

Introduktion till PHP

Designprojekt i webbt teknik

Montathar Faraon

Översikt

Definition av PHP, historiska trender, utvecklingsmiljö och manual

1

Innehåll

Indentering, kommentering, variabler, konstanter och vektorer

2

Utskrift av innehåll

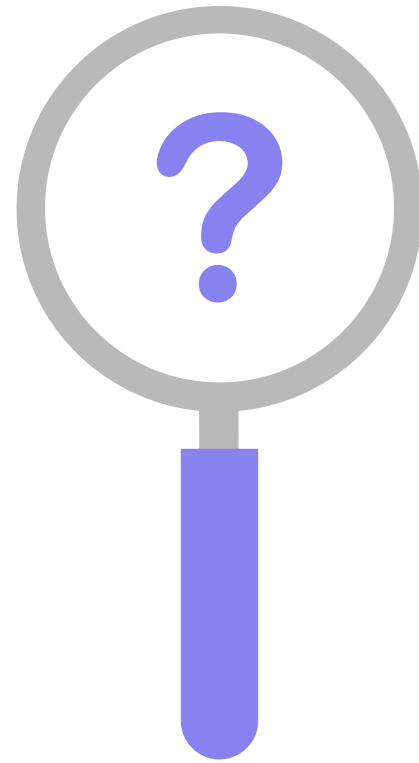
Echo & print, enkla & dubbla citationstecken, operatorer, villkorssatser

3

Övrigt

Frågor?

4



Definition av PHP

► Vad är PHP?

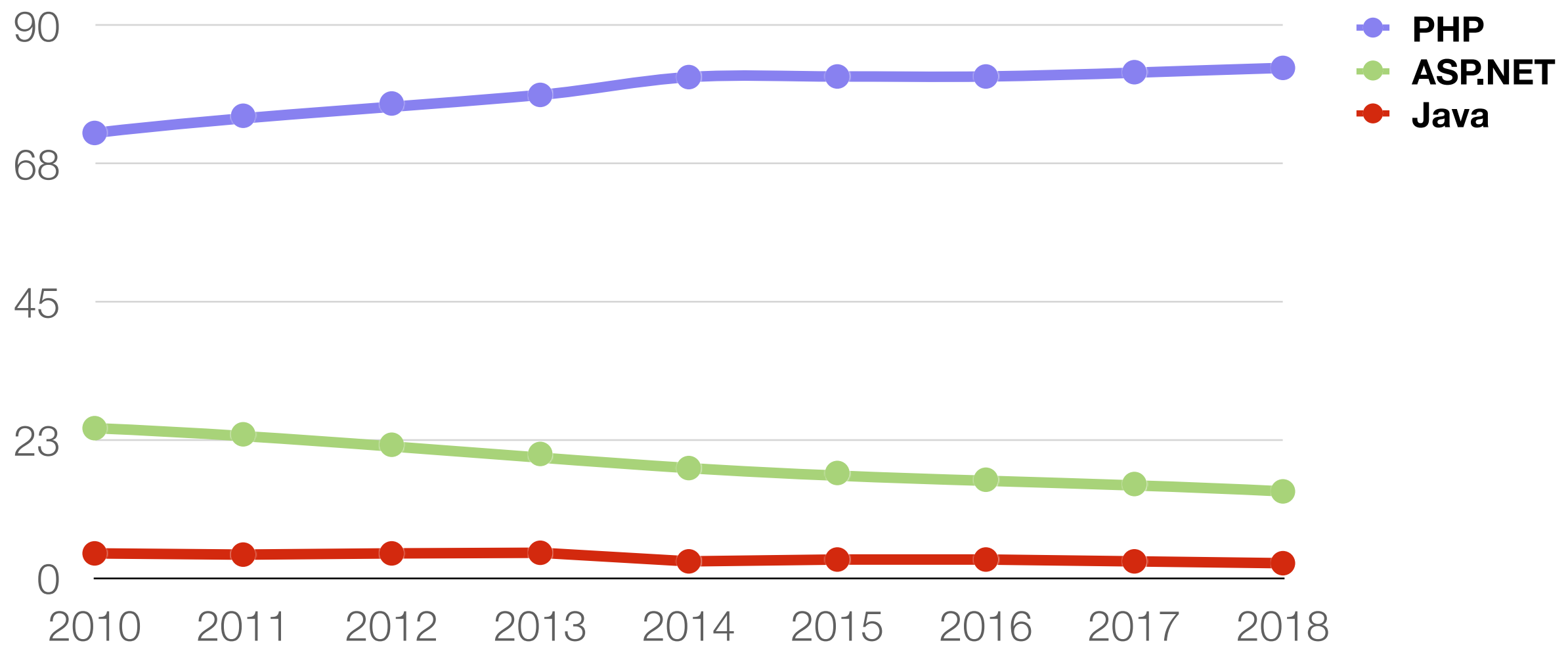
- PHP är en rekursiv akronym och står för:
 - **PHP**: **H**ypertext **P**reprocessor
- Utvecklades av Rasmus Lerdorf i juni 1995.
- **Användningsområden**
 - **Webbutik**. PHP används för att skapa och använda elektronisk handel som gör det möjligt för kunder att köpa produkter såväl som tjänster.
 - **Socialt nätverk**. PHP är en av byggstenarna i världens största sociala nätverk Facebook med två miljarder användare.
 - **Blogg**. Världens största bloggverktyg heter WordPress och är skrivet i PHP.



