikation

- Zenit?
- Slack?

PHP

- "PHP [...] är ett populärt skriptspråk som främst körs på webbservrar för att driva internetsajter med dynamiskt innehåll (det vill säga innehåll som genereras från till exempel databas eller besökarens formulärdata). Det finns dock också en CLI-applikation för PHP som gör att skript kan köras direkt, utan webbrowser." - <https://sv.wikipedia.org/wiki/PHP>

Skriptspråk

- "Skriptspråk är inom datavetenskap en benämning på "små" högnivåspråk inriktade på specialiserade uppgifter inom redan befintliga miljöer, i motsats till systemspråk, som används för programmering av tillämpningsprogram. Det finns ingen exakt avgränsning mellan systemspråk och skriptspråk, men skriptspråk är ofta interpreterande programspråk som använder dynamisk typning."
 - <https://sv.wikipedia.org/wiki/Skriptspr%C3%A5k>

Skriptspråk

- "I en interpreterande miljö tolkas programkoden samtidigt som programmet körs, till skillnad mot en kompilerande miljö, där koden översätts i en separat process till maskinkod."
 - https://sv.wikipedia.org/wiki/Interpreterande_programspr%C3%A5k
- "Dynamiskt typade variabler kan referera till vilken datatyp som helst."
 - https://sv.wikipedia.org/wiki/Typsystem#Dynamisk_typning

Övningsplats

- <http://sandbox.onlinephpfunctions.com/>
- ✓ Kräver inte någon installation
- Ger oss inte kontroll över inställningar
- Låter oss inte interagera med html
- Vi kommer även att använda andra siter.
- Vi kommer att fixa lokala utvecklingsmiljöer senare.

Starta med PHP

- PHP omges av den speciella PHP-taggen:
HTML-fil:

```
<?php // PHP code goes here ?>
```

```
<body>  
<h1>My first PHP page</h1>  
<?php // Do something ?>  
</body>
```

Variabler

- `$name = "Micke";`
- Behöver inte deklarerats innan den används.
- Svagt typat språk vilket betyder att variabler kan byta datatyp under körning.
- Startar med ett dollartecken följt av namnet.
- Namnet måste starta med en bokstav eller ett underscore: `_`
- Kan enbart innehålla stora eller små bokstäver (a-z), siffror och underscore.
- Kan inte innehålla mellanslag.
- Case sensitive, `$x` och `$X` är olika variabler.

Konstanter

- En konstant är som en variabel, men man kan inte byta värde på den efter att det är satt.

```
// Valid constant names
define( "FOO",      "something" );
define( "FOO2",     "something else" );
define( "FOO_BAR",  "something more" );
```

Skriva ut

Kommentar. Vad innebär det?

- `echo "Micke";` `// Vanligast`
- <http://php.net/manual/en/function.echo.php>
- `print "Micke";`
- <http://php.net/manual/en/function.print.php>

Datatyper

- PHP stöder följande datatyper:
 - String
 - Integer
 - Float (decimaltal, även kallat double)
 - Boolean
 - Array
 - Object
 - NULL
 - Resource

Aritmetiska operatörer

Exempel	Namn	Resultat
$+$a$	Identitet	Konvertering av a till int eller float.
$-$a$	Negation	Motsatsen till a .
$a + b$	Addition	Summan av a och b .
$a - b$	Subtraktion	Skillnaden mellan a och b .
$a * b$	Multiplikation	Produkten av a och b .
a / b	Division	Kvoten av a och b .
$a \% b$	Modulus	Resten av a delat med b .
$a ** b$	Exponent	a upphöjt i b .

Ökande/minskande operatorer

Exempel	Namn	Effekt
<code>++\$a</code>	Pre-increment	Ökar \$a med ett, returnerar sedan \$a.
<code>\$a++</code>	Post-increment	Returnerar \$a, ökar sedan \$a med ett.
<code>--\$a</code>	Pre-decrement	Minskar \$a med ett, returnerar sedan \$a.
<code>\$a--</code>	Post-decrement	Returnerar \$a, minskar sedan \$a med ett.

