

SPONTAAN

<?PHP

DOOR MICHIEL PATER



Spontaan PHP

Copyright © 2024: Tradosoft
Auteur: Michiel Pater
Versie: Tweede druk (v2.1)

Vormgeving omslag: Neline Akkermans | Zij ontwikkelt

ISBN: 97-890-90379-21-0
NUR: 989

Website: <https://spontaanphp.nl>

Alle rechten voorbehouden. Uit deze uitgave mag niets 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 Tradosoft.

OVERZICHT

| | |
|---|-----------|
| Overzicht | 2 |
| Welkom op reis | 8 |
| Een kijkje achter de schermen | 9 |
| Overzicht | 10 |
| Verzoeken en reacties | 11 |
| Hypertext | 12 |
| HTTP | 12 |
| Server side en client side | 13 |
| De rol van PHP | 14 |
| De webserver (<i>PHP Parser</i>) | 14 |
| Samenvatting | 15 |
| Lokale webserver installatie | 16 |
| Instellingen | 17 |
| Ontwikkelen in VSCode | 18 |
| Folders en Tabs | 18 |
| Tips voor editing | 19 |
| Instellingen | 20 |
| 1. Introductie in PHP | 21 |
| PHP Bestanden en editor | 21 |
| Speciale woorden (<i>keywords</i>) | 21 |
| Eerste bestand | 22 |
| 2. Variabelen en waarden (<i>variables & values</i>) | 23 |
| Concept | 23 |
| Regels voor namen van variabelen | 24 |

| | |
|--|-----------|
| Code | 24 |
| Typen (<i>data types</i>) | 25 |
| Constants | 26 |
| Letterlijke tekst (<i>literal</i>) | 27 |
| Speciale tekens (<i>special characters</i>) | 28 |
| Nummer notaties (<i>number notations</i>) | 29 |
| Speciale notaties (talstelsels) | 29 |
| Uitleg van talstelsels | 30 |
| Kleurcodes (<i>hexadecimaal</i>) | 31 |
| Waar en onwaar (<i>true and false</i>) | 32 |
| Typen weergeven en veranderen | 33 |
| Lege variabelen (<i>empty variables</i>) | 35 |
| 3. Leestekens (<i>punctuation</i>) | 36 |
| Puntcomma ; (<i>semicolon</i>) | 36 |
| Whitespace (<i>spaties, tabs en nieuwe regels</i>) | 37 |
| Tabs | 38 |
| Haakjes ((<i>parentheses</i>) | 39 |
| Vierkante haakjes [(<i>square brackets</i>) | 39 |
| Accolades { (<i>curly braces</i>) | 40 |
| 4. Commentaar (<i>comments</i>) | 41 |
| Enkele regels // (<i>single-line comments</i>) | 42 |
| Meerdere regels /* (<i>multi-line comments</i>) | 43 |
| 5. Bewerkingen en invoerwaarden (<i>operators & operands</i>) | 44 |
| Expressies (<i>expressions</i>) | 44 |
| Toewijzing = (<i>assignment operator</i>) | 44 |
| Tekst bewerkingen (<i>string operators</i>) | 45 |

| | |
|--|-----------|
| Samenvoegen . (<i>concatenation operator</i>) | 45 |
| Bijvoegen .= (<i>concatenation assignment operator</i>) | 46 |
| Rekenkundige bewerkingen (<i>arithmetic operators</i>) | 47 |
| Toevoegen + (<i>addition operator</i>) | 47 |
| Verminderen - (<i>subtraction operator</i>) | 47 |
| Omkeren - (<i>negation operator</i>) | 48 |
| Ophogen ++ en verlagen -- (<i>increment and decrement</i>) | 48 |
| Vermenigvuldigen * en delen / (<i>multiplication and division</i>) | 48 |
| Modulus % (<i>modulo</i>) | 49 |
| Rekenkundig toewijzen (<i>arithmetic assignment operators</i>) | 50 |
| Logische bewerkingen (<i>logical operators</i>) | 51 |
| En && (<i>AND operator</i>) | 51 |
| Of (<i>OR operator</i>) | 52 |
| Niet ! (<i>NOT operator</i>) | 52 |
| Combinaties | 53 |
| Vergelijkingen (<i>comparison operators</i>) | 54 |
| 6. Bestanden opsplitsen | 56 |
| Herhalen en recursie (<i>recursion</i>) | 57 |
| Inladen | 57 |
| Controle | 58 |
| 7. Fouten opsporen (<i>tracing errors</i>) | 59 |
| Fouten weergeven (<i>enable error reporting</i>) | 59 |
| Foutmeldingen | 60 |
| Notice | 61 |
| Warning | 62 |
| Parse error | 63 |