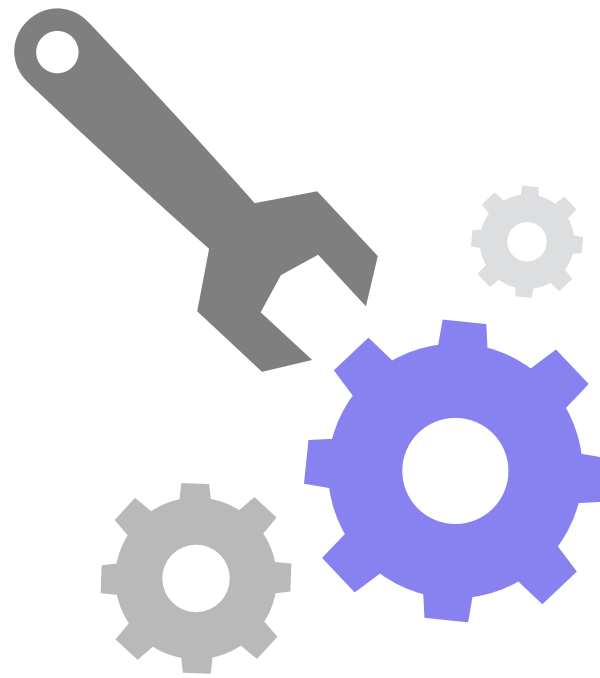
Historiska trender

► Historiska trender

- Diagram visar historiska trender i användning av skriptspråk sedan januari 2010.



