

Backend programmeren

XAMPP tech-stack

Derde druk – versie 2021

Gabriel Sánchez Cano

Boom beroepsonderwijs
info@boomberoepsonderwijs.nl
www.boomberoepsonderwijs.nl

Auteur: Gabriel Sánchez Cano
Redactie en opmaak: Henk Pel
Titel: Backend programmeren. XAMPP tech-stack. Versie 2021
ISBN 978 90 372 5907 0
Derde druk / eerste oplage
© Boom beroepsonderwijs 2021

Behoudens de in of krachtens de Auteurswet gestelde uitzonderingen mag niets uit deze uitgave worden verveelvoudigd, opgeslagen in een geautomatiseerd gegevensbestand, of openbaar gemaakt, in enige vorm of op enige wijze, hetzij elektronisch, mechanisch, door fotokopieën, opnamen of enige andere manier, zonder voorafgaande schriftelijke toestemming van de uitgever.

Voor zover het maken van reprografische verveelvoudigingen uit deze uitgave is toegestaan op grond van artikel 16h Auteurswet dient men de daarvoor wettelijk verschuldigde vergoedingen te voldoen aan de Stichting Reprorecht (www.reprorecht.nl). Voor het overnemen van gedeelte(n) uit deze uitgave in compilatiewerken op grond van artikel 16 Auteurswet kan men zich wenden tot de Stichting PRO (www.stichting-pro.nl).

De uitgever heeft ernaar gestreefd de auteursrechten te regelen volgens de wettelijke bepalingen. Degenen die desondanks menen zekere rechten te kunnen doen gelden, kunnen zich alsnog tot de uitgever wenden.

Door het gebruik van deze uitgave verklaart u kennis te hebben genomen van en akkoord te gaan met de specifieke productvoorwaarden en algemene voorwaarden van Boom beroepsonderwijs, te vinden op www.boomberoepsonderwijs.nl.

Inhoud

Voorwoord 1

1 PHP 3

- 1.1 Inleiding PHP 3
- 1.2 De ontwikkelomgeving 7
- 1.3 De PHP-programmeertaal 10
- 1.4 Datatypen 14
- 1.5 Array-methodes 20
- 1.6 Array-methodes [vervolg] 27
- 1.7 De if-opdracht 35
- 1.8 \$_POST-variabelen 42
- 1.9 String-methodes 47
- 1.10 Een simpel server-side winkelmandje 56
- 1.11 Switch 60
- 1.12 Functions 64
- 1.13 Externe functions 70
- 1.14 Controlestructuren 74
- 1.15 For-lus [vervolg] 80
- 1.16 De foreach-lus 81
- 1.17 De while-lus 84
- 1.18 Constanten en globale variabelen 89
- 1.19 Date-methodes 95
- 1.20 Cookies in local storage 103
- 1.21 JSON-objects en literals 108
- 1.22 Project blog 113

2 Gegevensanalyse 123

- 2.1 Top-down-analyse 123
- 2.2 Normaliseren 131
- 2.3 Normalisatieprocessen 134
- 2.4 Normalisatie [vervolg] 140
- 2.5 Normalisatie [vervolg] 142
- 2.6 Normalisatie [vervolg] 143

3 Inleiding MySQL 147

- 3.1 MySQL 147
- 3.2 De database-administrator 153
- 3.3 SELECT-clausules 158
- 3.4 UPDATE / INSERT / DELETE 164
- 3.5 Sub-queries 168
- 3.6 JOIN 172
- 3.7 CREATE / DROP 178
- 3.8 ALTER TABLE 181
- 3.9 String-methodes 185
- 3.10 Date-methodes 193
- 3.11 Stored programma's 198
- 3.12 CURSORS en HANDLERS 208
- 3.13 FUNCTIONS / VIEWS / TRIGGERS 211
- 3.14 Inleiding PHP Data Objects (PDO) 215
- 3.15 PDO-processen 217
- 3.16 Project Schiphol Meldpunt 224

4 Gestructureerd programmeren 227

- 4.1 Veilig coderen 227
- 4.2 Project webshop 234
- 4.3 De applicatie-header 238
- 4.4 Authenticatie 242
- 4.5 Aanmelden 246
- 4.6 E-mail-instellingen 250
- 4.7 E-mailen 253
- 4.8 Wachtwoord vergeten? 256
- 4.9 Multi-factor-authenticatie 259
- 4.10 Autorisatie met rollen 262
- 4.11 Profiel editen 265
- 4.12 Data-gestuurd programmeren 268
- 4.13 Dynamische webpagina's 271
- 4.14 Het winkelmandje 275
- 4.15 Order plaatsen 280
- 4.16 Order-factuur 283
- 4.17 De administrator 285
- 4.18 Album editen 287
- 4.19 Album update 289
- 4.20 Album delete 290
- 4.21 Album add 291
- 4.22 Rapporten genereren 294

4.23	Rapporten genereren in PHP	297
4.24	Voorraad rapport	301
4.25	MPDF installeren	302
4.26	PDF genereren met MPDF	304
4.27	Voorraad-rapport in PDF	306
4.28	Session-beheer	308
4.29	Project Webshop	309
4.30	Project Vacaturebank	310

Register 323

Voorwoord

Opbouw

Het boek bestaat uit vier hoofdstukken:

- 1 Inleiding PHP
- 2 Gegevensanalyse
- 3 Inleiding MySQL
- 4 Gestructureerd programmeren

Hoofdstukken 1, 2 en 3 kunnen parallel gegeven worden, daarna hoofdstuk 4. Dit boek bestaat uit 230 lesuren verdeeld over 70 lesblokken van 2 uur en vier zelfstudieprojecten van in totaal 90 uur.

