

# Inhoud

<b>1</b>	<b>PHP als scriptingtaal</b>	<b>1</b>
	Wat is PHP?	1
	Dynamische sites	2
	PHP als scriptingtaal	2
	PHP en MySQL: een koppel	4
	Waarom PHP?	5
	Libraries	6
	Frameworks	7
	Platformonafhankelijkheid	7
	De toekomst	7
	Intranet	8
	Webapplicaties	9
	Werking van PHP	9
	Databasegestuurde websites	11
	De webshop	13
	<a href="http://www.leer-php.nl">www.leer-php.nl</a>	14
	Volledige installatie	15
	Installatie van PHP vanaf het web	15
	De server starten	17
<b>2</b>	<b>Inleiding programmeren</b>	<b>19</b>
	Starten met programmeren	19
	Wat wil de gebruiker?	19
	De gebruiker als uitgangspunt	19
	De opdrachtgever als uitgangspunt	20
	Het projectplan	20
	Relatie html en PHP	21
	PHP en de server	22
	Html toevoegen	23
	Variabelen	25
	Variabelen in de praktijk	25
	Variabelen combineren	26
	Typen variabelen	26
	Met getallen aan de slag	28

<b>Condities</b>	<b>28</b>
<b>Lussen</b>	<b>32</b>
While	33
Do...while	34
<b>Stringbewerkingen</b>	<b>34</b>
Stringlengte	34
String opdelen	35
<b>Funcities</b>	<b>36</b>
Ingebouwde funcities	37
Zelf funcities maken	38
Functieopbouw	39
<b>Arrays</b>	<b>42</b>
Meerdimensionale arrays	44
<b>De webshop</b>	<b>45</b>
Basiseisen	45
Homepage	46
Een willekeurige pagina bepalen	46
De homepage bepalen	47
Tot slot	47

### **3 Webformulieren** **49**

<b>Inleiding formulieren</b>	<b>49</b>
Html-formulier	50
De methoden Get en Post	50
Afhandelen van een formulier	51
Het standaardmailformulier	52
Het formulier controleren	56
<b>Reguliere expressies</b>	<b>60</b>
Een patroon: de basis van reguliere expressies	60
Reguliere expressies toepassen	63
<b>Het bestelformulier</b>	<b>67</b>
<b>Sessies en cookies</b>	<b>75</b>
Verschillen tussen sessies en cookies	75
Cookies	76
Sessies	78
Authenticatie	79
Authenticatie van de webshop	85

<b>4</b>	<b>Bestanden en mappen</b>	<b>91</b>
	Inleiding tot bestanden	91
	De bestandsstructuur	91
	Toegankelijkheid van de server	93
	Een bestand schrijven	95
	Persoonlijke mails versturen	101
	Nieuws inlezen van een andere website	105
	Overige bestandsfuncties	109
	Map maken	109
	Map verwijderen	109
	Bestand verwijderen	110
	Bestand hernoemen	110
<b>5</b>	<b>Databases</b>	<b>111</b>
	Inleiding tot databases	111
	Waarom een database?	112
	De databasegeneratie	113
	Het databasemanagementsysteem	113
	Een eenvoudige webshop	114
	Het plan	115
	Entity Relationship Diagram	117
	Het ERD beschreven	118
	Een ERD van de webshop	119
	Database in tabelvorm	121
	Datatypes in MySQL	122
	Sleutels	123
	Een voorbeeldbestelling	126
	Fysieke implementatie van databasetabellen	127
	phpMyAdmin	127
	Tabellen implementeren in phpMyAdmin	129
	Data invoeren	131
	Data selecteren	132
	SQL schrijven	132
	Selectiequery's	132
	Meer tabellen selecteren	134
	Geavanceerde selectiequery's	135
	Data sorteren	136
	Data analyseren	136
	Data groeperen	138
	Rijen invoegen in de database	138
	Rijen aanpassen	139
	Rijen verwijderen	140

<b>Databases benaderen in PHP</b>	<b>141</b>
Contact maken met de database	141
Selectiequery's in PHP	142
Insertquery's in PHP	145
Updatequery's toepassen	151
Deletequery's toepassen	153
Meer items in een veel-op-veel-relatie	155
Afbeeldingen toevoegen aan een database	157
Afbeeldingen uploaden naar de server	158
Artikelen voorzien van een afbeelding	161
<b>6 Fouten identificeren</b>	<b>165</b>
<b>Fouten opsporen</b>	<b>165</b>
De puntkomma	165
Haakjes	166
Gebruik van punten en aanhalingstekens bij echo	166
Gebruik van variabelen	167
Fouten opsporen	167
Foutrapportage	169
Fouten signaleren in een logbestand	169
<b>Debuggen door middel van exception handling</b>	<b>170</b>
Een eenvoudig debugframework	172
<b>7 De webshop</b>	<b>175</b>
<b>De opmaak van de shop</b>	<b>175</b>
De homepage	178
<b>Zoeken in de webshop</b>	<b>178</b>
Het artikeloverzicht	179
<b>De winkelwagen</b>	<b>181</b>
Het authenticatiescript	187
Uitloggen	189
De winkelwagen in de praktijk	190
De bestelling uitvoeren	192
<b>De beheerfuncties</b>	<b>193</b>
Artikeloverzicht	194
Artikel toevoegen	196
Afbeelding uploaden	198
Artikel verwijderen	200
Categorie toevoegen en wijzigen	201
Een bestelling uitvoeren	203

<b>8 E-mail verzenden</b>	<b>207</b>
Technische kant van e-mail	207
Standaardmailfunctie	207
Phpmailer: de eerste mail versturen	208
Hoe werkt phpmailer?	209
Afbeeldingen	211
Externe afbeeldingen	211
Interne afbeeldingen	212
Attachments	213
Spam	215
Eenmalig en batchgewijs versturen	216
Eenmalig versturen	216
Batchgewijs versturen	217
Het script automatisch aanroepen	217
Nieuwsbrief versturen vanuit de webshop	218
Alle velden omzetten op basis van database velden	221
<b>9 Html-editor</b>	<b>223</b>
FCKeditor	223
Compatibiliteit: cross browser	224
Programma downloaden	224
Verschillende programmeertalen	224
Een html-invoerveld	225
Uploaden	227
Verschillende werkbalken	234
Zelf een werkbalk maken	235
De html-editor in MyWebshop	236
Een artikel toevoegen	236
De html-nieuwsbrief	239
<b>10 Pdf-bestanden maken</b>	<b>243</b>
FPDF	243
Programma downloaden	244
Uw eerste pdf-document	244
Afbeeldingen toevoegen	245
Headers en footers	246
Tabellen	248
Factuur genereren	252