| | |
|---|-----------|
| Fatal error | 64 |
| Debugging | 65 |
| Informatie over variabelen weergeven | 65 |
| Informatie over arrays weergeven..... | 66 |
| Sporen achterlaten..... | 68 |
| Het script tussendoor stoppen..... | 69 |
| 8. Voorwaardelijke code (<i>if statements</i>) | 70 |
| Verkorte if statement (<i>ternary operator</i>)..... | 76 |
| Switch statement..... | 77 |
| 9. Code herhalen (<i>loops</i>)..... | 79 |
| While loop..... | 79 |
| For loop..... | 82 |
| Foreach loop..... | 82 |
| 10. Functies (<i>functions</i>) | 84 |
| Ingebouwde functies (<i>built-in functions</i>)..... | 84 |
| Op maat gemaakt (<i>user-defined functions</i>)..... | 85 |
| Zelf-herhalende functies (<i>recursion</i>) | 87 |
| 11. Lijsten (<i>arrays</i>)..... | 88 |
| Introductie | 88 |
| Soorten | 91 |
| Loops | 93 |
| Functies | 94 |
| Shift en unshift | 94 |
| Push en pop | 94 |
| Samenvoegen | 95 |
| Sorteren | 96 |

| | |
|---|------------|
| 12. Objecten (<i>classes & objects</i>) | 97 |
| Voorbeeld Liniair programmeren..... | 97 |
| Voorbeeld Object-georiënteerd programmeren (<i>OOP</i>) | 98 |
| Sjabloon (<i>class definition</i>) | 99 |
| Object (<i>instance</i>)..... | 100 |
| Constructors en destructors..... | 100 |
| Object-georiënteerd (<i>OOP - object oriented programming</i>)..... | 102 |
| Herbruiken van code (<i>reuse</i>)..... | 102 |
| Inkaderen van code (<i>encapsulation</i>) | 102 |
| Overerven (<i>inheritance</i>)..... | 103 |
| Meerdere vormen (<i>polymorphism</i>) | 104 |
| Abstracte sjablonen (<i>abstract classes</i>)..... | 104 |
| 13. Formulieren (<i>forms</i>) | 106 |
| Overzicht..... | 107 |
| Verzoeken (<i>GET en POST requests</i>) | 108 |
| GET | 108 |
| POST | 108 |
| Voorbeelden | 109 |
| Sessies en doorsturen (<i>sessions and redirect</i>)..... | 112 |
| Andere elementen..... | 114 |
| 14. Bestanden (<i>files</i>) | 115 |
| Hele bestanden inladen en opslaan..... | 115 |
| Regel voor regel inlezen (<i>reading</i>)..... | 117 |
| Schrijven naar bestanden (<i>writing or appending</i>) | 119 |

| | |
|---|------------|
| 15. Databases (MySQL) | 120 |
| Werkwijze..... | 120 |
| Server instellen | 121 |
| Database Interface (<i>PhpMyAdmin</i>)..... | 122 |
| Verbinden met PHP | 130 |
| SQL Queries | 131 |
| SELECT | 131 |
| INSERT | 132 |
| UPDATE..... | 134 |
| DELETE | 135 |
| Linken leggen tussen tabellen | 135 |
| Gefeliciteerd | 135 |
| Veel succes! | 136 |
| Referenties | 137 |
| Spontaan PHP | 137 |
| PHP Handleiding..... | 137 |
| Google | 137 |
| ChatGPT | 138 |
| Stackoverflow..... | 138 |
| Algemene links..... | 139 |
| Links naar de PHP Handleiding..... | 139 |

WELKOM OP REIS

Leuk dat je hebt gekozen om dit boek te lezen!