Bij de opbouw is gebruikgemaakt van de taxonomie van Romiszowski waar onderscheid wordt gemaakt tussen kennis (het opslaan van informatie) en vaardigheden (acties uitvoeren om een doel te bereiken).

Elk lesblok eindigt met een kennistoets en een vaardigheid-lab. Elk hoofdstuk eindigt met een zelfstudieproject.

- **Kennistoetsen:** kennis van begrippen en procedures.
- **Vaardigheid-labs:** reproductieve vaardigheid, acties uitvoeren om een doel te bereiken.
- **Zelfstudieprojecten:** productieve vaardigheid. In tegenstelling tot reproductieve vaardigheden doen productieve vaardigheden een beroep op de creativiteit en planningsvaardigheden van de student; ze gaan gepaard met (complexe) beslissingsvorming op bewust of onderbewust niveau. De student moet de geleerde informatie spontaan toepassen in nieuwe situaties, waarin niet van tevoren geoefend is. Er moeten nieuwe oplossingen voor nieuwe problemen bedacht worden.

Dit boek is met de grootst mogelijke zorg geschreven. De juistheid en volledigheid van de gegevens kunnen echter niet worden gegarandeerd. De auteur en uitgever aanvaarden geen aansprakelijkheid voor schade, van welke aard dan ook, die het directe of indirecte gevolg is van handelingen zoals onethisch hacken.

Ik wil al mijn studenten en collegae bedanken voor hun feedback tijdens het maken van dit boek. Speciale dank aan Bart Schrap voor zijn zorgvuldige feedback en commentaar. Ik heb dit boek met veel plezier geschreven en ik hoop dat zowel de studenten als docenten er met veel plezier mee zullen werken.

1 PHP

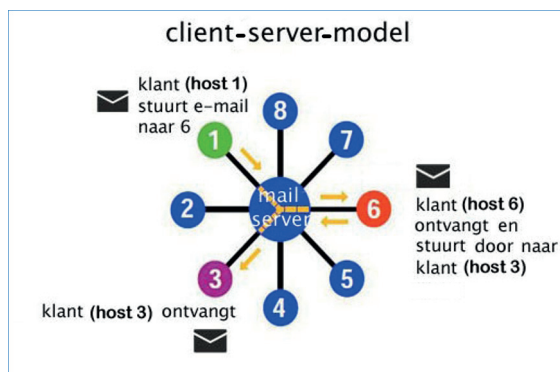
Voorkennis

Kennis van en ervaring met web-applicatieontwikkeling is vereist, met name html, CSS en JavaScript.

1.1 Inleiding PHP

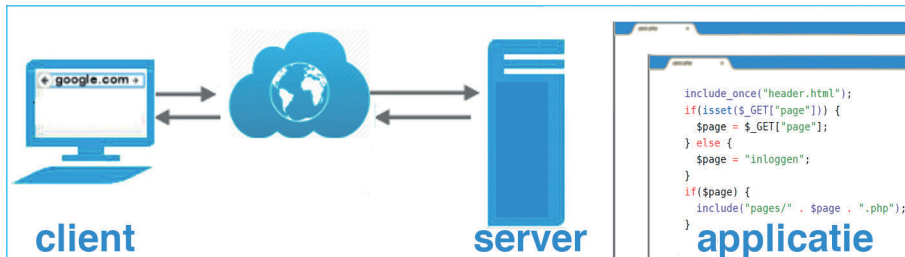
Lesblok 1.1		
	Tijdsduur	2 uur
	Doel	XAMPP installeren en instellen
	Benodigdheden	XAMPP tech-stack

Het client-server-model is een model waar twee computers in samenwerking twee of meer programma's uitvoeren, bijvoorbeeld e-mail of internetbankieren. Dit model werkt als volgt: een cliëntprogramma doet een aanvraag (request) bij het serverprogramma. Het serverprogramma voert hierop handelingen uit en geeft informatie terug aan de cliënt. Bijvoorbeeld, een webserver ontvangt requests van verschillende webbrowsers. De server voert acties uit en geeft de informatie terug in de vorm van webpagina's.



Figuur 1.1 Client-server-model

Dit client-server-model gebruiken we ook voor het realiseren van web-applicaties. In dit hoofdstuk houden we ons bezig met het opbouwen van web-applicaties met PHP.



Figuur 1.2 Een web-applicatie

Wat is PHP?

PHP (oorspronkelijk Personal Home Page) is een server-side scripting-taal, speciaal ontworpen voor het web. Je kunt PHP-code binnen een html-pagina embedden. De PHP-code wordt uitgevoerd door de webserver wanneer iemand jouw webpagina bezoekt. PHP is ontworpen door Rasmus Lerdorf in 1994. Het is een open-source product en gratis te downloaden. De homepage voor PHP vind je hier: www.php.net.

PHP is een scripting-taal die geïnterpreteerd wordt door de PHP-interpreter. Je hoeft PHP niet te compileren zoals C of Java. Een Java-programma moet eerst gecompileerd worden door een compiler en daarna uitgevoerd. Voorbeelden van scripting-talen zijn PHP en JavaScript.