<b>11 Geavanceerd MySQL</b>	<b>257</b>
Het resultaat van een query	257
Joins	260
Nieuwe tabel: Specials	261
Inner joins	262
USING	263
Left join	264
Right join	266
Joins in MyWebshop	267
Subquery's	269
Subselects met WHERE en =	270
Fulltext search in MySQL	271
Hoe werkt fullsearch?	272
Verschil met LIKE	274
Fullsearch met booleaanse expressies	274
De zoekfunctie in MyWebshop	277
Transactiemangement	279
Transaction safe versus Not transaction safe	280
De transactie in code	281
Voorbeeldtransactie	282
Transactiefunctielibrary	283
Transacties in MyWebshop	286
<b>12 Html-sjablonen in PHP</b>	<b>289</b>
Waarom templates?	289
Smarty installeren	289
Beginnen met Smarty	290
Alternatieve bestandslocaties	292
Meerdimensionale variabele doorgeven	292
Opmaak van een template	294
Gegevens verwerken in een template	296
Caching	299
Templates koppelen	300
Constanten in configuratiebestanden	302
Conditie	304
MyWebshop met templates	306
De zoekfunctie	309
Toon categorie met afbeeldingen	310

<b>■ 13 Geavanceerd gebruik van variabelen</b>	<b>317</b>
Primitieve variabelen	317
Complexe variabelen	318
De scope van variabelen	319
De stack en de heap	321
De stack	322
De heap	324
Referentievariabelen	327
De garbage collector	332
Verskil in variabelenbeheer PHP 4 en PHP 5	334
<b>■ 14 Objectgeoriënteerd programmeren</b>	<b>339</b>
Inleiding	339
Het idee achter objecten	342
Codeerstandaard	343
Bestandsnamen	344
Namen van functies en methoden	345
Namen van variabelen	346
Codestructuren	349
Commentaar	351
Codevoorbeeld	353
Codeerstandaarden	355
Klassen en objecten	356
Membervariabelen	357
Methoden	359
Constructor	360
Encapsulation	361
Toegang tot variabelen	363
Setters en Getters	365
Toegang tot methoden	369
Voorbeeld: webshop	372
Overerving	379
Polymorfisme	386
Beschikbaarheid van klassen	387
Include of require	387
__autoload	388

<b>15 Unified Modeling Language (UML)</b>	<b>389</b>
Wat is UML?	389
Use-casediagrammen en use cases	391
Use case Zoeken op artikel	394
Use case Bestellen	396
Use case Bestelling uitvoeren	397
Activiteitendiagram (activity diagram)	398
Klassediagram (class diagram)	403
Associaties	405
Programmeren van associaties	413
Van functioneel ontwerp naar technisch ontwerp	416
Componentendiagram (component diagram)	426
Interfaces	427
Componenten in de webshop	429
Communicatiediagram (communication diagram)	430
Coderen aan de hand van diagrammen	434
Toestanddiagram (state machine diagram)	440
Overige diagrammen	454
Sequentiediagram (sequence diagram)	454
Timingdiagram	455
Netwerkdigram (deployment diagram)	456
AGILE-softwareontwikkeling	457
Het beste product op het moment dat geld of tijd op is	458
SCRUM en eXtreme Programming (XP)	459
<b>16 OO-programmeren, geavanceerd</b>	<b>461</b>
Abstracte klassen en methoden	461
Abstracte klassen	465
Abstracte methoden	466
Interfaces	468
Meer interfaces implementeren	472
Final	473
Methoden met final	475
Klassevariabelen	476
Statische methoden	478
Wanneer gebruiken we statische methoden?	478
Klasseconstanten	479



Namespaces	482
Het keyword use	488
De magische variabelen <code>__NAMESPACE__</code> en <code>__METHOD__</code>	490
Namespaces gelden ook voor functies en constanten	491
Autoloading klassen met namespaces	492
<b>Ontwerpoverwegingen</b>	<b>494</b>
Sterke samenhang (strong cohesion)	494
Losse koppeling (loose coupling)	495
<b>■ 17 Ontwerppatronen (design patterns)</b>	<b>499</b>
Design patterns	499
Singleton	500
Voorbeeld: klasse Debug	500
Private constructor	504
Wanneer gebruiken we het singletonpatroon?	505
Waarschuwing voor gebruik van het singletonpatroon	505
Factory	505
Voorbeeld: SchermFactory	506
Wanneer gebruiken we het factorypatroon method?	514
Registry	515
Voorbeeld: gebruikersobject in registry	517
Wanneer gebruiken we het registrypatroon?	521
Waarschuwing voor gebruik van het registrypatroon	522
Model View Controller (MVC)	522
Voorbeeld: MVC-structuur in de webshop	524
Wanneer gebruiken we het MVC-patroon?	527
<b>■ 18 Errors en exceptions afhandelen</b>	<b>529</b>
Wat fouten zijn	529
Foutafhandeling	529
Uitzonderingen (exceptions)	533
Wat een exception is	534
Exception propagation	538
Automatisch exceptions afvangen	539
Eigen typen exceptions	540

<b>19 De bibliotheken van PHP</b>	<b>545</b>
PHP Data Objects (PDO)	545
Ondersteunde databases	548
Transacties	549
Prepared statements	554
Standard PHP Library (SPL)	561
Spl_autoload	561
Iterator	564
DirectoryIterator	574
<b>20 Extensible markup language (xml)</b>	<b>579</b>
Wat is xml?	579
Well-formed xml-documenten	581
Valide xml-documenten	583
Modelleren van een xml-document	584
Hiërarchie	584
Attribuut of subtag?	587
Speciale tekens	587
Encoding	589
Document type definition (dtd)	590
Cardinaliteit	592
Attributen	593
Volgorde van elementen en attributen	594
Een dtd koppelen aan een xml-document	595
Namespace	598
Xml-schema	601
Xml-schema is ook een xml-document	601
Gedefinieerde datatypen	606
Eigen datatypen	607
Een xml-schema koppelen aan een xml-document	611
XPath	612
Predicaten	614
Functies	615
Axes	616
Het document object model (DOM)	619
Genereren van xml	619
Importereren van xml	628
SimpleXML	632
Geen CDATA-elementen mogelijk	636
Omzetten van DOM naar SimpleXML	637
XPath in SimpleXML	637
Voor- en nadelen SimpleXML	638