Jämförelseoperatorer

Exempel	Namn	Resultat
<code>\$a == \$b</code>	Equal	TRUE if \$a is equal to \$b after type juggling.
<code>\$a === \$b</code>	Identical	TRUE if \$a is equal to \$b, and they are of the same type.
<code>\$a != \$b</code>	Not equal	TRUE if \$a is not equal to \$b after type juggling.
<code>\$a <> \$b</code>	Not equal	TRUE if \$a is not equal to \$b after type juggling.
<code>\$a !== \$b</code>	Not identical	TRUE if \$a is not equal to \$b, or they are not of the same type.
<code>\$a < \$b</code>	Less than	TRUE if \$a is strictly less than \$b.
<code>\$a > \$b</code>	Greater than	TRUE if \$a is strictly greater than \$b.
<code>\$a <= \$b</code>	Less than or equal to	TRUE if \$a is less than or equal to \$b.
<code>\$a >= \$b</code>	Greater than or equal to	TRUE if \$a is greater than or equal to \$b.
<code>\$a <=> \$b</code>	Spaceship	An integer less than, equal to, or greater than zero when \$a is respectively less than, equal to, or greater than \$b. Available as of PHP 7.

Övningar

- <https://www.w3schools.com/php/>
 - Variables
 - Echo / Print
 - Data Types
 - Constants
- Extra övningar för er som hinner:
<https://www.learn-php.org/>
 - Hello, World!
 - Variables and types

If-satser

- Låter oss styra programmet beroende på olika villkor.

```
$x=1;
```

```
if( $x==1 )  
    echo "x är lika med ett.";
```

- Vad händer om vi vill skriva ut två gånger?

```
if( $x==1 )  
    echo "x är lika med ett.";  
    echo "x är inte lika med två.";
```

Block

- Ett block utgörs av kod som står mellan måsvingar. All kod inom ett block hör ihop som en instruktion.

```
if($x==1) {  
    echo "x är lika med ett.";   
    echo "x är inte lika med två.";   
}
```

- Det är best practise att alltid använda sig av block, oavsett om man vill utföra en eller flera instruktioner.

If-else-satser

```
$x=1;
```

```
if($x==1) {  
    echo "x är lika med ett."  
}
```

Else är inte obligatorisk

```
else if($x==2) {  
    echo "x är lika med två."  
}
```

```
else {  
    echo "x är varken lika med ett eller två."  
}
```

Loopar

- ”en konstruktion inom imperativa programmeringsspråk för att åstadkomma en iteration, så att en serie satser upprepas flera gånger”
 - for
 - while
 - switch