Ontwikkelomgeving

Om te beginnen moet je een webserver, een datasever en een mailserver op je computer installeren. Een webserver biedt webpagina's aan ('serveert') de browsers van de gebruikers aan. PHP draait op de Apache-webserver. Om PHP-scripts te kunnen draaien moet je eerst de Apache-webserver op je computer installeren. De populairste open-source tech-stack (software-bundel) voor PHP is XAMPP. De tech-stack bestaat voor Multi-platform Apache MySQL PHP en Pearl. Je kunt XAMPP op een web-host installeren. Maar voor het ontwikkelen en testen van je websites kun je XAMPP op je eigen pc of laptop installeren. XAMPP kan gedownload worden voor Windows, MacOS of Linux:

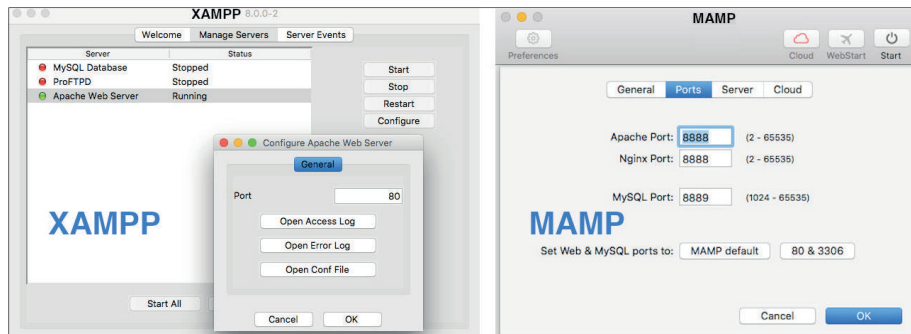
- WAMP voor Windows
- MAMP voor MacOS
- LAMP voor Linux
- XAMPP is multiplatform

Je kunt de tech-stack voor Windows downloaden vanaf de site: www.apachefriends.org.

Voor MacOS kun je MAMP downloaden vanaf de site:
www.mamp.info/en/downloads/

Voor Linux kun je MAMP installeren vanaf de site:
www.linux.com/learn/easy-lamp-server-installation

Als je XAMPP downloadt en installeert voor je operating system start je XAMPP door op de icoon van XAMPP-control-panel te klikken (zie figuur 1.3), waarna de XAMPP Control Panel Application verschijnt.



Figuur 1.3 XAMPP Control Panel en MAMP Control Panel

In het eerste deel van het XAMPP controlepaneel kun je de Apache- en MySQL-services zien. In het tweede deel zien we de *Start*- en *Stop*-knoppen voor de services. In het derde deel zien we onder andere de *Configuratie*-knop waar we de installatie kunnen instellen.

We zien dat de Apache-server in XAMPP draait in port 80. In MAMP draait deze in port 8888.

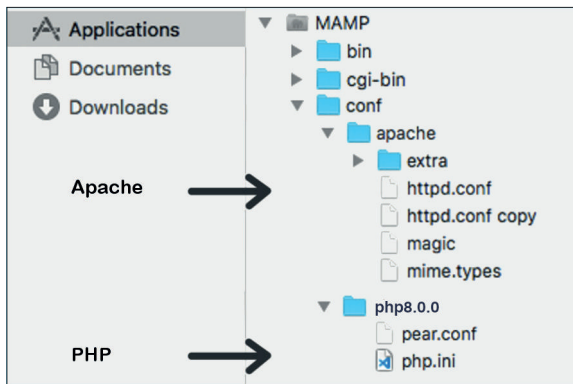
Om te testen of de installatie goed verlopen is typ je in het adresvakje van de browser het volgende:

`localhost:80` voor XAMPP of

`localhost:8888` voor MAMP

Nu moet de `index.php`-pagina verschijnen.

De mappenstructuur van XAMPP verschilt per operating system. De mappenstructuur van, in dit geval, een MAMP-installatie, ziet er als volgt uit:



Figuur 1.4 Mappenstructuur van de installatie

Belangrijk voor ons zijn de volgende mappen en bestanden:

httpd.conf: in *Applications/MAMP/conf/apache/extra* is dit het bestand voor het configureren van de Apache-server.

php.ini: in *Applications/MAMP/conf/php8.0.0* is dit het bestand waar de standaardinstellingen van PHP te vinden zijn.

Kennistoets 1.1: XAMPP instellingen

Bestudeer de theorie van dit lesblok en maak de volgende kennistoets.

Toets 1.1	
	1. Wat is het client-server-model?
	2. Wat is een tech-stack?
	3. Welke modules heeft de XAMPP tech-stack?
	4. Wat staat in de conf-map van XAMPP/MAMP?
	5. Waar vind je het php.ini-bestand?
	6. Wat is PHP?
	7. Wat is het verschil tussen een scripting- en een gecompileerde taal?
	8. Noem twee scripting-talen.
	9. Noem twee compileertalen.
	10. Wie is Rasmus Lerdorf?