<b>21 Webservices</b>	<b>639</b>
Wat een webservice is	639
Serialisatie	642
Xml-remote procedure calls (xml-rpc)	646
Simple object access protocol (soap)	648
Parameters voor SoapClient en SoapServer	651
Web service description language (wsdl)	656
Wsdl-cache	661
Uitwisselen van objecten	662
Afhandelen van soap-fouten met SoapFault	674
Voorbeeld in de webshop	675
<b>22 Inleiding tot het Zend Framework</b>	<b>681</b>
Wat is Zend Framework?	681
Welke versie van Zend Framework?	683
Waarom Zend Framework?	683
Andere frameworks	683
Andere PHP-libraries	684
Licentie	685
Volledig getest	685
Het MVC-onderdeel van Zend Framework	686
Installatie van Zend Framework	686
De FrontController	689
De MVC-structuur	690
Samenvatting Zend Framework MVC	694
Een Artikellijst in Zend Framework	694
De Controller	694
De Model-klassen	697
Het View-script	698
Zend Framework componenten	700
Meer weten over Zend Framework?	701

<b>23 De toekomst van PHP</b>	<b>703</b>
PHP 5.4	703
Geen Safe-mode meer	704
Optimalisatie	704
Achterwaartse compatibiliteit (downward compatibility)	704
PHP 5.x	706
Ondersteuning van Unicode	706
PHP is niet Unicode-veilig	707
PHP 6 of PHP 7?	710
De toekomst van PHP	711
Overeenkomsten met andere talen	711
Verschillen met andere talen	711
Ontwikkeling van frameworks en standaardapplicaties op PHP	712
<b>Index</b>	<b>715</b>

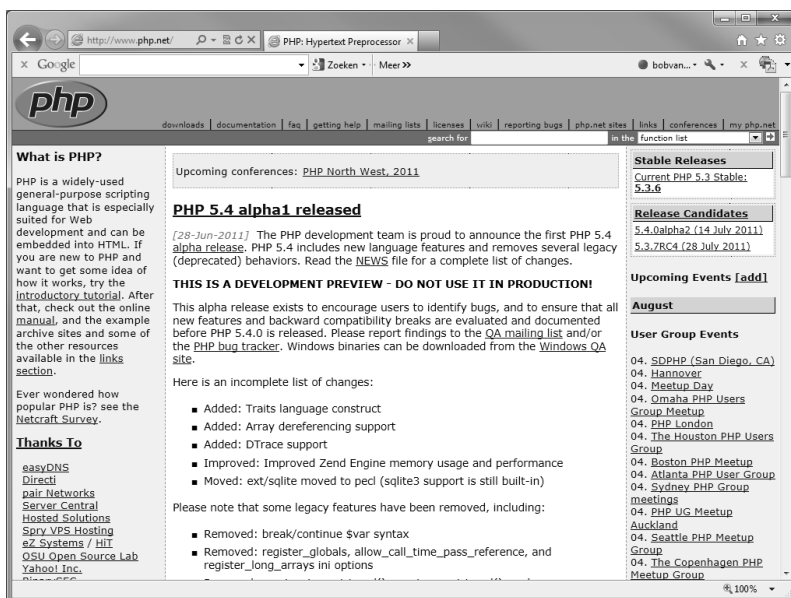
## Hoofdstuk 1

# PHP als scriptingtaal

*PHP is een zogeheten scriptingtaal. In tegenstelling tot de traditionele programmeertalen, zoals C++ en Pascal, bestaat PHP uit een aantal onafhankelijke pagina's, vergelijkbaar met html. Deze pagina's zijn door koppelingen (links) met elkaar verbonden. In dit hoofdstuk bekijken we PHP als scriptingtaal. Wat zijn eigenlijk de mogelijkheden van PHP? Hoe verhoudt PHP zich tot bijvoorbeeld html? We beginnen met de algemene opbouw van PHP en de verhouding tot html. We zullen onze eerste PHP-programmaatjes schrijven. Dit hoofdstuk is tevens het vertrekpunt van onze webshop. We zullen snel zien dat we eerst een goede opzet moeten maken voor de shop, die als rode draad door het boek loopt. Pas daarna begint het echte programmeren. De internetwereld gaat snel... PHP levert een bijdrage om aan de huidige vraag naar dynamische websites te kunnen voldoen. Het programmeren daarvan is leuk, dus laten we snel een start maken!*

## ■ Wat is PHP?

Traditioneel worden websites ontwikkeld in html (*hypertext markup language*). Html wordt meestal een taal genoemd. Eigenlijk is html niet zozeer een programmeertaal, maar meer een stijlconventie. Door middel van tags bepalen we hoe tekst, afbeeldingen en andere objecten er in de browser uitzien.



■ Afbeelding 1.1  
www.php.net.

Willen we een html-pagina bijvoorbeeld laten reageren op een actie van de gebruiker, dan kunnen we gebruikmaken van JavaScript. Deze taal wordt ingebed in de html-code. JavaScript is een zogeheten clientside scriptingtaal. Dit betekent dat de code op de clientcomputer, dus het werkstation, wordt uitgevoerd<sup>1</sup>. De combinatie van html en JavaScript wordt vooral gebruikt om aan de clientside acties ‘af te vangen’ en te controleren; denk bijvoorbeeld aan aanwijseffecten (*rollovers*), korte animaties enzovoort. We noemen deze combinatie van html en JavaScript ook wel dhtml.

## Dynamische sites

De eisen die aan het web worden gesteld, zijn door de ontwikkeling van internet steeds verder toegenomen. Waar een aantal jaren geleden statische websites nog voldeden, maken tegenwoordig steeds meer sites gebruik van dynamische elementen: een vacature zoeken bij uw favoriete bedrijf of een receptje zoeken om vanavond een heerlijke schotel op tafel te zetten. In deze tijd van e-commerce zijn er ook tal van sites die helemaal dynamisch opgezet zijn. Denk aan het enorme aantal webwinkels en zoekmachines. Al deze sites hebben één ding met elkaar gemeen: ze maken gebruik van een scriptingtaal en een database.