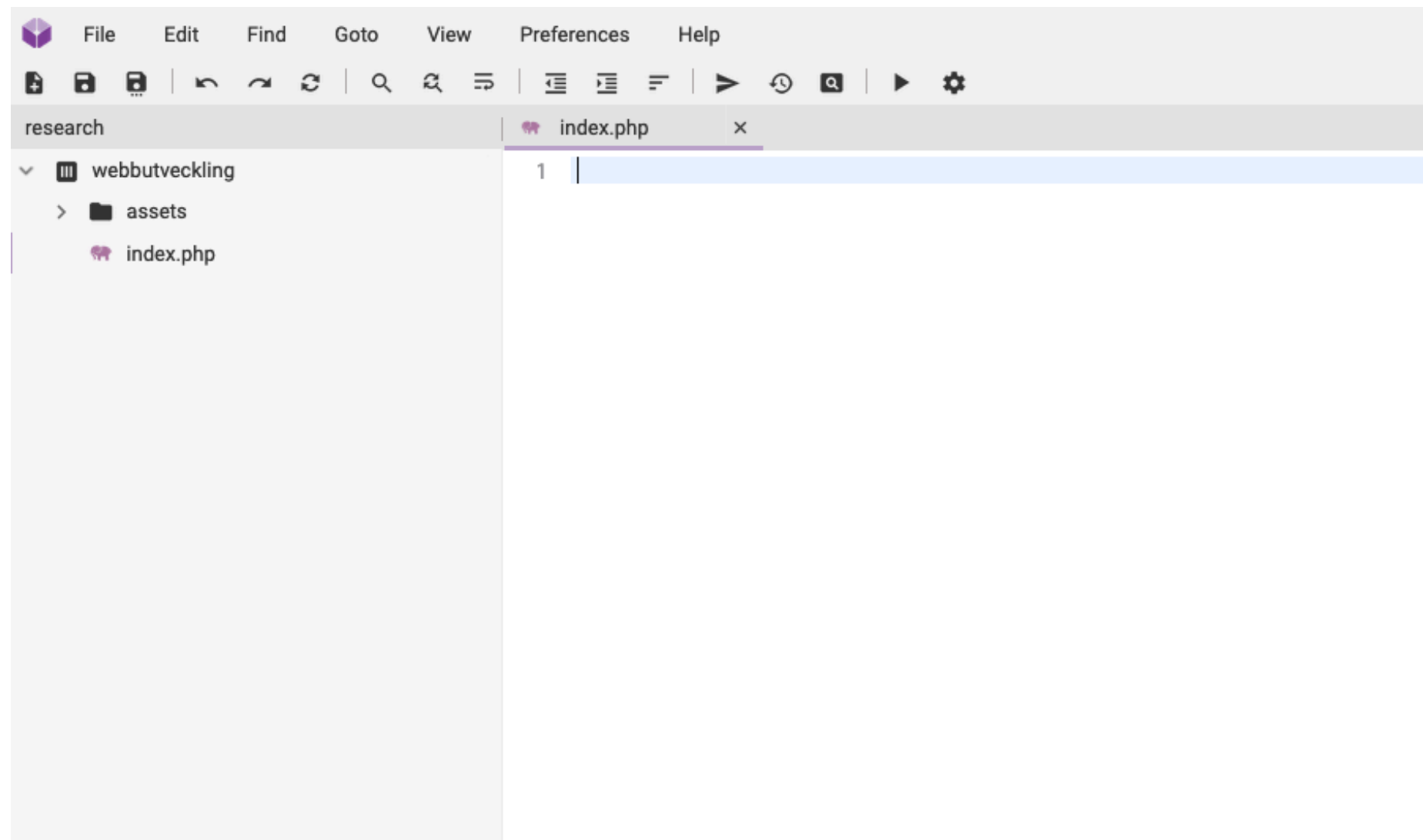
Källa: [W3Techs Historical Trend](#)