Vaardigheid-lab 1.01: XAMPP-instellingen

Het kan gebeuren dat bij de installatie van XAMPP problemen ontstaan. In deze lab-opdracht ga je controleren of de configuratiebestanden correct ingesteld zijn. Soms kun je bij het opstarten van de Apache-server de volgende melding krijgen: Port 80 geblokkeerd. Dit betekent dat software in je computer gebruikmaakt van poort 80, bijvoorbeeld Skype.

Stap 1: Set de software uit en start XAMPP opnieuw.

Stap 2: Als dit niet werkt, open het MAMP/conf/apache/httpd.conf-bestand en zoek de regel: `Listen 80`. Hier kun je een nieuwe poort, bijvoorbeeld 8080, voor Apache gebruiken. Als je een nieuwe poort gebruikt moet je dit voortaan in je browser als volgt aangeven: `http://localhost:8080`
Tijdens het maken van je PHP-opdrachten is het handig om alle foutmeldingen te kunnen zien.

Stap 3: Open MAMP/conf/php8.0.0/php.ini en controleer dat de regel er zo uitziet:

```
commando: error_reporting = E_ALL
```

Om je databases te beschermen kun je een nieuwe gebruiker en een nieuw wachtwoord aanmaken.

Stap 4: Ga naar: `http://localhost/phpmyadmin` en maak een nieuwe gebruiker aan voor je MySQL database-server.

1.2 De ontwikkelomgeving

Lesblok 1.2		
	Tijdsduur	2 uur
	Doel	Ontwikkelomgeving installeren
	Benodigdheden	Code-editor en XAMPP

XAMPP wordt ook gebruikt voor het maken van web-applicaties met behulp van Content Management Systems (CMS) zoals Joomla en WordPress. Een CMS is software om web-applicaties te creëren en managen met weinig of geen programmeerervaring. We gebruiken geen CMS omdat we willen leren programmeren en onze eigen web-applicaties en databases (backend) gaan coderen.

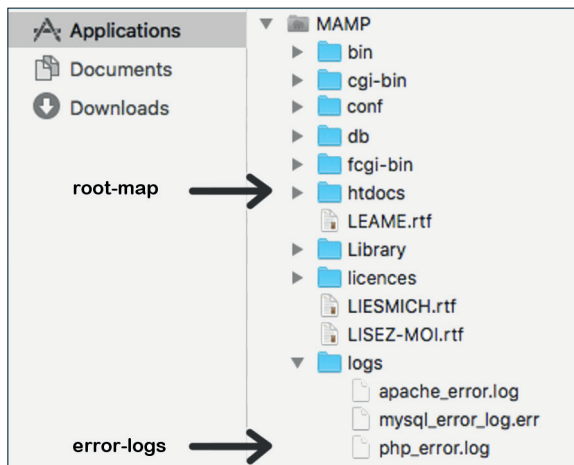
Code-editor

Om de PHP-opdrachten te kunnen coderen heb je een code-editor nodig. Als je nog geen favoriete code-editor hebt, download en installeer dan een gratis editor uit de volgende lijst:

- Sublime
- Brackets
- Visual Studio Code

Ontwikkelomgeving

Na de installatie van XAMPP zien we de volgende mappenstructuur. Bij het maken van je PHP-opdrachten zijn de volgende mappen belangrijk:



Figuur 1.5 Mappenstructuur van de installatie

htdocs: is de root-map van de te coderen applicatie en waar we onze PHP-scripts gaan plaatsen.

php_error.log: is het bestand in de map *logs* waar we onze codefouten kunnen traceren.

Eerste PHP-script

De webserver herkent een PHP-script aan de volgende tags:

```
<?php ... ?>
```

In het volgende voorbeeld zien we een blok PHP ingebed in html. Codeer de volgende code en sla het op als **index.php** in de map *htdocs*. In Windows moet dat de map *www* zijn.

```
<!DOCTYPE html>
<html lang="nl">
<head>
  <meta charset="utf-8">
  <title>testen</title>
</head>
<?php
  echo phpversion();
  phpinfo();
?>
<body>
</body>
</html>
```

Binnen een PHP-blok coderen we PHP-commando's met de juiste syntaxis. Syntaxis is de structuur en de regels van commando's in een computerprogrammeertaal. Bijvoorbeeld, om output weer te geven gebruik je echo als opdracht of commando. De syntaxis is:

```
echo "jouw tekst";
```

Alle instructies binnen de tags <?php en ?> eindigen met een puntkomma (;).

De `phpinfo()`-function in bovenstaand voorbeeld is een standaard-PHP-function die de webpagina met de PHP-instellingen weergeeft (functions worden behandeld in paragraaf 1.12).

Open je browser en typ

localhost:80 voor Windows

of

localhost:8888 voor MAC.

Als je geen pagina te zien krijgt, dan is het mogelijk dat je een typefout of codeerfout gemaakt hebt. Probeer dan de fouten te traceren door het **php_error.log**-bestand te openen in MAMP of door op *Open Error Log* te klikken in XAMPP. Maak hiervoor de volgende opzettelijke typefout.

Verander de regel: `phpinfo()`;

in: `phpinf()`;

En probeer het script opnieuw te draaien vanuit de adresbalk van je browser. Je krijgt de volgende foutmelding in **php_error.log** te zien:

```
[23-Jul-2018 01:50:13 Europe/Berlin] PHP Fatal error: Uncaught  
Error: Call to undefined function phpinf()
```

Kennistoets 1.2

Bestudeer de theorie van dit lesblok en maak de volgende kennistoets.