## PHP als scriptingtaal

Een traditionele applicatie draait continu. De toepassing is verantwoordelijk voor een geheel proces: van elke interactie met de gebruiker, de invoer van gegevens tot en met de verwerking ervan. Wanneer de gebruiker bijvoorbeeld gegevens invoert, kan de applicatie die invoer onmiddellijk beïnvloeden, controleren enzovoort. Het proces verloopt realtime. Bij een tekstverwerker tikt de gebruiker een regel in. Het programma reageert door de karakters op het scherm te plaatsen en de invoer meteen op spelfouten te controleren. Het gehele proces verloopt onder verantwoording van de applicatie.

Een scriptingtaal werkt anders. Deze komt slechts op één moment aan bod. Wanneer de gebruiker een nieuwe pagina ophaalt, wordt deze door de server van de vaste schijf afgehaald en doorgestuurd naar de client. Wanneer er gebruik wordt gemaakt van een scriptingtaal, wordt aan deze gang van zaken nog een element toegevoegd. De server zal in dat geval de gevraagde pagina eerst naar de *parser* (interpretatieprogramma) van de scriptingtaal sturen om verwerkt te worden. Pas daarna stuurt de server de pagina naar de client toe.

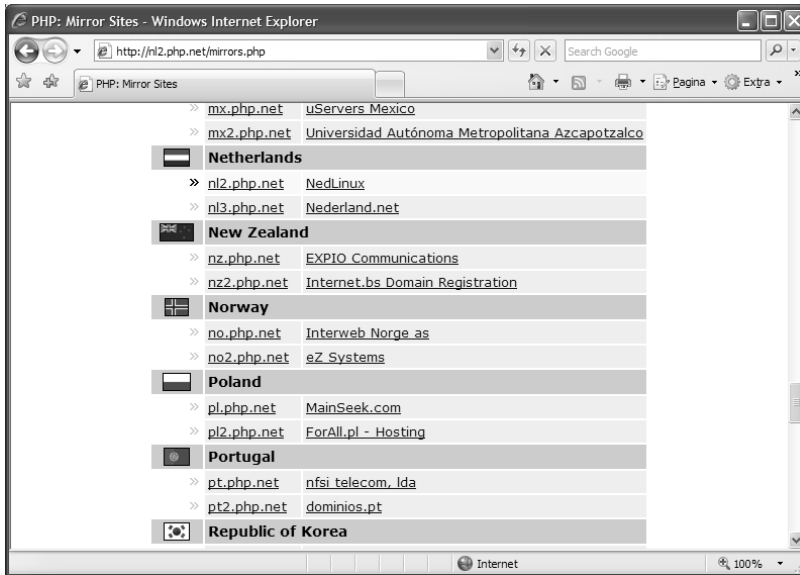
---

1 JavaScript kan in bepaalde gevallen ook serverside worden gebruikt, maar dit wordt vrijwel niet toegepast.



## Gebruik een snelle mirrorsite

De officiële website [www.php.net](http://www.php.net) krijgt veel bezoekers. Daarom kunt u voor een snelle verbinding beter gebruikmaken van een van de vele (Nederlandse) mirrors. De officiële mirrors zijn bekend onder de namen [nl2.php.net](http://nl2.php.net) en [nl3.php.net](http://nl3.php.net). Een volledig overzicht staat op [www.php.net/mirrors.php](http://www.php.net/mirrors.php).



**Afbeelding 1.2**  
Een overzicht van PHP-mirrorsites.

Er bestaat een groot aantal verschillende scriptingtalen; bekende voorbeelden zijn PHP, ASP.NET, JSP, Perl en Python. PHP en ASP.NET zijn de meest gebruikte. ASP.NET (onderverdeeld in VB.NET en C#) is het vlaggenschip van Microsoft op webapplicatiegebied. Het .NET-framework is zeer uitgebreid en biedt mogelijkheden tot het bouwen van zeer complexe webapplicaties.

PHP mag zich in grote populariteit verheugen. De ontwikkeling van PHP is begonnen in 1994. Rasmus Lerdorf wilde meer dan alleen een statische website. Het idee voor PHP, in die begindagen een afkorting van Professional Home Page, was geboren. De eerste versies van de scriptingtaal konden niet veel meer dan het afhandelen van een webformuliertje: de gegevens controleren en de inhoud versturen. PHP is echter volwassen geworden en mag zich een volwaardig lid van de familie der scriptingtalen noemen.

Tegenwoordig biedt PHP mogelijkheden om zeer complexe webapplicaties te ontwikkelen, geschoeid op professionele leest. Werd de scriptingtaal twee jaar geleden nog gezien als een leuk open-sourceproject, tegenwoordig wordt het toegepast door beursgenoteerde bedrijven.

## PHP en MySQL: een koppel

MySQL wordt vaak in één adem genoemd met PHP. MySQL is een populair open-sourcedatabasesysteem. De reden dat de combinatie PHP/MySQL vaak wordt genoemd, is omdat ze samen verantwoordelijk zijn voor duizenden database-gestuurde websites.



Afbeelding 1.3

Wanneer een pagina wordt opgehaald, wordt de PHP-code door de server geparsed. De server zoekt op zijn beurt contact met de database. Het resultaat wordt teruggestuurd naar de client.

Databasegestuurde websites zijn sites die gebaseerd zijn op een combinatie van een scriptingtaal en een database.<sup>2</sup> PHP wordt in dit geval gebruikt om informatie op te slaan in de database, de database te bevragen en het resultaat te presenteren in de browser. De meeste webapplicaties zijn terug te brengen tot deze processen. Een webwinkel bijvoorbeeld heeft aan de ene kant een interface voor de eigenaar. Hij kan artikelen invoeren, foto's uploaden, prijzen aanpassen enzovoort. De bezoeker daarentegen kan de artikelen bekijken of een selectie maken van artikelen. Wil hij een artikel bestellen, dan vraagt het systeem om gegevens zoals naam en rekening-



Afbeelding 1.4

De officiële site van MySQL: [www.mysql.com](http://www.mysql.com).

2 Een database is een structuur waar informatie in kan worden ondergebracht.



nummer. Deze worden opgeslagen in het systeem. De beheerder kan deze gegevens in de administratieve interface ophalen en verwerken: een factuur afdrukken, de bestelling als afgehandeld noteren enzovoort. Het geheel van processen komt ook hier voornamelijk neer op het bewerken van de data. PHP vertaalt de wensen van de gebruiker naar databasefuncties en stuurt MySQL aan.