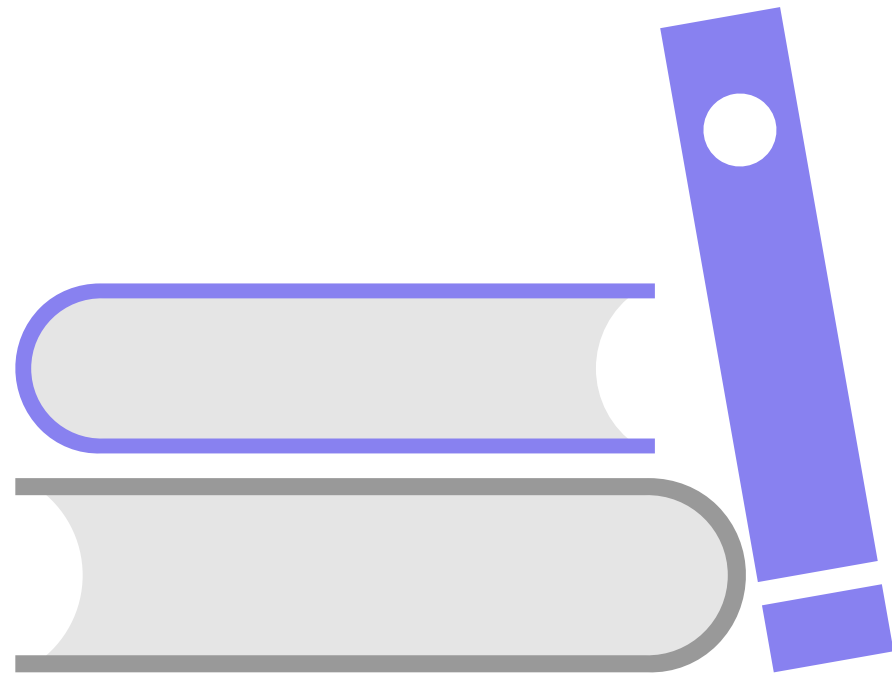


Utvecklingsmiljö

► Utvecklingsmiljö

- Codeanywhere



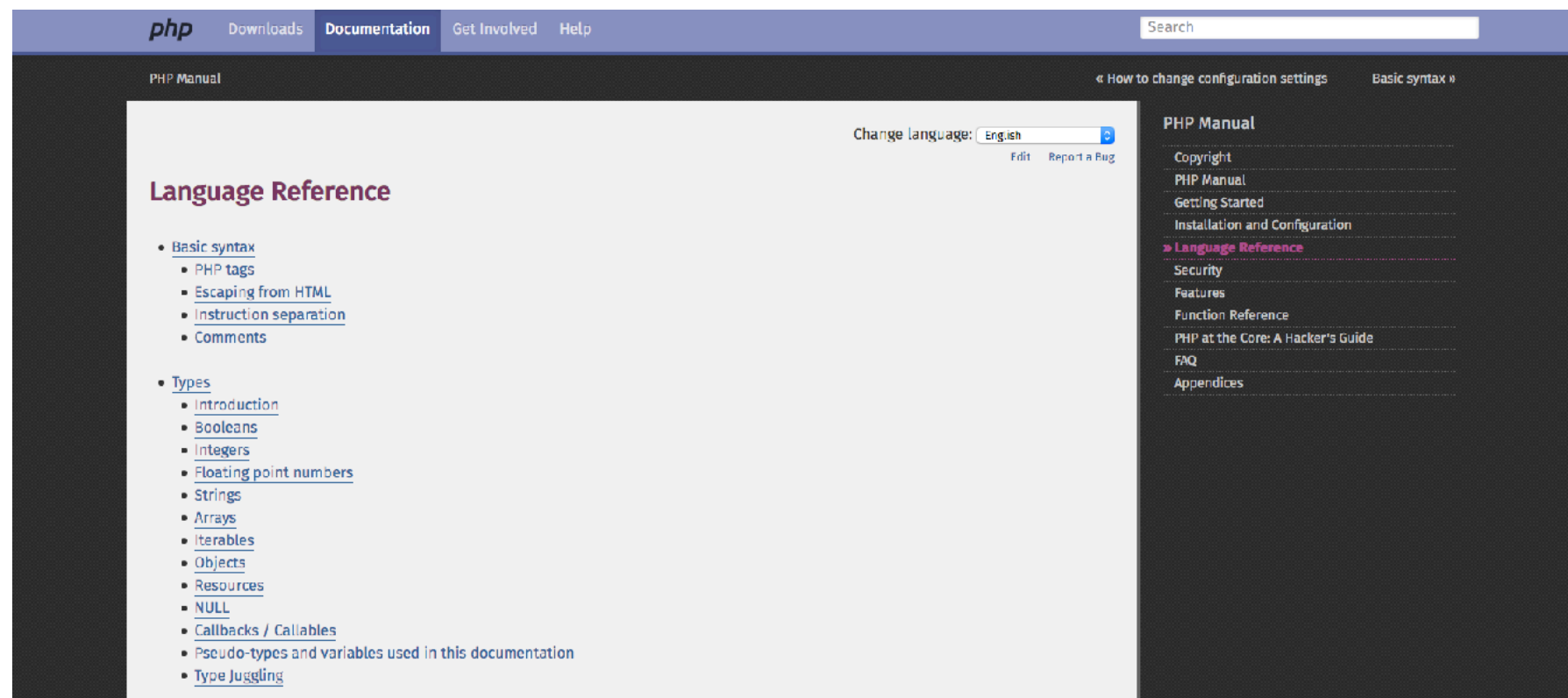


Manual för PHP

► Manualen för PHP

- Finns i två versioner:

- På nätet: <http://php.net/manual/en/langref.php>
- I nedladdningsbart format: <http://php.net/download-docs.php>



Översikt

Definition av PHP, historiska trender, utvecklingsmiljö och manual

1

Innehåll

Indentering, kommentering, variabler, konstanter och vektorer

2

Utskrift av innehåll

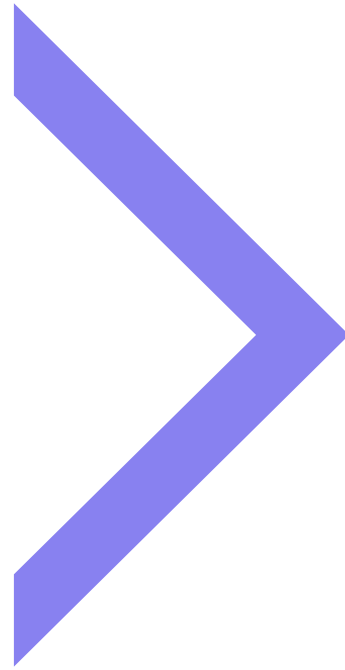
Echo & print, enkla & dubbla citationstecken, operatorer, villkorssatser

3

Övrigt

Frågor?

4



Indentering

► Indentering

- Definition

- **Indentering** (eng. *indent*) innebär **indrag** och syftar till att skapa en struktur för kod som är sammanhängande med varandra.

- Flesta företag har en standard hur kod ska skrivas.
- **PEAR-standard**en tillämpas i de flesta fallen - <http://pear.php.net>
- Varför tillämpar man en standard?
- Indentering görs med **tabb-knappen**:  i Windows och  i OS X.