Hierin leg ik in eenvoudige stappen de basisprincipes van **PHP** uit, om je op weg te helpen. Dit geeft je handvaten om zelf aan de slag te kunnen gaan en verder te ontdekken wat er allemaal mogelijk is.

Als je een bestaand systeem zoals **Wordpress** of een **webshop** zou vergelijken met een gerecht, dan leg ik een goede basis aan ingrediënten uit om het precies zo te maken als je wilt.

In dit boek ga ik ervan uit dat je basiskennis hebt van het werken met computers en al wat ervaring hebt met **HTML** en **CSS**.

We houden **Engels** aan als de basistaal, maar ik leg het uit in het Nederlands. Het is handig om de Engelse termen te leren kennen, zodat je ze herkent als je ze tegenkomt. Bijvoorbeeld in foutmeldingen.

Bij het uitleggen maak ik gebruik van **Windows** als besturingssysteem, **XAMPP** als lokale webserver, **VSCode** als editor en **Google Chrome** als webbrowswer. Voor **Mac** blijft alles hetzelfde, alleen de installatie van de **webserver** is dan iets anders dan mijn uitleg.

Nadat ik wat meer heb uitgelegd over de achtergrond van **PHP**, laat ik zien hoe je **XAMPP** zelf kunt installeren om snel code te kunnen testen.

Tijdens het ontwikkelen raad ik je aan om gebruik te maken van de officiële PHP handleiding (**manual**) op <https://www.php.net/>. Het is een goede gewoonte om daarin te kijken.

Onder **Referenties** staan **links** naar meer informatie over de onderwerpen die we behandelen en naar artikelen van **Spontaan PHP**.

Als laatste wil ik je aanmoedigen om vooral uit te proberen wat je leest. Je kunt het natypen, wat experimenteren en kijken wat je al kunt.

Veel plezier bij het lezen van dit boek!

Hypertext

Het centrale woord in de afbeelding is het woord **hypertext**. Dit is de kale (**raw**) tekst die verstuurd wordt via het **HTTP** protocol. Daarom heet het protocol **Hypertext Transfer Protocol**. *(Tegenwoordig wordt vaak **HTTPS** gebruikt, waarbij de **S** voor **Secure** staat en voorzien is van een veiligheidscertificaat. Voor nu behandelen we **HTTP**).*

Als we alleen **hypertext** ontvangen zien we nog geen website. De **hypertext** kan namelijk worden vormgegeven door gebruik te maken van **HTML**, dat staat voor **Hypertext Markup Language**. Dit wordt vaak geschreven in **HTML** bestanden of **PHP** bestanden.

Elke **domeinnaam** heeft een eigen **map** op de **server** met **bestanden**. Die **bestanden** worden aangeleverd aan de **webserver**, die het weer doorstuurt naar de **webbrowser** op de **client**.

Als de **webserver** een **HTTP** verzoek (**request**) ontvangt van een **client** om een bepaald bestand (**file**) binnen een bepaalde **domeinnaam** te openen, dan kan de **hypertext** (eventueel met **HTML**) terug worden gestuurd (**response**) via het **HTTP** protocol. Hiervoor is het nodig dat de opgevraagde **domeinnaam** gekoppeld is aan de server die het verzoek ontvangt op dat **IP adres**.

HTTP

De term **GET** die in het voorbeeld naast **HTTP** staat is een **commando** van het **HTTP** protocol. Deze wordt normaal gesproken gebruikt om **verzoeken** te sturen. Er is ook nog een **POST commando**, waarmee **data** (zoals uit formulieren) verzonden kan worden naar de **server**.