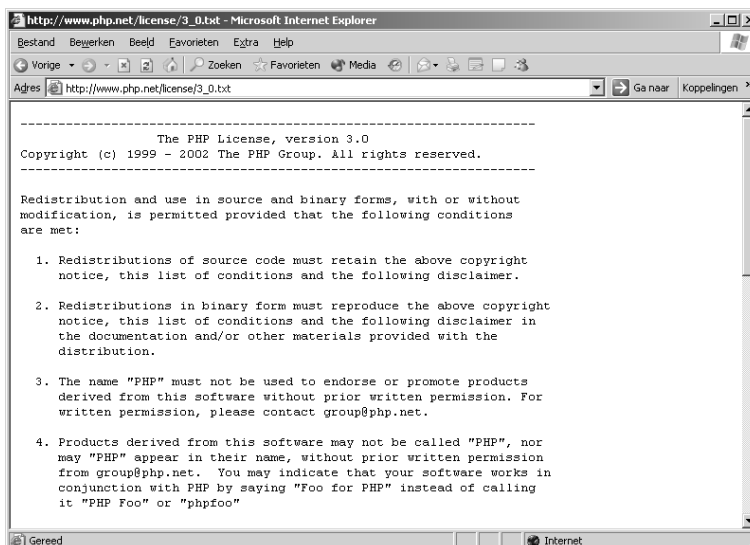
## ■ Waarom PHP?

PHP is erg populair en nog steeds in opmars. Het succes van PHP is te danken aan het open-sourcemodel. Het principe van open-source is dat de software vrij te gebruiken is en door iedereen aangepast mag worden. PHP wordt uitgebracht onder een eigen open-sourcelicentie. Deze licentie kent weinig beperkingen. Versie 3 werd nog uitgebracht onder de GNU General Public Licence. Kortweg houdt deze in dat iedereen de software mag gebruiken en aanpassen, maar dat de ontwikkelaar zich verplicht eventuele aanpassingen kosteloos beschikbaar te stellen aan derden, oftewel publiceert onder de GNU-licentie. De PHP-ontwikkelaarsgroep vindt dat deze licentie te veel beperkingen oplegt en daarom gebruiken ze sinds versie 4 een eigen licentie.



### Vragen over de PHP-licentie

Op [www.php.net/license/](http://www.php.net/license/) is een groot aantal veelgestelde vragen gepubliceerd.



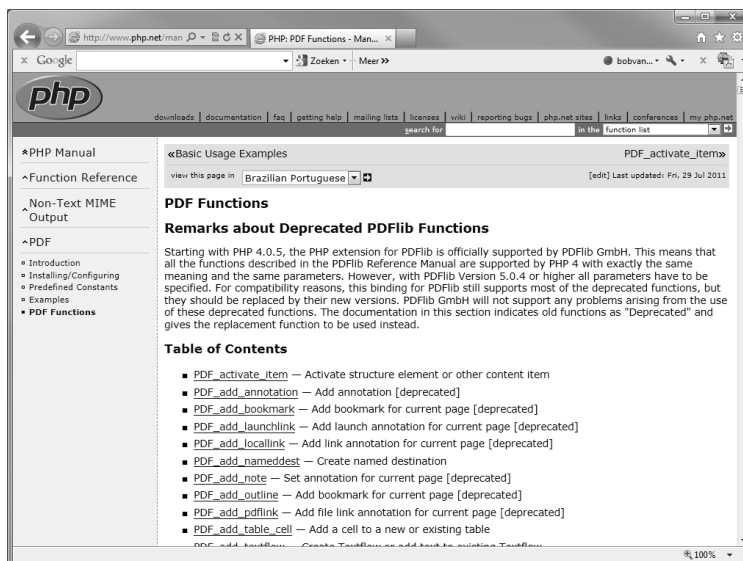
■ Afbeelding 1.5

*PHP wordt geleverd onder een open-sourcelicentie.*

Open-sourcesoftware mag zich in grote populariteit verheugen. Er wordt – vooral door grote bedrijven – nog wel eens gedacht dat open-sourcesoftware wel van B-kwaliteit zal zijn, maar dat is vaak niet het geval. De open-sourcemarkt is volwassen aan het worden. Het marktaandeel zal toenemen, omdat open-sourcesoftware zoals Linux steeds gebruiksvriendelijker wordt. Daarnaast breidt de support en ander-soortige dienstverlening op dit gebied enorm uit. Ook de hoge licentiekosten van commerciële programmatuur spelen een belangrijke rol. Een webserver met een combinatie van Linux (besturingssysteem), Apache (webserver), PHP (scriptingtaal) en MySQL (database) is kosteloos, terwijl voor vergelijkbare commerciële programmatuur vele duizenden euro's betaald moeten worden. Inmiddels heeft deze combinatie zich ruimschoots bewezen: men schat dat meer dan zestig procent van de webserver gebruikmaakt van Apache, in veel gevallen gecombineerd met PHP en MySQL.

## Libraries

PHP wordt op open-sourcebasis ontwikkeld door programmeurs uit de hele wereld. Op de mastersite [www.php.net](http://www.php.net) zijn tal van verschillende versies te vinden. PHP is enorm in beweging. En dat is te merken. Regelmatig verschijnen er nieuwe versies met steeds meer mogelijkheden. Daarnaast maakt PHP gebruik van zogeheten *libraries*, grote bibliotheken met bepaalde functionaliteit. Zo zijn er libraries op grafisch gebied, voor het genereren van pdf-bestanden, Flash-bestanden en xml. Ook bestaan er gespecialiseerde, commerciële libraries.



■ Afbeelding 1.6

PDF Library is een voorbeeld van een van de vele PHP-libraries.

## Frameworks

Relatief nieuw is het gebruik van zogeheten *frameworks* in PHP. Een framework lijkt qua gebruik op een library, maar is geschreven in PHP (libraries zijn dat per definitie niet). Het Zend Framework is een van de belangrijke frameworks. Het bevat functies voor authenticatie, configuratie, templating enzovoort.



Afbeelding 1.7

Zend Framework: [framework.zend.com](http://framework.zend.com).

## Platformonafhankelijkheid

PHP is steeds platformonafhankelijker geworden. Waren de eerste versies alleen geschikt voor Linux, de huidige versies zijn ook geschikt voor Windows en Macintosh. PHP kan prima samenwerken met Internet Information Server. Ook de ondersteuning van databases van Microsoft is geen enkel probleem. Op die manier kan onder Windows samengewerkt worden met een database van MS-SQL server.