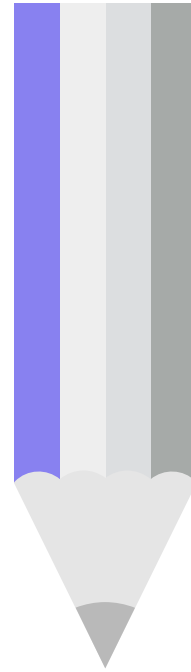
► Indentering

✓ Med indentering:

```
1 <?php
2     // Skapa och tilldela värdet 10
3     $counter = 10;
4
5     // Iterera och skriv ut värdet 1 till 9
6     for ($i = 0; $i < $counter; $i++) {
7         // Kontrollera om värdet är 5
8         if ($i == 5) {
9             echo '<p>Detta är ett annat meddelande för siffran 5.</p>';
10        }
11        else {
12            echo '<p>Detta är siffran ' . $i . ' .</p>';
13        }
14    }
15 ?>
```

✗ Utan indentering:

```
1 <?php
2 // Skapa och tilldela värdet 10
3 $counter = 10;
4
5 // Iterera och skriv ut värdet 1 till 9
6 for ($i = 0; $i < $counter; $i++) {
7 // Kontrollera om värdet är 5
8 if ($i == 5) {
9 echo '<p>Detta är ett annat meddelande för siffran 5.</p>';
10 }
11 else {
12 echo '<p>Detta är siffran ' . $i . ' .</p>';
13 }
14 }
15 ?>
```



Kommentarer

► Kommentarer

- Definition

- **Kommentarer** innebär att förklara *vad* som händer i koden och syftar till att göra det enklare för andra att sätta sig in i koden.