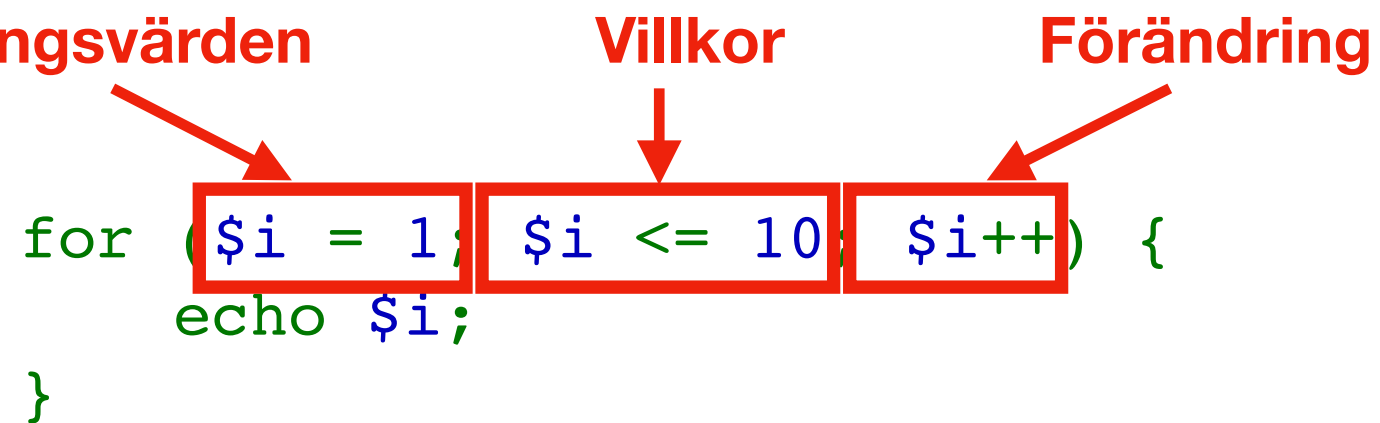
for

Initieringsvärden

Villkor

Förändring

```
for ($i = 1; $i <= 10; $i++) {  
    echo $i;  
}
```



for

```
for ($i = 1; $i <= 10; $i++) {  
    echo $i;  
}
```

Loop	\$i	Fortsätta?
1	1	true
2	2	true
3	3	true
4	4	true
5	5	true
6	6	true
7	7	true
8	8	true
9	9	true
10	10	true
11	11	false

while

Villkor
↓

```
$i = 1;  
while ($i <= 10) {  
    echo $i++; /* the printed value would be  
               $i before the increment  
               (post-increment) */  
}
```


switch

- Det finns ett annat sätt att skriva if-satser som ser ut såhär:

```
if ($i == 0) {  
    echo "i equals 0";  
} elseif ($i == 1) {  
    echo "i equals 1";  
} elseif ($i == 2) {  
    echo "i equals 2";  
}
```

switch

Värde att kolla

↓

```
switch ($i) {  
  case 0:  
    echo "i equals 0";  
    break;  
  case 1:  
    echo "i equals 1";  
    break;  
  case 2:  
    echo "i equals 2";  
    break;  
}
```

```
if ($i == 0) {  
  echo "i equals 0";  
} elseif ($i == 1) {  
  echo "i equals 1";  
} elseif ($i == 2) {  
  echo "i equals 2";  
}
```

switch

Gäller för alla tre fallen.



```
switch ($i) {  
    case 0:  
    case 1:  
    case 2:  
        echo "i is less than 3 but not negative";  
        break;  
    case 3:  
        echo "i is 3";  
}
```

switch

```
switch ($i) {  
    case 0:  
        echo "i equals 0";  
        break;  
    case 1:  
        echo "i equals 1";  
        break;  
    case 2:  
        echo "i equals 2";  
        break;  
    default:  
        echo "i is not equal to 0, 1 or 2";  
}
```

Default gäller om inget annat matchar.



Övningar

- <https://www.w3schools.com/php/>
 - If ... Else ... Elseif
 - Switch
 - While Loops
 - For Loops
- Extra övningar för er som hinner:
<https://www.learn-php.org/>
 - For Loops
 - While Loops

Funktioner

- Funktioner är kod vi kan anropa flera gånger.
 - Systemfunktioner
 - Användarfunktioner

Systemfunktioner

- PHP har en massa inbyggda funktioner för att koppla upp databaser, hantera strängar osv.
- <http://php.net/manual/en/funcref.php>

```
$mystring = 'abc';  
$findme   = 'a';  
$pos = strpos($mystring, $findme);  
  
// Note our use of ===. Simply == would not work as expected  
// because the position of 'a' was the 0th (first) character.  
if ($pos === false) {  
    echo "The string '$findme' was not found in the string '$mystring';"  
} else {  
    echo "The string '$findme' was found in the string '$mystring';"  
    echo " and exists at position $pos";  
}
```

- <http://php.net/manual/en/function.strpos.php>

Användarfunktioner

- Vi kan skriva egna funktioner

```
function writeMsg() {  
    echo "Hello world!";  
}
```

```
writeMsg(); // call the function
```

Argument

- Vi kan skicka data, parametrar, argument till en funktion.

```
function familyName($fname, $year) {  
    echo "$fname Refsnes. Born in $year <br>";  
}
```

```
familyName("Hege", "1975");  
familyName("Stale", "1978");  
familyName("Kai Jim", "1983");
```

Argument

- Argument kan ha default-värden.

```
function setHeight($minheight = 50) {  
    echo "The height is : $minheight <br>";  
}
```

```
setHeight(350);  
setHeight(); // will use the default value of 50
```