Toets 1.2	
	1. Wat is een Call to undefined function-error?
	2. Wat is de <code>phpinfo</code> -function?
	3. Waar kun je Apache-server-fouten traceren?
	4. Wat is de root-map van de Apache-server?
	5. Benoem alle error-logs van XAMPP.
	6. Wat is een CMS?
	7. Wat is Joomla?

	8. Wat is syntaxis?
	9. Wat doet de echo-opdracht?
	10. Waar kunnen we een PHP-script embedden in een html-script?

Vaardigheid-lab 1.02

Stap 1: Maak een nieuwe map *lessen* in de map *htdocs* van XAMPP.

Stap 2: Maak een PHP-script en sla het op in de map *lessen* als **lab1.02.php**.

Stap 3: Codeer een echo-opdracht met de volgende tekst:

“Dit is mijn eerste PHP-script”.

Stap 4: Voeg de `phpinfo()`-opdracht eraan toe.

Stap 5: Start XAMPP op.

Stap 6: Om je eerste PHP-script te kunnen testen, typ je het volgende in de adresbalk van je browser: `localhost/lessen/lab1.02.php`

1.3 De PHP-programmeertaal

Lesblok 1.3		
	Tijdsduur	2 uur
	Doel	Taalcomponenten
	Benodigdheden	Code-editor en XAMPP

Taalcomponenten

De PHP-syntaxis is hetzelfde als de syntaxis van andere programmeertalen, zoals C en Java. De volgende tabel is een samenvatting van de PHP-syntaxis:

Tabel 1.1 Taalcomponenten	
Taalcomponent	PHP-syntaxis
Openen en sluiten script-tags	<code><?php ?></code>
Block	<code>{ }</code>
Commentaarregel	<code>// commentaar</code>
Commentaarblok	<code>/* commentaar */</code>
Declareren variabelen	<code>\$mijnvar = 0;</code>
Declareren strings	<code>\$mijnstring = "tekst"</code>
Escape-teken \	<code>"filmtitel \"Serpico\" "</code>

Een tekst begint en eindigt met aanhalingstekens. Het escape-teken (\) gebruiken we binnen een tekst tussen aanhalingstekens wanneer we een teken letterlijk willen gebruiken. Bijvoorbeeld in de tekst "filmtitel \"Serpico\" " gebruiken we escape + aanhalingstekens (\") als onderdeel van de tekst, NIET voor het markeren van het einde van de tekst.

LET OP

Binnen dubbele aanhalingstekens kunnen we enkele aanhalingstekens gebruiken, dan is duidelijker wat bij elkaar hoort: "tekst 'tekst' ".

En binnen enkele aanhalingstekens kun je dubbele gebruiken als het nodig is:

'tekst "tekst" '.

PHP-variabelen

Wat is een variabele?

Variabelen kun je gebruiken om tijdelijk gegevens in op te slaan. Deze gegevens kunnen bijvoorbeeld bestaan uit getallen, stukken tekst of booleans oftewel waar-/niet-waar-gegevens. Ook kun je een reeks gegevens (array) in een variabele zetten. Alles wat je in PHP kunt aanmaken, kun je in een variabele zetten.

Een PHP-variabele is een data-container met een naam en een waarde. De naam van de variabele is het adres van de variabele in het computergeheugen. Op dit adres kun je data tijdelijk bewaren. Een variabele kun je je voorstellen als een laatje met een naam in het geheugen van de computer. Door middel van de naam kun je de inhoud van dat laatje bekijken of wijzigen.

Om een variabele te kunnen gebruiken moet je hem eerst aanmaken. Het aanmaken van een variabele, ook wel declareren genaamd, doe je door een dollarteken (\$) gevolgd door een zelf gekozen naam te tikken. Met het is-gelijk-aan-teken (=) kun je vervolgens een waarde aan de variabele toekennen. Tenslotte sluit je de regel af met een puntkomma zoals dat bij elke PHP-regel hoort.

Namen van variabelen

De naam van een variabele:

- begint altijd met het \$-teken en een letter;
- mag letters, getallen en underscores (_) bevatten;
- is hoofdlettergevoelig, bijvoorbeeld \$straat en \$Straat zijn twee verschillende variabelen.

Bij variabelen met twee woorden gebruiken PHP-programmeurs twee conventies:

- 1 underscores, bijvoorbeeld: mijn_auto
- 2 camelCase, bijvoorbeeld: mijnAuto

Bij camelCase begint het tweede woord met een hoofdletter. CamelCase is de meest gebruikte conventie.

Een variabele mag de volgende datatypen en waarden hebben:

Tabel 1.2 Datatypen		
Datatype	Betekenis	Voorbeeld
string	tekst	"Carl"
boolean	ja/nee	true/false
integer	geheel getal	19
float	getal met drijvende komma	19,99
array	verzameling	[1,2,3,4,5]

Om variabelen te kunnen verwerken gebruiken we operatoren. Hieronder zie je de belangrijkste rekenkundige operatoren:

Tabel 1.3 Rekenkundige operatoren		
Operator	Betekenis	Voorbeeld
++	plus 1	\$teller++
--	min 1	\$teller--
+	optellen	\$teller + 1
-	afrekken	\$teller - 1
*	vermenigvuldigen	\$som * 2
/	delen	\$bedrag / 2
.	tekst samenvoegen	"Error" . "01"

Hieronder declareren we drie variabelen:

```
$naam = "Carl";
$leeftijd = 16;
$leerplichtig = true;
```

De waarde van een variabele kan gewijzigd worden:

```
$leeftijd = $leeftijd + 2;
```

Gereserveerde sleutelwoorden in PHP

Gereserveerde sleutelwoorden in PHP hebben speciale betekenissen. Bijvoorbeeld, het woord *array* is de naam van de methode voor het creëren van arrays. We mogen geen gereserveerde woorden gebruiken bij het maken van onze eigen variabelen want dan krijg je een foutmelding. In de volgende tabel zien we de gereserveerde woorden in PHP.