- Tre sätt att kommentera kod:

1. Enradiga kommentarer (vanligast)

```
// Detta är en kommentar
```

2. Enradiga kommentarer

```
# Detta är en kommentar
```

3. Flerradiga kommentarer

```
/*  
    Detta är en lång kommentar som består  
    av flera olika rader. Syftet är att ge  
    en bättre förklaring till koden.  
*/
```




Variabel

► Variabel

- Definition

- En variabel kan ses likt en **behållare** för att lagra en viss typ av data.

- Regler för tillämpning

- En variabel **inleds alltid** med ett **dollar-tecken (\$)**.
- Giltiga tecken är **bokstäver**, **siffror** och **understreck**. Variabelnamn får aldrig inledas med en siffra.
- Deklaration av en variabel **avslutas alltid med ett semi-kolon (;)**.
- PHP är **skiftlägeskänsligt**; det spelar roll om det är stora och små bokstäver.
- En variabel kan tillämpa **svenska bokstäver (åäö)** men bör undvikas.

► Variabel

- Det finns olika typer av data och de grundläggande i PHP är:

TYP	VÄRDE	EXEMPEL
Boolesk	SANT eller FALSKT	<code>\$logged_in = true;</code>
Heltal	17	<code>\$integer = 17;</code>
Flyttal	9,1	<code>\$float = 9.10;</code>
Sträng	Jag tycker om PHP.	<code>\$string = 'Jag tycker om PHP';</code>
Vektor	1, 3, Ann, Karl	<code>\$multiple = array(1, 3, 'Ann', 'Karl');</code>

Konstant

► Konstant

- Definition

- En **konstant** (eng. *constant*) är en variabel som **inte kan förändras**.

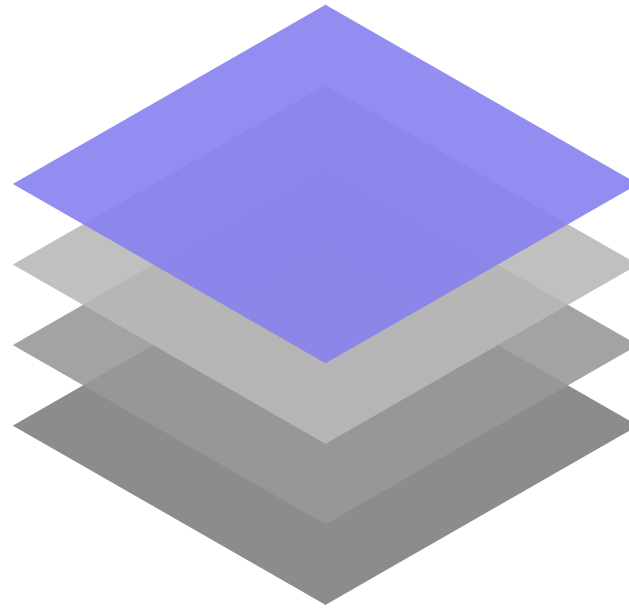
- Syntax

- `define(namn, värde);`

- Exempel

- `define(NAME, 'Anders Karlsson');`

```
1 <?php
2     // Konstant definieras och tilldelas värdet Anders Karlsson
3     define(NAME, 'Anders Karlsson');
4
5     // Utskrift av konstant görs
6     echo NAME;
7 ?>
```



Vektor

► Vektor

- Definition

- En **vektor** (eng. *array*) är en variabel som kan lagra **ett eller flera värden**.

- Praktisk om du vill att flera värden ska hänga ihop och inte delas i flera variabler.