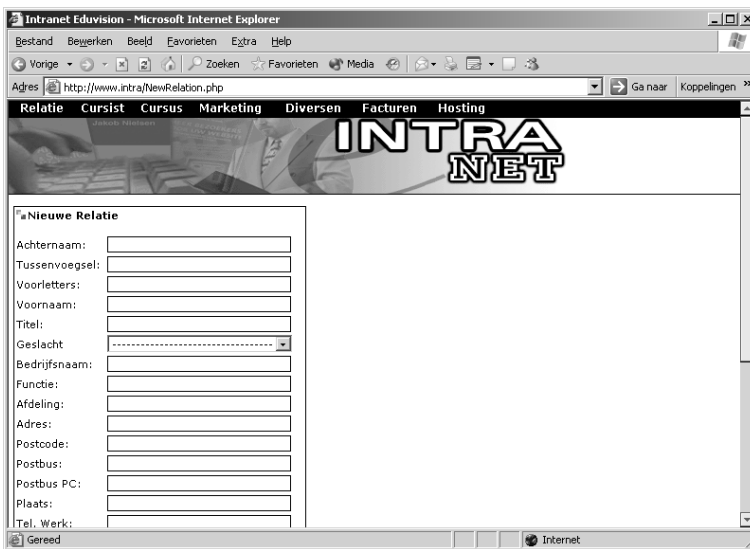
## De toekomst

Voor dynamische websites is een goede toekomst weggelegd. Als we naar de markt kijken, dan valt op dat het aantal databasegestuurde websites enorm toeneemt. De groei is logisch te verklaren. Bij zijn opkomst was het World Wide Web het domein van gespecialiseerde professionals. Door de enorme populariteit van het web, is de vraag naar geavanceerde functionaliteit toegenomen. Een website van de huidige generatie moet gemakkelijk te onderhouden zijn door iemand die verder geen verstand van webdesign heeft. Door middel van een contentmanagementsysteem wordt het onderhoud van een site gedistribueerd in een organisatie. Elk onderdeel is verantwoordelijk voor zijn eigen deel van de website.

Een dynamische website impliceert eigenlijk informatie op maat. Dat kan bijvoorbeeld aan de hand van een persoonlijk profiel of persoonlijke wensen. Een gebruiker krijgt alleen de informatie te zien die hij daadwerkelijk wil zien: de juiste artikelen in de cd-shop en de meest interessante vacatures.

## Intranet

Een relatief onontgonnen gebied dat erg belangrijk gaat worden, is het intranet. Steeds meer grotere organisaties hebben een eigen intranet. Een intranet is een website die alleen beschikbaar is binnen een bepaalde organisatie. Het kan afgesloten zijn van de buitenwereld (fysiek of bijvoorbeeld door middel van een *firewall*), maar het kan ook voor een deel over het openbare netwerk van internet lopen. In het laatste geval spreken we van een *virtual private network* (VPN). Wanneer een bedrijf verschillende vestigingen heeft en/of medewerkers thuis op het bedrijfsnetwerk moeten kunnen werken, wordt vaak een VPN gebruikt. Het mag duidelijk zijn dat beveiliging bij een VPN een grote rol speelt.



**Afbeelding 1.8**

*Intranetapplicatie van een opleidingsbedrijf.*

Een intranet kan een ‘gewone’ website zijn. Het zou bijvoorbeeld een lijst van medewerkers, een vacaturebank en mededelingen kunnen bevatten. Steeds vaker kunnen we spreken van een echte intranetapplicatie. De website vervult dan de rol die vroeger het administratiesysteem toekwam. Het bevat alle contactgegevens, agenda’s, maar ook het artikelenbestand, bestellingen, factuurgegevens enzovoort. In dit soort toepassingen worden vaak internetgerelateerde zaken zoals e-mail (webmail) meteen geïntegreerd. Voor PHP is inmiddels een aantal open-source CRM (Customer Relationship Management) applicaties ontwikkeld. Daarvan is Vtiger ([www.vtiger.com](http://www.vtiger.com)) de bekendste.



**Afbeelding 1.9**  
*Vtiger als voorbeeld van een intranetapplicatie.*

## Webapplicaties

Een intranetsysteem biedt veel voordelen ten opzichte van een traditioneel systeem. De intranetapplicatie draait op een server. Op de client is slechts een browser nodig om het systeem te kunnen benaderen. De applicatie is daarmee platformafhankelijk geworden, want voor elk platform is wel een goede browser beschikbaar.

Omdat PHP relatief laagdrempelig is, is de opzet van een dergelijk systeem in relatie tot een traditioneel systeem verhoudingsgewijs gemakkelijk te realiseren. Eventueel zou er zelfs gekozen kunnen worden voor uitbreiding van een bestaand systeem zoals Vtiger. Deze applicatie biedt standaard namelijk uitbreidingsmogelijkheden. Terwijl een traditionele toepassing meestal moeilijk uit te breiden is, is Vtiger vanwege het open-sourcekarakter relatief gemakkelijk uit te breiden. Omdat veel werk al gedaan is, worden kosten bespaard.

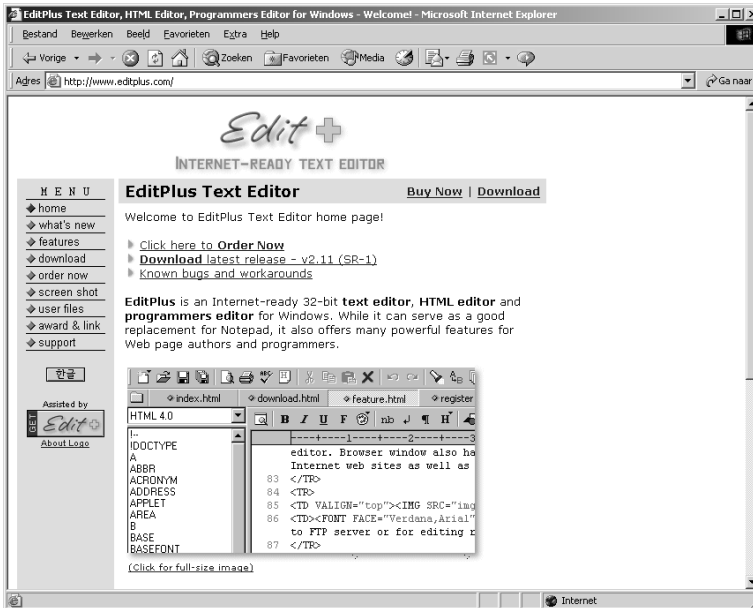
## ■ Werking van PHP

PHP is een serverside scriptingtaal. Dat betekent dat de PHP-code door de server geïnterpreteerd wordt en pas daarna naar de client wordt gestuurd. Hoe gaat ontwikkelen met PHP dan in zijn werk?