Returnera värden

- Funktioner kan även returnera värden.

```
function sum($x, $y) {  
    $z = $x + $y;  
    return $z;  
}
```

```
echo "5 + 10 = " . sum(5, 10) . "<br>";  
echo "7 + 13 = " . sum(7, 13) . "<br>";  
echo "2 + 4 = " . sum(2, 4);
```

Övningar

1. Skriv en funktion som skriver ut en rektangels area. Låt den ta två sidor som argument.
2. Skriv en funktion som returnerar dubbla argumentets värde, dvs om du skickar in 4 till funktionen ska den returnera 8.
3. Skriv en funktion som tar ett startvärde och ett slutvärde som argument. Slutvärdet ska ha ett default-värde. Funktionen ska med hjälp av en loop anropa din andra funktion en gång för varje värde.
4. Skriv en funktion som konverterar grader i F till C.
5. Skriv en funktion som tar två värden och skriver ut det största värdet.
6. Research-uppgift: Skriv en rekursiv funktion som räknar ut fakulteten för ett heltal.

Internets uppbyggnad

- Decentraliserat
- Kunna koppla upp via olika vägar
- IP-adress
- DNS
- Ping/Traceroute
- http / https
- ftp / ftps
- ssh

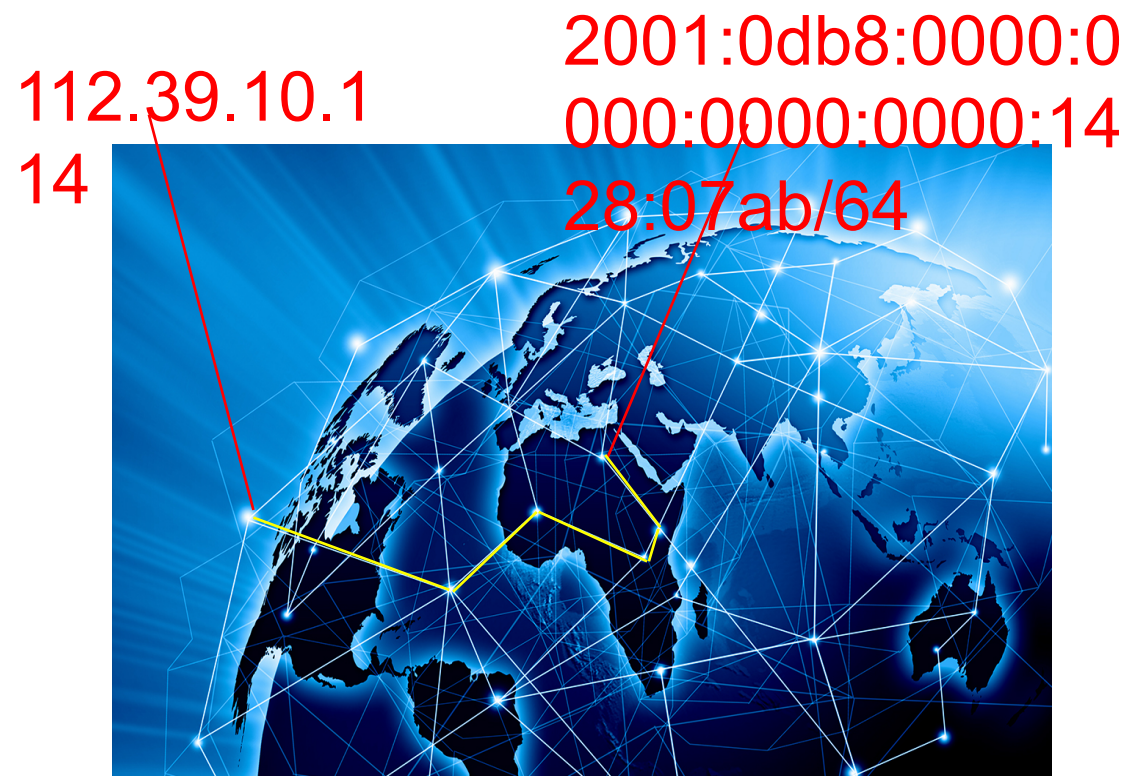


Internet - decentraliserat

- Utvecklat av militären
- Om en stad wipas ut måste informationen fortfarande kunna komma fram



Internet - IP-adress



- Alla maskiner som vill kunna kommunicera över internet behöver en adress.
- v4: nnn.nnn.nnn.nnn
- v6:
xxxx:xxxx:xxxx:xxxx:xxxx:xxx
x:xxxx:xxxx (/nn)

Internet - DNS

- Domain Name System
- Adressbok för IP-adresser

Sweet! Nu vet jag vilken IP-adress jag ska till.