Tabel 1.4 Gereserveerde sleutelwoorden				
<code>__halt_compiler()</code>	<code>abstract</code>	<code>and</code>	<code>array()</code>	<code>as</code>
<code>break</code>	<code>callable</code>	<code>case</code>	<code>catch</code>	<code>class</code>
<code>clone</code>	<code>const</code>	<code>continue</code>	<code>declare</code>	<code>default</code>
<code>die()</code>	<code>do</code>	<code>echo</code>	<code>else</code>	<code>elseif</code>
<code>empty()</code>	<code>enddeclare</code>	<code>endfor</code>	<code>endforeach</code>	<code>endif</code>
<code>endswitch</code>	<code>endwhile</code>	<code>eval()</code>	<code>exit()</code>	<code>extends</code>
<code>final</code>	<code>finally</code>	<code>for</code>	<code>foreach</code>	<code>function</code>
<code>global</code>	<code>goto</code>	<code>if</code>	<code>implements</code>	<code>include</code>
<code>include_once</code>	<code>instanceof</code>	<code>insteadof</code>	<code>interface</code>	<code>isset()</code>
<code>list()</code>	<code>namespace</code>	<code>new</code>	<code>or</code>	<code>print</code>
<code>private</code>	<code>protected</code>	<code>public</code>	<code>require</code>	<code>require_once</code>
<code>return</code>	<code>static</code>	<code>switch</code>	<code>throw</code>	<code>trait</code>
<code>try</code>	<code>unset()</code>	<code>use</code>	<code>var</code>	<code>while</code>
<code>Xor</code>	<code>yield</code>			

Kennistoets 1.3: Taalcomponenten

Bestudeer de theorie van dit lesblok en maak de volgende kennistoets.

Toets 1.3	Geef aan of de volgende variabelenamen juist zijn.
	1. \$auto2
	2. \$auto's
	3. \$auto/onderdelen
	4. \$thuis&tuin
	5. \$naw
	6. \$n-a-w
	7. \$n_a_w
	8. \$22
	9. \$\$auto
	10. \$autoKleur

Vaardigheid-lab 1.03: Taalcomponenten

Stap 1: Codeer een PHP-script waarin je de volgende variabelen met de afmetingen van een container opslaat:

```
breedte = 10  
lengte = 11  
hoogte = 5
```

Stap 2: Reken in je script de volgende variabelen uit:

- oppervlakte van de basis
- volume

Stap 3: Sla het script op als **lab1.03.php** in je map *lessen*.

Het resultaat moet er als volgt uitzien:

- Containerbasisoppervlakte is: 110
- Containervolume is: 550

1.4 Datatypen

Lesblok 1.4		
	Tijdsduur	2 uur
	Doel	Inleiding datatypen
	Benodigheden	Code-editor en XAMPP

PHP werkt, zoals iedere programmeertaal, met de volgende datatypen:

- **String** voor teksten of tekenreeksen;
- **Integer** voor gehele getallen, bijvoorbeeld 3 of 500;
- **Double** voor reële getallen, bijvoorbeeld 67 of 123.465;
- **Boolean** voor de waarden true of false;
- **NULL** voor variabelen die (nog) geen waarde hebben.

Het is belangrijk om goed te begrijpen wanneer en waarom je een specifiek datatype moet gebruiken. Hier volgt een overzicht van datatypen.

Het datatype string

Strings zijn tekenreeksen. Een string geef je aan als een tekst tussen aanhalingstekens, bijvoorbeeld "tekst". De meeste informatie in databases of web-applicaties bestaat in de vorm van teksten, bijvoorbeeld NAW-gegevens, productinformatie enzovoort.

- **Opgave 1**

Typ het volgende over en sla het op als **datatypen.php**.

```
<?php
    $naam = "Carl";
    $straat = "Kruislaan 111";
    $woonplaats = "Utrecht";
?>
```

Samenvoegen van twee tekstelementen

Met de operator punt (.) kun je twee tekstelementen samenvoegen. Bijvoorbeeld: "boek" . "titel" geeft als resultaat de string "boektitel".

- **Opgave 2**

Open **datatypes.php** en voeg de volgende code eraan toe:

```
$naw = $naam . " " . $straat . " " . $woonplaats;
echo "Gegevens: $naw";
?>
```

Hier hebben we de variabele `$naw` binnen een string als volgt geplaatst:

```
echo "Gegevens: $naw";
```

De waarde van de variabele wordt als eerste geëvalueerd en daarna wordt de echo-opdracht uitgevoerd.