Het protocol **HTTP** geeft ook **status codes** terug. Op de volgende pagina staan een paar veel voorkomende.

1. INTRODUCTIE IN PHP

PHP Bestanden en editor

In dit boek maken we gebruik van je **lokale webserver** (zoals uitgelegd in *Lokale webserver installatie*). We starten onze reis met het maken van ons eerste **PHP bestand!**

Deze **PHP bestanden** kun je herkennen aan dat hun namen altijd eindigen met **.php**. Om ervoor te zorgen dat het bestand meteen wordt ingeladen als we onze website openen, kunnen we het **index.php** noemen.

In dit boek wordt code weergegeven in **VSCode**. Voel je vrij om een **editor** te kiezen waar je graag in werkt.

Speciale woorden (*keywords*)

In **PHP** zijn veel speciale woorden (**keywords**).

Deze zijn gereserveerd om **instructies** te geven aan **PHP**.

Voorbeelden zijn **echo**, **if**, **else** en **function**.

Het **echo** keyword stuurt **output** (tekst) naar het scherm. Een alternatief hiervan is **print**.

De **if** en **else** keywords worden gebruikt in **if statements**.

Het **function** keyword verteld aan **PHP** dat een nieuwe functie wordt gedefinieerd.

Over **if statements** en **functies** wordt later in het boek meer uitgelegd.

Letterlijke tekst (*literal*)

Het `string` data type heeft letterlijke tekst (`literal`) als waarde.

Teksten kunnen weergegeven worden met enkele aanhalingstekens (`'`), dat zijn `single quotes`. Of met dubbele aanhalingstekens (`"`), dat zijn `double quotes`.

| Letterlijk (<i>literal</i>) | Voorbeeld |
|-------------------------------|----------------|
| Single quotes | 'Hello world!' |
| Double quotes | "Hello world!" |

Binnen `single quotes` kun je wel `double quotes` gebruiken en binnen `double quotes` kun je wel `single quotes` gebruiken.

Maar je kunt niet zomaar `single quotes` binnen `single quotes` of `double quotes` binnen `double quotes` gebruiken.

Deze zou je dan moeten uitsluiten (`escape`) met een backslash (`\`).

| Letterlijk (<i>literal</i>) | Voorbeeld |
|---------------------------------|--------------------|
| Single quotes met double quotes | 'Hello "world"!' |
| Double quotes met single quotes | "Hello 'world'!" |
| Single quotes met single quotes | 'Hello \'world\''! |
| Double quotes met double quotes | "Hello \"world\"!" |

Typen weergeven en veranderen

Om het **type** van een **variabele** of **waarde** weer te geven kunnen we gebruik maken van de functie `var_dump()`. Je ziet dan het **type**, de **waarde** en eventueel de **lengte**.

| Resultaat |
|-------------------------|
| int(1) string(4) "test" |

We kunnen ook **controleren** of een **variabele** van een bepaald **type** is door gebruik te maken van specifieke **functies** die het type **controleren**. Dit zijn een paar voorbeelden.

| Type | Code |
|---------|--------------------------|
| Boolean | <code>is_bool()</code> |
| Integer | <code>is_int()</code> |
| Float | <code>is_float()</code> |
| String | <code>is_string()</code> |
| Array | <code>is_array()</code> |

```
1 <?php
2     $a = 1;
3
4     var_dump( is_int( $a ) );
5     var_dump( is_bool( 'test' ) );
6 ?>
```

| Resultaat |
|------------------------|
| bool(true) bool(false) |

Daarnaast kunnen we het **type** van een **variabele** veranderen (**type casting**) door een ander **type** tussen **haakjes** ervoor te plaatsen.