Mickes dator

www.aftonbladet.se?

Yes, den har 104.20.55.70

www.aftonbladet.se
104.20.55.70

Jag har din sida, kommer här!

Nice! Sparar det ett tag.

Ingen aning, frågar vidare.



Internet - Ping / Traceroute

- Ping används för att se om en maskin svarar.
- Vad innebär det om IP-adressen svarar men inte
- Traceroute används för att se vilka maskiner man passerar på vägen.
- Bra för att ta reda på var på

```
2. micke@Mikaels-MacBook-Pro: ~ (zsh)
~ ping www.aftonbladet.se -c 1
PING www.aftonbladet.se.cdn.cloudflare.net (104.20.55.70): 56 data bytes
64 bytes from 104.20.55.70: icmp_seq=0 ttl=59 time=5.191 ms

--- www.aftonbladet.se.cdn.cloudflare.net ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 5.191/5.191/5.191/0.000 ms
~ ping 104.20.55.70 -c 1
PING 104.20.55.70 (104.20.55.70): 56 data bytes
64 bytes from 104.20.55.70: icmp_seq=0 ttl=59 time=11.252 ms

--- 104.20.55.70 ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 11.252/11.252/11.252/0.000 ms
~
```

```
2. micke@Mikaels-MacBook-Pro: ~ (zsh)
--- 104.20.55.70 ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 11.252/11.252/11.252/0.000 ms
~ clear
~ traceroute www.aftonbladet.se
traceroute: Warning: www.aftonbladet.se has multiple addresses; using 104.20.55.70
traceroute to www.aftonbladet.se.cdn.cloudflare.net (104.20.55.70), 64 hops max, 52 byte packets
 1 192.168.1.1 (192.168.1.1)  1.316 ms  0.936 ms  0.850 ms
 2 ti3068a430.ti.telenor.net (146.172.71.23)  1.696 ms  1.599 ms  1.378 ms
 3 ti3163c360-ae68-0.ti.telenor.net (146.172.21.2)  9.867 ms  4.530 ms  4.433 ms
 4 ti3001b400-ae3-0.ti.telenor.net (146.172.105.62)  5.593 ms  4.369 ms  4.472 ms
 5 netnod-ix-ge-b-sth-1500.cloudflare.com (194.68.128.246)  4.140 ms  4.111 ms  4.042 ms
 6 104.20.55.70 (104.20.55.70)  4.042 ms  4.037 ms  4.007 ms
~
```

Internet - http

- HyperText Transfer Protocol
 - används för att överföra webbsidor
 - definierar åtta kommandon
 - GET
 - HEAD
 - POST
 - PUT
 - DELETE
 - TRACE
 - OPTIONS
 - CONNECT
 - Svaret från webbservern innehåller en HTTP-statuskod
- 200: OK
- 301: Moved Permanently
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 500: Internal Server Error

Internet - http

- Exempel klientförfrågan

```
GET /index.html HTTP/1.1  
Host: www.example.com
```

- Exempel serversvar
- (följs av en blankrad och texten i det efterfrågade dokumentet)

```
HTTP/1.1 200 OK  
Date: Mon, 23 May 2005 22:38:34 GMT  
Server: Apache/1.3.27 (Unix) (Red-Hat/Linux)  
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT  
Etag: "3f80f-1b6-3e1cb03b"  
Accept-Ranges: bytes  
Content-Length: 438  
Connection: close  
Content-Type: text/html; charset=UTF-8
```

Internet - https

- HyperText Transfer Protocol Secure
- Krypterad http
- Använder TLS, tidigare SSL
- Betrodd tredje part tillhandahåller ett undertecknat digitalt certifikat. Certifikatet installeras på webbservern och kontrolleras av webbläsaren med hjälp av den tredje partens lokalt installerade certifikat.
- Har varit dyrt och prestandaförsämrade, inte längre.