Het resultaat van deze opgave moet er als volgt uitzien:

Gegevens: Carl Kruislaan 111 Utrecht

De heredoc-syntaxis (<<<)

Met de heredoc-syntaxis kun je lange teksten maken. Je hoeft alleen een eindmarkering te specificeren en aan het einde van je tekst de eindmarkering te plaatsen. Dit kan worden gebruikt om complexe strings te tonen. Je kunt nu zonder problemen aanhalingstekens in je echo plaatsen zonder een error te krijgen.

- **Opgave 3**

Open **datatypes.php** en voeg de volgende code eraan toe:

```
echo <<<TEKST
<br>Salarisspecificatie van $naam: 2000 euro
<br>Maand: November
<br>Jaar: 2020
TEKST;
```

LET OP

De eind-tag moet dezelfde naam als de begin-tag hebben, in dit geval <<<TEKST. De eind-tag TEKST (zonder <<<) moet direct aan het begin van de regel (dus tegen de linkerkantlijn) beginnen anders krijg je een error.

Het resultaat van deze opgave zie je hieronder:

Salarisspecificatie van Carl: 2000 euro Maand: November Jaar: 2020
--

Het datatype integer

Een integer is een getal zonder decimale komma (of decimale punt), bijvoorbeeld het getal 6. Een getal van het datatype integer kan positief of negatief zijn. Bijvoorbeeld:

```
$tarief = 13;  
$rente = -0.01;
```

Het datatype double

Een getal van het datatype double is een getal met een decimale punt. Bijvoorbeeld:

```
$tarief = 13.637;
```

Getallen van het datatype double kunnen geformatteerd worden voor het printen, bijvoorbeeld met drie of twee decimale posities. Dit doen we met de methode `printf()`.

De methode `printf()`

De methode `printf()` formatteert de waarde van een variabele naar een gegeven printf-formaat. De oorspronkelijke waarde van de variabele blijft hetzelfde. Bijvoorbeeld:

```
printf("<br />tarief is: %.2f", $tarief);
```

De methode `printf("tarief is: %.2f", $tarief)` formatteert de double-waarde van de variabele `$tarief` in het formaat `%.2f`, oftewel afgerond naar twee decimale posities: 13.64.

`<br/>` is de html-tag voor een nieuwe regel.

De conversie van double naar integer met `intval()`

Met de methode `intval()` maken we conversies van double- naar integer-datatypes. Bijvoorbeeld:

```
$tarief = 13.637;  
$tarief = intval($tarief);
```

De `intval()`-methode levert het integer-deel van een double-getal op. In dit voorbeeld krijg je in `$tarief` het resultaat 13. Het getal wordt niet afgerond.

Afronden van double naar integer met `round()`

Met de methode `round()` kunnen we een double afronden naar een integer. Bijvoorbeeld:

```
$tarief = 13.637;  
$tarief = round($tarief);
```

In dit voorbeeld is het resultaat 14.

Met de `round()`-function kunnen we ook het aantal decimalen aangeven.

```
$tarief = round($tarief,2);
```

In dit voorbeeld is het resultaat van `$tarief`: 13.64.

De conversie naar double met `doubleval()`

Met de methode `doubleval()` maken we conversies naar double. Bijvoorbeeld:

```
$tarief = "13.99";  
$tarief = doubleval($tarief);
```

In dit voorbeeld is het resultaat in `$tarief`: 13.99.

De string "13.99" wordt geconverteerd naar de double-waarde 13.99.

Het datatype boolean

Het datatype boolean kan alleen twee waarden krijgen: `true` of `false` (waar of onwaar). Op basis van de waarde van boolean variabelen kunnen we straks interessante controles programmeren. Bijvoorbeeld:

```
$minderjarig = true;  
$sportief = false;
```

De waarde `true` wordt als het getal 1 weergegeven. De waarde `false` wordt als blanco weergegeven.

`gettype()`

Deze methode geeft als resultaat het datatype van een variabele. Bijvoorbeeld:

```
$tarief = 13.0;  
$type = gettype($tarief) // Resultaat is "double"  
$tarief = "tien";  
$type = gettype($tarief) // Resultaat is "string"
```

settype()

Met deze methode kunnen we het datatype van een variabele omzetten naar een ander datatype. Bijvoorbeeld:

```
$tarief = 13.0;  
settype($tarief, "string");  
$type = gettype($tarief) // Resultaat is "string"  
$tarief = "33.99tien";  
settype($tarief, "double");  
$type = gettype($tarief) // Resultaat is "double"  
echo $tarief // Resultaat is 33.99
```

Als we de string 33.99tien naar een double converteren krijgen we de double-waarde 33.99.

• Opgave 4

Open **datatypen.php** en voeg de volgende code eraan toe:

```
$dollars = 999.99;  
$koers = 1.2;  
$euros = $dollars * $koers;
```

Toon het bedrag in euro's afgerond op twee decimalen. Het resultaat moet er als volgt uitzien:

Bedrag in euro's is: 1199.99

Kennistoets 1.4: Datatypes

Bestudeer de theorie van dit lesblok en maak de volgende kennistoets.