3. LEESTEKENS (*punctuation*)

In **PHP** worden veel **leestekens** gebruikt om de **code** te **structureren**. Dit bepaald een groot deel van de opmaak van de taal (**syntax**), hoe we het **schrijven**. Als we daarin een **foutje** maken dan geeft **PHP** al snel een **Parse error**. Daarom is het belangrijk om goed te begrijpen hoe de **opmaak** van de **taal** is, om **typfouten** te voorkomen.

In dit hoofdstuk staat de basis uitgelegd om het iets beter te begrijpen.

Puntcomma ; (*semicolon*)

Een puntcomma geeft aan **PHP** aan dat het het einde van een **regel code** is.

Daarom hoeft er niet per sé een nieuwe **regel** na een **puntcomma** geschreven te worden.

Niet alle regels **code** eindigen met een **puntcomma**, zoals bijvoorbeeld **if statements** of **functies** (*hierover wordt later in dit boek meer uitgelegd*).

In dit voorbeeld wordt de **code** in het **if statement** altijd uitgevoerd (**true**).

```
1  <?php
2      echo 'Hello';
3
4      if( true )
5      {
6          echo ' world!';
7      }
8  ?>
```

5. BEWERKINGEN EN INVOERWAARDEN

(operators & operands)

Expressies (expressions)

Eigenlijk iedere **regel** code die iets berekent noemen we een uitdrukking (**expression**). Dit is de basis van waaruit we aan **PHP** vertellen wat we willen verklaren (**declare**).

PHP weet uit zichzelf niets van het **stelsel** dat wij willen **ontwikkelen** in onze **code**. We moeten alles aan **PHP** uitleggen doormiddel van **expressies**.

Een voorbeeld is het **toewijzen** van een **waarde** aan een **variabele**:

```
1 <?php
2     $variable_a = 3;
3 ?>
```

Ook andere **berekeningen** die we doen zijn **expressies**. In dit hoofdstuk wordt meer uitgelegd over bewerkingen (**operators**) en invoerwaarden (**operands**) die daar een groot deel van uit maken.

Toewijzing = (assignment operator)

Als eerste wordt in het voorbeeld hierboven in de **expressie** een waarde **toegewezen** aan een **variabele**. Dat wordt gedaan doormiddel van een **assignment operator**, het = teken.

Dit is de basis voor alle andere **bewerkingen**.

7. FOUTEN OPSPOREN (*tracing errors*)

Eén van de meest belangrijke vaardigheden bij programmeren is fouten (**errors**) leren opsporen en oplossen.

Daarom is het handig om te begrijpen welke soorten foutmeldingen (**error messages**) er zijn.

Fouten weergeven (*enable error reporting*)

Het kan voorkomen dat je een **Internal Server Error** of **HTTP 500 error** terug krijgt, zoals op de afbeelding hieronder. Dan worden **PHP fouten** niet weergegeven.

Internal Server Error

The server encountered an internal error or misconfiguration and was unable to complete your request.

Please contact the server administrator at postmaster@localhost to inform them of the time this error occurred, and the actions you performed just before this error.

More information about this error may be available in the server error log.

Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12 Server at localhost Port 80

Gebruik dan deze **code** om het **weergeven** van **PHP** foutmeldingen aan te zetten.

```
1  <?php
2      ini_set( 'display_startup_errors', 1 );
3      ini_set( 'display_errors', 1 );
4      error_reporting( E_ALL );
5  ?>
```

Met de functie **ini_set()** worden **PHP instellingen** tijdelijk aangepast die in het configuratiebestand **php.ini** op de **server** staan **opgeslagen**. In dit geval gaat het over of **fouten** (bij het opstarten) worden weergegeven

12. OBJECTEN (*classes & objects*)

Een **object** in **PHP** is een uitgebreid **type waarde** waarin **variabelen** en **functies** kunnen worden opgeslagen. Elk **object** is gemaakt van een sjabloon (**class**) die eerst **gedefinieerd** wordt voordat deze gebruikt kan worden.