- Syntax

- `$myArray = array(värden);`

- Exempel

- `$cars = array('Volvo', 'Mercedes', 'BMW');`

```
1 <?php
2     // Vektor definieras och tilldelas tre värden för bilmärken
3     $cars = array('Volvo', 'Mercedes', 'BMW');
4 ?>
```

Översikt

Definition av PHP, historiska trender, utvecklingsmiljö och manual

1

Innehåll

Indentering, kommentering, variabler, konstanter och vektorer

2

Utskrift av innehåll

Echo & print, enkla & dubbla citationstecken, operatorer, villkorssatser

3

Övrigt

Frågor?

4

I eller II

Utskrift av innehåll

► Enkla och dubbla citationstecken

- Vilket ska man använda vid arbete med PHP?

```
1 <?php
2     // Deklaration av en sträng
3     $string = 'Hello World';
4
5     // Kommer att skriva ut $string
6     echo '$string';
7
8     // Kommer att skriva ut Hello World
9     echo "$string";
10 ?>
```

- Använd enkla citationstecken.
 - **Minskad exekveringstid.** Dubbla citationstecken utvärderar innehållet först och skriver sedan ut det (längre tid). Enkla skriver ut innehållet direkt.
 - **Ökad prestanda.** När vi arbetar med tjänster som Facebook finns det fördelar.
 - **Ökad läsbarhet.** Lättare att läsa kod med enkla citationstecken.

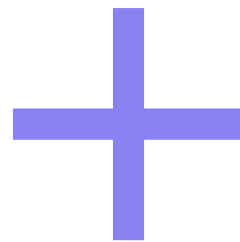
► Enkla och dubbla citationstecken

- Praktiskt exempel

```
1 <?php
2     // Deklaration av strängar
3     $text = 'Google is a search engine';
4     $link = 'http://www.google.com';
5
6     // Utskrift med enkla citationstecken
7     echo '<a href="'. $url .' ">'. $text .'</a>';
8
9     // Utskrift med dubbla citationstecken
10    echo "<a href=\"$url\">$text</a>";
11 ?>
```

- För- och nackdelar

- ✓ Du slipper **escape-tecken** som \ vid arbete med enkla citationstecken.
- ✓ Enkla citationstecken gör att du kan **tillämpa HTML-kod utan modifiering**.
- ✗ Ökad användning av **tecknet punkt (strängkonkatenering)** för att lägga ihop strängar och variabler.



Operatorer

► Operatorer

- Definition
 - Operatorer används för att "operera på värden".
- I beräkningen $1 + 2 = 3$ är **plus- (+)** och **tilldelningstecknet (=)** operatorer.
- **6 operatorer:**
 - Jämförelse
 - Aritmetiska
 - Logiska
 - Upp- och nedräkning
 - Sträng
 - Tilldelning

► Operatorer

- **Jämförelse** (lika med, identisk med, skild från, ...)
 - 2 == 4 ger falskt (lika med; kontrollerar värde)
 - "3" === 3 ger falskt (identisk med; kontrollerar värde och datatyp)
 - 2 != 4 ger sant (skild från)

► Operatorer

- Aritmetiska (addition, subtraktion, ...)
 - $\$a = 1;$
 - $\$b = 1;$
 - Då blir $\$a + \$b = 2;$

► Operatorer

- **Logiska** (och, inte, ...)