PHP-code kan in een willekeurige teksteditor (bijvoorbeeld Kladblok) geschreven worden. Er zijn echter voldoende handige editors, zoals EditPlus ([www.edit-plus.com](http://www.edit-plus.com)), CrimsonEditor ([www.crimsoneditor.com](http://www.crimsoneditor.com)) en PSPad ([www.pspad.com](http://www.pspad.com)) die de PHP-opdrachten kennen en opdrachten, variabelen,

# Hoofdstuk 1

## PHP als scriptingtaal



Abbeelding 1.10

EditPlus.

waarden enzovoort in verschillende kleuren weergeven. PHP Designer ([www.mpsoftware.dk](http://www.mpsoftware.dk)) is speciaal voor PHP ontwikkeld. Dit programma is veel uitgebreider dan de gemiddelde teksteditor. Een enorme hulp voor de programmeur! De meest professionele ondersteuning vindt u in Zend Studio van het commerciële PHP-bedrijf Zend ([www.zend.com](http://www.zend.com)). Een gelijkwaardige omgeving vindt u in de vorm van de open-sourceomgeving Eclipse ([www.eclipse.org](http://www.eclipse.org)). Er is een speciale PHP-versie van Eclipse beschikbaar die ook door Zend wordt ontwikkeld. U vindt deze op [www.zend.com/pdt](http://www.zend.com/pdt). Ik raad u aan met een eenvoudige editor, zoals EditPlus, te beginnen. De Zend-producten zijn namelijk vrij ingewikkeld en bieden pas een echte meerwaarde voor de gevorderde programmeur.

Nadat de code is geschreven, wordt die geüpload naar de webserver (die uiteraard PHP moet ondersteunen). Wanneer een bezoeker een willekeurige, statische html-pagina opvraagt, stuurt de server die meteen naar de client. Wanneer er echter een dynamische PHP-pagina wordt opgevraagd, dan zal de server de PHP-code eerst naar de parser (interpretatieprogramma) sturen. Pas wanneer de code geïnterpreteerd is, stuurt de server het resultaat door naar de client.

Wat is het resultaat van de geïnterpreteerde code? In principe is dat html. In feite laten we de PHP-code html genereren.

```
echo "<b>Deze regel wordt naar de browser gestuurd</b>";
```

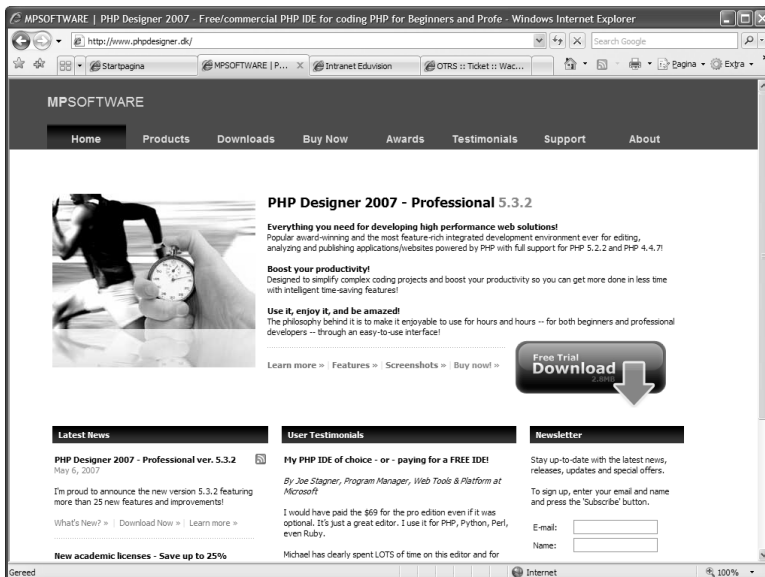
De instructie hierboven zal resulteren in deze html-code:

```
<b>Deze regel wordt naar de browser gestuurd</b>
```

Naast html kunnen we PHP in principe alles laten genereren wat naar een browser kan worden gestuurd, bijvoorbeeld JavaScript maar ook een gif-afbeelding.

Het is niet in alle gevallen handig om rechtstreeks via internet op een server te werken. Gelukkig kunnen we een willekeurige computer gemakkelijk als server laten fungeren (zie ook hoofdstuk 2). In dat geval zal het schema er iets anders uitzien.

Houd er rekening mee dat de werking van PHP van server tot server kan verschillen. De internetaanbieder (ISP: Internet Service Provider) kan PHP voor een groot gedeelte instellen, waardoor er verschillen kunnen ontstaan in configuratie tussen de ene ISP en de andere. De afwijkingen hebben meestal betrekking op rechten. Wat mag een ontwikkelaar wel binnen PHP en wat niet? Uit beveiligings-oogpunt kan de ISP besluiten bepaalde rechten strikter in te stellen.



**Afbeelding 1.11**

*PHP Designer.*

## Databasegestuurde websites

De meeste sites die ontwikkeld worden met PHP, zijn zogeheten databasegestuurde websites. Al deze websites hebben een gemeenschappelijke noemer, namelijk dat ze een database als hart hebben. Maar wat is een database nu eigenlijk precies?