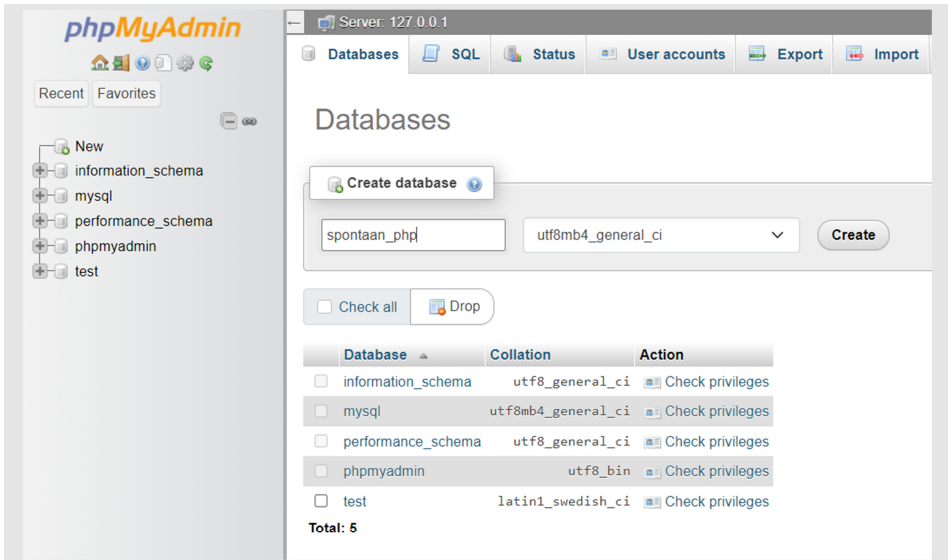
Een manier van programmeren die vaak gebruikt wordt is object-georiënteerd programmeren (**Object Oriented Programming** of **OOP**). Dit is een werkwijze, waarbij de focus op **objecten** ligt. Het idee is dat je meer denkt vanuit **objecten** die **eigenschappen** hebben en **acties** uitvoeren, in plaats van **lineair programmeren**, waarbij vooral **stapsgewijs** wordt gedacht.

Hieronder volgt een voorbeeld van **lineair programmeren** en daarna hetzelfde voorbeeld volgens de wijze van **object-georiënteerd programmeren**.

Voorbeeld Lineair programmeren

```
1  <?php
2      $dictionary = [
3          'zero' => 'Nul',
4          'one' => 'Eén'
5      ];
6
7      function translate( $word, $dictionary )
8      {
9          if( !array_key_exists( $word, $dictionary ) )
10             exit( 'Vertaling niet gevonden voor: ' . $word );
11
12             return $dictionary[ $word ];
13         }
14
15         echo translate( 'zero', $dictionary ) . ': First item<br />';
16         echo translate( 'one', $dictionary ) . ': Second item<br />';
17     ?>
```

Klik op **Databases**, vul de **naam** in voor de nieuwe **database** en klik op **Create**.