- Om \$a = 18;
- \$a > 10 **&&** \$a < 22 ger sant (**och**; båda villkor måste utvärderas till sant)
- (**!** (\$a > 20)) ger sant (**inte**; om det inte är falskt, så är det sant)

► Operatorer

- Upp- och nedräkning (förhandsökning, efterhandsökning, ...)
 - Om $a = 5$;
 - Utskrift av $++a$ ger 6 (förhandsökning; ökar först / skriver ut sedan)
 - Utskrift av $a++$ ger 5 (efterhandsökning; skriver ut först / ökar sedan)

► Operatorer

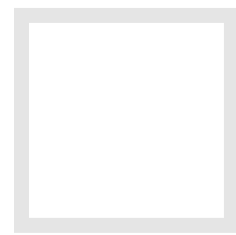
- **Sträng** (konkatenering, ...)
 - **Konkatenering** innebär **att slå ihop** och görs med en punkt.
 - `$a = 'Helsing';`
 - `$b = $a . 'borg';`
 - Utskrift av `$b` ger Helsingborg

► Operatorer

- **Tilldelning** (tilldelning, sammanslagning, ...)
 - `$a = 1;` (**tilldelning**; variabeln får ett värde)
 - `$b = 'Anders';`
 - `$b .= ' Karlsson';` (**sammanslagning**; variabeln slår ihop värden)
 - Utskrift av `$b` ger Anders Karlsson



SANT



FALSKT

Villkorssatser

► Villkorssatser

- If-sats
 - Utför handlingar beroende på olika kriterier.
 - Inleds med **if**, följt av ett **villkor inom parentes** och slutar med **ett kodstycke** inom klammerparentes.



► Villkorssatser

- If-else sats

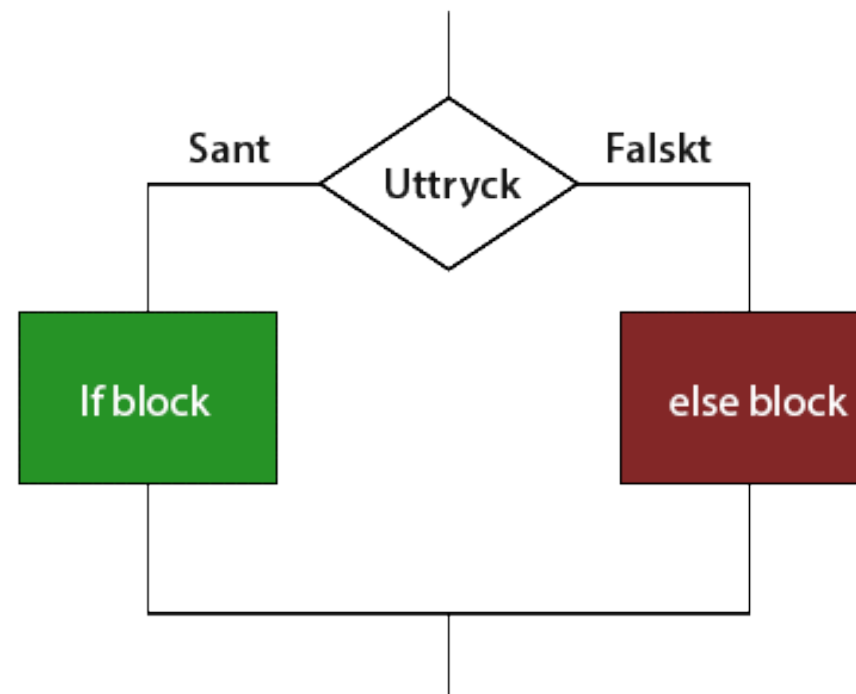
- Exekverar en **viss kod** om ett villkor är **sant**; en **annan kod** om det är **falskt**.
- **Om** klockan är 12 skriv ut “Lunchdags!”, **annars** skriv ut “Inte lunch ännu!”

If (*uttryck*):

Kodstycke som exekveras
endast om uttrycket är *sant*

else :

Kodstycke som exekveras
endast om uttrycket är *falskt*



► Villkorssatser

- Exempel if-else sats
 - Undersöka villkoret “2 är större än 1”.
- Kodexempel

```
1 <?php
2     // Kontrollerar om 2 är större än 1
3     if (2 > 1) {
4         echo 'Sant';
5     }
6     else {
7         echo 'Falskt';
8     }
9
10 ?>
```

- Resultatet kommer bli **sant**.

Frågor