Internet - http/2

- Nyaste versionen av http
- Utvecklades från Googles SPDY för att vara snabbare och säkrare
- https är en förutsättning

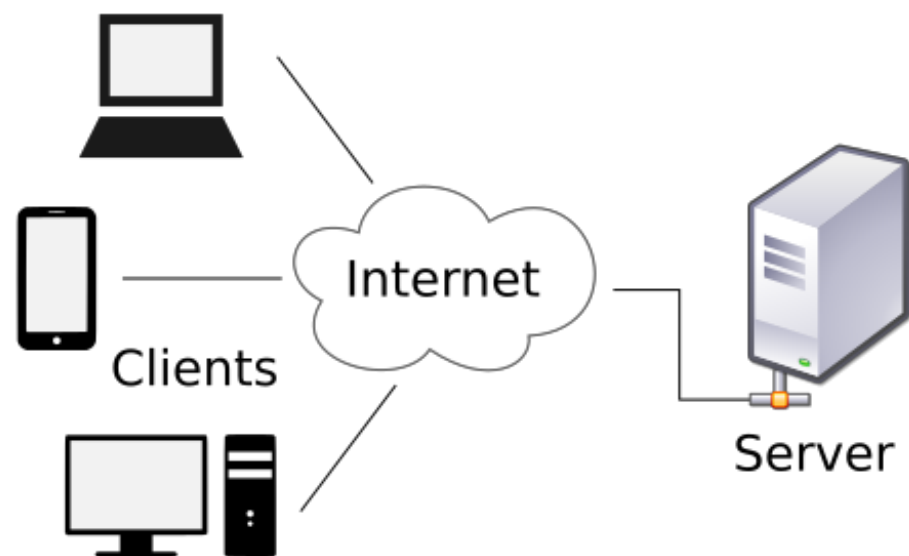
Internet - SSH

- Secure SHell
- Används för att logga in på säkert sätt mot andra maskiner
- Terminalverktyg

Internet - ftp / ftps / sftp

- File Transfer Protocol
- Protokoll för filöverföring
- FTPS = FTP over SSL
- SFTP = SSH FTP

Klient/server-modellen



Server utför något slags arbete, delar resurser

Klient är en beställare.

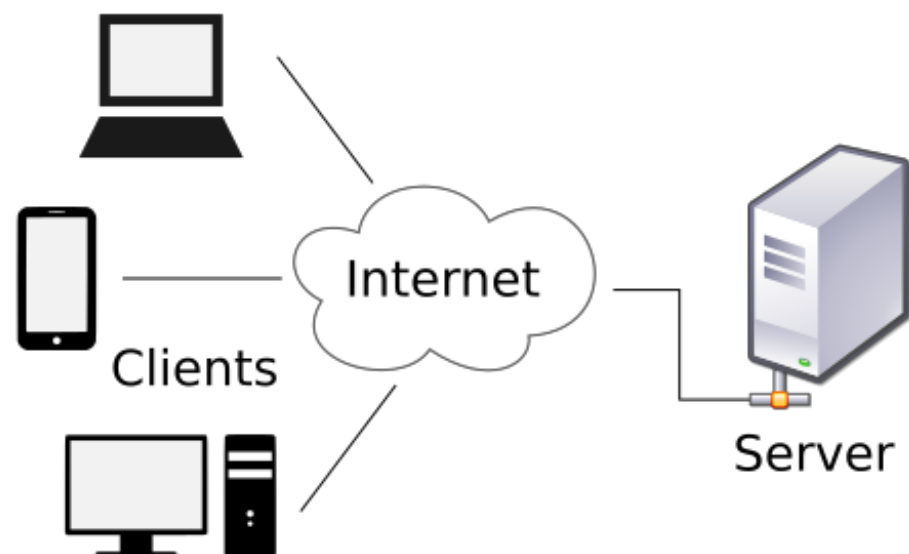
Ofta över ett nät, men inte alltid.

Klienten delar inte med sig av sina resurser, den begär en tjänst.