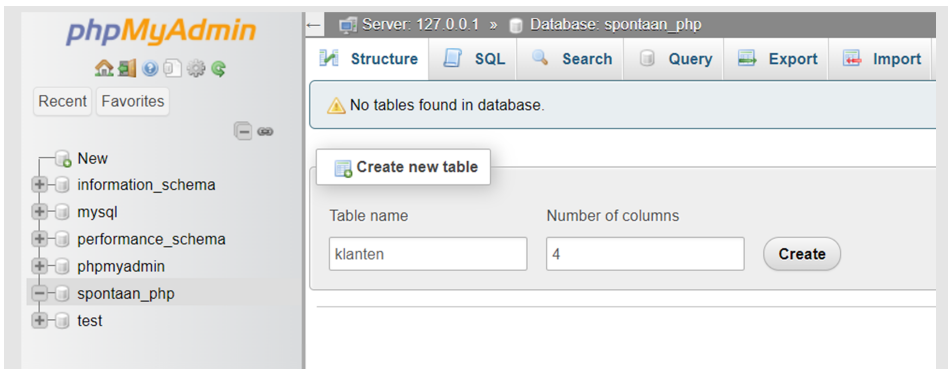


The screenshot shows the phpMyAdmin interface. On the left, a sidebar lists databases: information_schema, mysql, performance_schema, phpmyadmin, and test. The main area is titled 'Databases' and features a 'Create database' button. Below it, a form contains the text 'spontaan_php' in a text box and 'utf8mb4_general_ci' in a dropdown menu, with a 'Create' button to the right. Below the form are 'Check all' and 'Drop' buttons. A table lists existing databases with columns for Database, Collation, and Action. The table contains five rows: information_schema (utf8_general_ci), mysql (utf8mb4_general_ci), performance_schema (utf8_general_ci), phpmyadmin (utf8_bin), and test (latin1_swedish_ci). A 'Total: 5' label is at the bottom of the table.

| Database | Collation | Action |
|---|--------------------|----------------------------------|
| <input type="checkbox"/> information_schema | utf8_general_ci | Check privileges |
| <input type="checkbox"/> mysql | utf8mb4_general_ci | Check privileges |
| <input type="checkbox"/> performance_schema | utf8_general_ci | Check privileges |
| <input type="checkbox"/> phpmyadmin | utf8_bin | Check privileges |
| <input type="checkbox"/> test | latin1_swedish_ci | Check privileges |

Total: 5

De **database** is aangemaakt en is nog **leeg**.



The screenshot shows the phpMyAdmin interface for a newly created database named 'spontaan_php'. The main area is titled 'Structure' and displays a warning: 'No tables found in database.' Below the warning is a 'Create new table' button. A form for creating a new table is visible, with 'Table name' set to 'klanten' and 'Number of columns' set to '4'. A 'Create' button is to the right of the form.

Algemene links

Complete lijst met **HTTP** status codes:

https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

XAMPP website voor downloaden van installatie:

<https://www.apachefriends.org/>

VSCode website voor downloaden van installatie:

<https://code.visualstudio.com/>

StackOver-flow:

<https://stackoverflow.com/>

Links naar de PHP Handleiding

Dit is een lijst met links naar pagina's in de officiële **PHP handleiding** met meer informatie over onderwerpen die we behandeld hebben in dit boek.

Complete lijst met **PHP Keywords**:

<https://www.php.net/manual/en/reserved.keywords.php>

Meer informatie over **variabelen**:

<https://www.php.net/manual/en/language.variables.basics.php>

Meer informatie over **datatypen**:

<https://www.php.net/manual/en/language.types.php>

Meer informatie over **strings**:

<https://www.php.net/manual/en/language.types.string.php>

Meer informatie over **integers** en andere nummer notaties:

<https://www.php.net/manual/en/language.types.integer.php>

Meer informatie over **booleans**:

<https://www.php.net/manual/en/language.types.string.php>

Wil je graag op een laagdrempelige manier de basis van **PHP** leren? Of heb je al ervaring met het maken van websites en wil je graag meer leren over **PHP** en hoe het jou kan helpen?

In eenvoudige stappen leg ik in dit boek een aantal basisprincipes van **PHP** uit, om je op weg te helpen. Dit geeft je handvaten om zelf aan de slag te kunnen gaan en verder te ontdekken wat er allemaal mogelijk is.

Hiermee leer je ingrediënten van **PHP** beter te begrijpen en te gebruiken om je websites in meer detail te kunnen ontwikkelen. Doordat je de achterliggende principes beter leert kennen, krijg je meer inzicht in wat er precies achter de schermen gebeurt en kom je minder snel voor verrassingen te staan. Je kunt dan stap voor stap volgen wat er gebeurt in de code en precies zien welke dingen je kunt veranderen om tot het resultaat te komen dat je graag wilt!

In dit boek ga ik ervan uit dat je basiskennis hebt van het werken met computers en al wat ervaring hebt met HTML en CSS.

Veel plezier met ontdekken!

Michiel



 **TRADOSOFT**

ISBN 978-90-903792-1-0



9 789090 379210 >