Toets 1.4	Als \$tarief = 21.21 wat is het resultaat van de codes:
	1. \$type = gettype(\$tarief);
	2. \$euros = \$tarief * \$11;
	3. \$tarief = round(\$tarief,2);
	4. \$tarief = round(\$tarief);
	5. printf(" tarief is: %.2f", \$tarief);
	6. echo "Tarief: \$tarief";
	7. settype(\$tarief, "string");
	8. \$type = gettype(\$tarief);
	9. echo \$tarief;
	10. echo \$tarief;

Vaardigheid-lab 1.04: Datatypes

Typ de volgende code over en sla dit op als **lab1.04.php** in je map *lessen*.

```
<h3>php lab 1.4</h3>
<?php
    $naam = "Karim";
    $nederlands = "8.5";
    $engels = "7.7";
    $rekenen = "6.7";
    $programmeren = "8.5";
    $databases = "9.4";
    echo "<table border='1'>
    <caption>
        <strong>Rapport</strong>
    </caption>
    <thead>
        <tr>
            <th>Naam</th><th>Nederlands</th><th>Engels</th>
            <th>Rekenen</th><th>Programmeren</th>
            <th>Databases</th>
            <th>Gemiddeld</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>$naam</td><td>$nederlands</td><td>$engels</td>
            <td>$rekenen</td><td>$programmeren</td>
            <td>$databases</td><td>-----</td>
        </tr>
```

```

</tbody>
<tfoot>
  <tr>
    <td colspan='6'>Groep gemiddeld</td><td>-----</td>
  </tr>
</tfoot>
</table>;
?>

```

Het resultaat moet er als volgt uitzien:

Rapport						
Naam	Nederlands	Engels	Rekenen	Programmeren	Databases	Gemiddeld
Karim	8.5	7.7	6.7	8.5	9.4	-----
Groep gemiddeld						-----

Figuur 1.6 Resultaat van vorige script

Maak nieuwe variabelen voor de tweede student en maak de nodige conversies aan om de gemiddelde cijfers met twee decimalen en het groepsgemiddelde met één decimaal te kunnen uitrekenen. Zie de volgende figuur:

Rapport						
Naam	Nederlands	Engels	Rekenen	Programmeren	Databases	Gemiddeld
Karim	8.5	7.7	6.7	8.5	9.4	8.08
Sophie	8.9	8.7	9.7	9.5	9.2	9.12
Groep gemiddeld						8.6

Figuur 1.7 Resultaat van Labo4

1.5 Array-methodes

Lesblok 1.5		
	Tijdsduur	2 uur
	Doel	Basis array-methodes
	Benodigdheden	Code-editor en XAMPP

Een array is een lijst met items, een beetje zoals een boodschappenlijstje. Wanneer je boodschappen moet doen zou je voor elk artikel dat je moet kopen één geeltje kunnen gebruiken. Een geeltje functioneert dan als een variabele. Maar wanneer je alle boodschappen op één lijstje zet dan is dit lijstje de array.

Geïndexeerde arrays in PHP zijn zoals de geïndexeerde arrays in andere programmeertalen. We kunnen deze geïndexeerde verzamelingen van gegevens op de volgende drie manieren declareren:

```
$arrayNaam = array(1,2,3,4,5);
```

of

```
$arrayNaam = [1,2,3,4,5];
```

of

```
$arrayNaam[0] = 1;  
$arrayNaam[1] = 2;  
$arrayNaam[2] = 3;  
$arrayNaam[3] = 4;  
$arrayNaam[4] = 5;
```

Bij de laatste vorm krijgen de elementen een index tussen haakjes [] die de volgorde in de array bepaalt.

In de volgende opgaven behandelen we de volgende PHP-array-methodes:

- `print_r`
- `var_dump`
- `unset`
- `array_key_exists`
- `in_array`
- `array_search`
- `array_push`
- `array_pop`
- `array_shift`
- `array_unshift`
- `array_rand`
- `array_walk`
- `implode`
- `explode`
- `array_walk_recursive`
- `array_merge`
- `array_slice`

- **Opgave 5**

Open een nieuw bestand en sla het op als **arrays.php** in de map *lessen*. Voeg de volgende code eraan toe:

```
<?php
    echo "<br>---- Opgave 5";
    $producten = [];
    $producten[0] = "Boeken";
    $producten[1] = "CD's";
    $producten[2] = "Smartphones";
    $producten[3] = "DVD's";
?>
```

De methode `print_r()`

PHP beschikt over de methode `print_r()` om snel de inhoud van een array te printen. De syntax is als volgt:

```
print_r($arrayNaam);
```

- **Opgave 6**

Open **arrays.php**. Voeg de volgende methode `print_r()` eraan toe:

```
echo "<br>---- Opgave 6: print_r()";
print_r($producten);
```

In de volgende opgaven sla je **arrays.php** op en start je die in je browser. We laten alleen de resultaten van de laatste opgave zien. Het resultaat zie je hieronder:

```
---- Opgave 6: print_r() Array ( [0] => Boeken [1] => CD's [2] => Smartphones [3] => DVD's )
```

De methode `var_dump()`

Deze methode gebruiken we om snel de inhoud van een variabele of array te printen. De syntax is als volgt:

```
var_dump($producten)
```

- **Opgave 7**

Open **arrays.php**. Voeg de volgende methode `var_dump()` toe:

```
echo "<br>---- Opgave 7: var_dump(): ";
var_dump($producten);
```

Het resultaat zie je hieronder:

```
---- Opgave 7: var_dump(): array(4) { [0]=> string(6) "Boeken" [1]=> string(4) "CD's" [2]=> string(11) "Smartphones" [3]=> string(5) "DVD's" }
```