Klienten initierar kommunikation, servern inväntar den.

Ex: mail, nätverksskrivare, www.

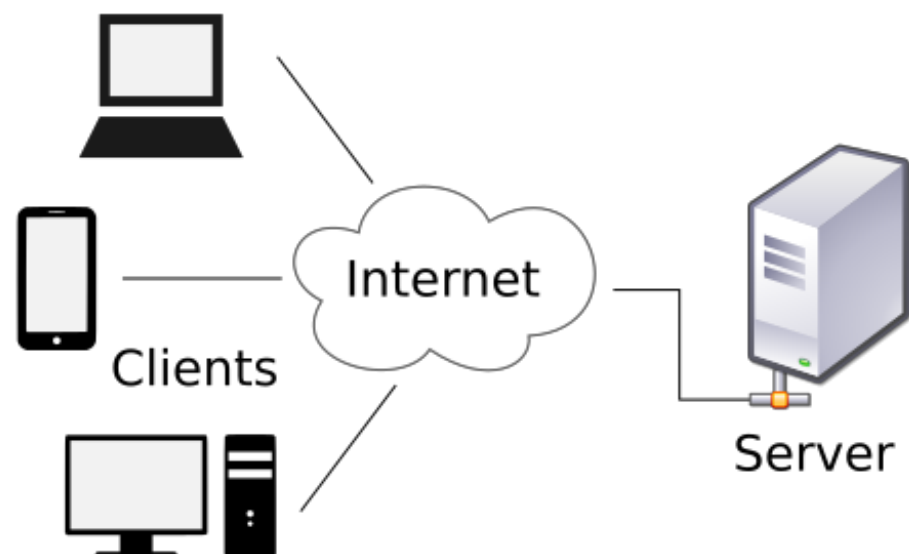
Klient/server-modellen



- Server är inte en slags dator.
- Server är ett program.
 - Lyssnar på requests
 - Kan i princip köras på vilken dator som helst
 - Datorn som kör server-programmet kallas ofta för server.
 - Man kan köra en server på sin egen dator. Då agerar samma dator både server och klient.

Förvirrande!

Klient/server-modellen



- Vad finns det för klienter?
 - Browsers
 - Mailklienter
 - Ftp-klient
 - SSH-klient