Een database is een model, waarbinnen informatie gestructureerd wordt. Eigenlijk vormt een database een afspiegeling van de werkelijkheid. Stel dat we kunst verzamelen. De kunstcollectie is onder te brengen in een databasemodel. Een werk uit de collectie bestaat bijvoorbeeld uit een objectnummer, een maker, afmetingen, het

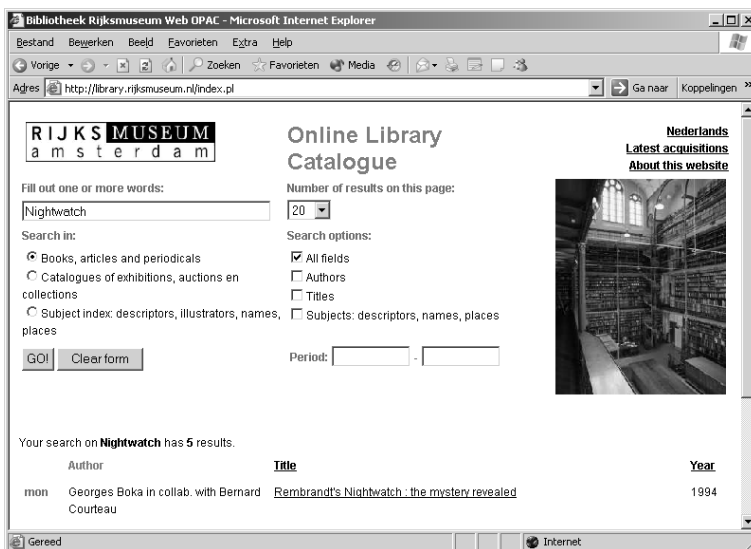
soort werk (schilderij, beeldhouwwerk enzovoort), een datering, materiaalkeuze, een omschrijving en een afbeelding. Deze kenmerken van het object vormen een afspiegeling ervan; ze zijn niet equivalent aan het object.

Door de objecten onder te brengen in een databasestructuur, kunnen we het overzicht bewaren. We kunnen gemakkelijk lijsten bekijken van de objecten, we kunnen ze sorteren of erin zoeken. Ik wil bijvoorbeeld een lijst van alle schilderijen uit de verzameling of een lijst van alle werken die dateren van voor 1978. Het interessante van databases is, dat we er vragen aan kunnen stellen.

De vragen hierboven zijn handmatig ook vrij gemakkelijk te beantwoorden. Databasevragen kunnen echter veel ingewikkelder en gedetailleerder van aard zijn. Hierdoor kunnen complexe vraagstellingen beantwoord worden, die binnen een niet-geautomatiseerd proces moeilijk te beantwoorden zijn, bijvoorbeeld omdat het veel te veel tijd zou kosten om tot een antwoord te komen.

Op het web maken we vooral gebruik van databases om bezoekersinformatie op maat te bieden. We stellen gaandeweg het bezoek een profiel van de bezoeker op en bieden op basis daarvan informatie aan. Een andere mogelijkheid is om een bezoeker zelf te laten zoeken. In ieder geval wordt er tijdens het virtuele bezoek steeds een zoekvraag voor de database geformuleerd.

Een bezoeker komt bijvoorbeeld op de website van het Rijksmuseum. Hij kan een virtuele tour maken door de collectie. Is hij in een bepaald werk geïnteresseerd, dan kan hij daar aanvullende informatie over opvragen. In databasetermen gebeurt er het volgende: de bezoeker wordt langs verschillende objecten uit de database geleid.



**Afbeelding 1.12**

*Een zoektocht naar materiaal over de Nachtwacht, afkomstig uit de database van het Rijksmuseum ([www.rijksmuseum.nl](http://www.rijksmuseum.nl)).*



Wanneer hij uitgebreidere informatie wil, wordt een zoekvraag aan de database gesteld. Het resultaat daarvan wordt vervolgens in de browser gepresenteerd.

Wat voor de collectie van het Rijksmuseum geldt, gaat ook op voor een online dvd-shop, zoekmachine, e-learningstelsel, gastenboek, forum enzovoort. De lijst van toepassingen van databasegestuurde websites is enorm. Omdat een databasegestuurde website de belangrijkste toepassing is van PHP, zullen we in dit boek een complete webshop uitwerken.

## De webshop

Het uitgangspunt van dit boek is het ontwikkelen van een webshop. Een webshop is immers een goed voorbeeld van een databasegestuurde website.

In onze webshop is van alles te koop: van boeken en cd's tot en met speelgoedtreintjes. De verschillende artikelen worden gerubriceerd in diverse categorieën. Alle cd's vallen bijvoorbeeld in de categorie CD. Bezoekers hoeven dan niet rechtstreeks te zoeken naar een specifieke cd, maar kunnen ook bladeren in de categorie CD.

Natuurlijk bevat onze database ook klantgegevens. Om een bestelling uit te voeren, moeten we hier immers over beschikken. Elke bestelling wordt door de klant ingevoerd en door de administrator verder afgehandeld.

Welke kennis hebben we nodig om een shop te ontwikkelen? Natuurlijk databasekennis; dat spreekt voor zich. We maken een databasemodel en implementeren dat vervolgens in MySQL. Daarnaast hebben we aanvullende html- en PHP-kennis nodig.



Afbeelding 1.13

Wehkamp: een voorbeeld van een webshop.

Kennis van formulieren is noodzakelijk. We gaan er in dit boek van uit dat u een redelijk ervaren html-er bent en het ontwerpen van formulieren geen grote geheimen voor u heeft. Natuurlijk zullen we uw formulierenkennis opfrissen.

Een formulier is een onlosmakelijk onderdeel van de webshop. We gebruiken het om klantgegevens in te voeren en bezoekers de mogelijkheid te geven te zoeken in de database.

Sessies en *cookies* gebruiken we om het gebruikersprofiel tijdelijk te bewaren. Wie is onze bezoeker? Wat heeft hij tot nu toe besteld?

De webshop bevat op dit moment nog veel geheimen. Aan het eind van dit deel bent u echter zelf in staat er een te ontwikkelen. Alvorens we daadwerkelijk met die ontwikkeling beginnen en dus met PHP aan de slag gaan, moeten we de *scripting engine* installeren.

## www.leer-php.nl

Ik heb Leer-php.nl geïntroduceerd in mijn eerste boek, *Leer jezelf MAKKELIJK PHP*. Duizenden gebruikers zijn geregistreerd en maken geregeld gebruik van het forum. Het PHP-veld is immers continu aan verandering onderhevig. Daarom is een aantal belangrijke zaken, zoals links, adressen van belangrijke sites, errata van dit boek enzovoort opgenomen op een aparte website: [www.leer-php.nl](http://www.leer-php.nl). U vindt hier ook alle broncode van de scripts die in dit boek worden besproken. Ook kunt u de werking van de scripts zien. In elke schermafdruk vindt u de url naar het desbetreffende script. Daarnaast kunt u vragen stellen aan collega-programmeurs op het forum.



Afbeelding 1.14  
[www.leer-php.nl](http://www.leer-php.nl).