Internet - webbrowser

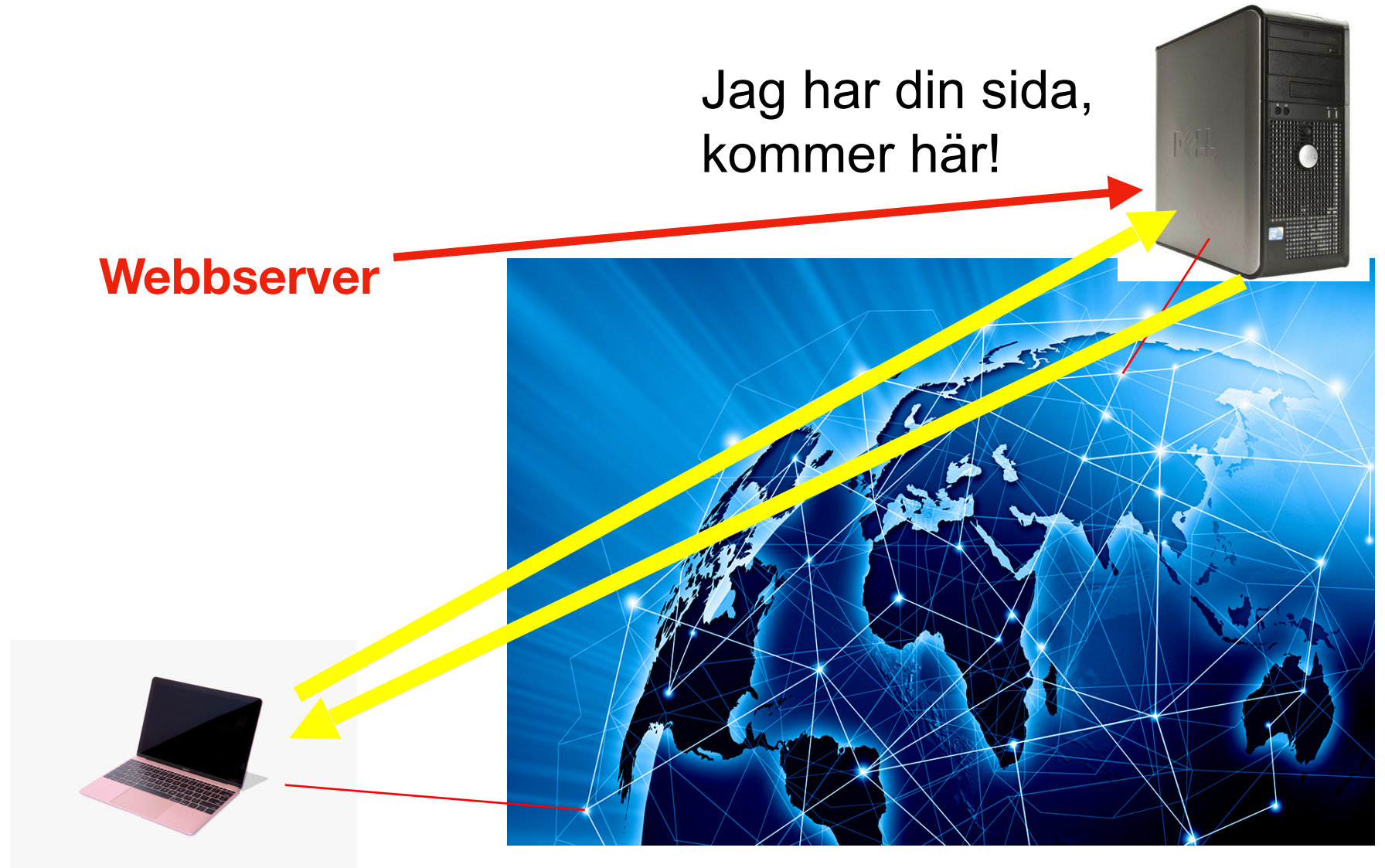
www.aftonbladet.se

104.20.55.70

Jag har din sida,
kommer här!

Webbserver

Mickes
dator



Lokal utvecklingsmiljö

- Webbserver
 - Lyssnar på anrop
 - Avgör vilken typ av resurs (fil) som efterfrågas.
 - Statiska filer (html, css, js, bilder) skickas tillbaka direkt.
 - PHP-filer måste *parsas* först.

Lokal utvecklingsmiljö

- Databasserver
 - Hanterar databas.
 - Kan oftast kommunicera med webserver.

Lokal utvecklingsmiljö

- LAMP
 - Linux
 - Apache
 - MySQL
 - PHP
- Finns i många varianter.
 - [https://en.wikipedia.org/wiki/LAMP \(software bundle\)](https://en.wikipedia.org/wiki/LAMP_(software_bundle))
- Populär variant för Windows: XAMPP
 - <https://www.apachefriends.org/>
- Mac: MAMP
 - <https://www.mamp.info/en/>

Lokal utvecklingsmiljö

- Annan variant: Virtuellt dator
 - VirtualBox - <https://www.virtualbox.org/>
 - Låter användare installera en virtuellt dator på datorn.
 - Vagrant - <https://www.vagrantup.com/>
 - Låter användare skipta ihop utvecklingsmiljöer på en virtuellt dator.
 - <https://puphpet.com/>

Lokal utvecklingsmiljö

- En tredje variant: En annan virtuell dator
 - Docker - <https://www.docker.com/>
 - Typ som VirtualBox men med annan teknik.

Uppgift

- Installera en lokal utvecklingsmiljö.
- När ni kan skapa en sida t ex index.php och få den att skriva ut "Hello world" med funktionen echo är ni klara.

Sammanfattning

- Formalia
- Variabler
- Utskrift
- Kontrollstrukturer
- Utvecklingsmiljö

Till nästa gång

- Se till att ni har en lokal utvecklingsmiljö.
- Gör övningarna tills ni behärskar dem.

Tack för idag!

Föreläsning 01 - Grunderna i PHP

Mikael Olsson
mikael.olsson@emmio.se
076-174 90 43

