

Bouwsteen
Inleiding programmeren in PHP

Studiejaar 1
Code IPPHP

Stenden University
Education
Van Schaikweg 94
Po box 2080
7801 CB Emmen
tel: 0591 853100
www.stenden.com

Inleiding programmeren in PHP

Version 1.6: 23 januari 2011



Studiejaar 2010-2011

Bouwsteen coördinator:	Auteurs :
Naam: Jeroen Pijpker	Naam: Bert Meijerink / Jeroen Pijpker
Code: Pij	Code: Mk/Pij
Room: 131	Room: 130/131
Tel: 0591-853277	Tel: 0591-853270/853277
e-mail: jeroen.pijpker@stenden.com	e-mail: bert.meijerink@stenden.com jeroen.pijpker@stenden.com

Voorwoord

In deze bouwsteen leer je de eerste beginselen van het programmeren, in ons geval PHP.

Dat doe je door een boek te bestuderen maar daarnaast vooral door opdrachten uit te voeren. De reden daarvan is dat programmeren een vaardigheid is die je alleen maar onder de knie kunt krijgen door veel te oefenen.

De taal die we gebruiken is PHP, een programmeertaal die veel wordt gebruikt in webpagina's. Het primaire doel van deze bouwsteen is echter niet het leren van PHP, maar het leren programmeren.

Het programmeren van PHP zal in ons geval plaatsvinden in de ontwikkelomgeving Netbeans. Na deze bouwsteen heeft de student een goede basis gelegd voor een goede start van de bouwsteen "Unleash your potential in PHP".

Aan het eind van deze periode zal er in projectvorm een Professionele website worden gemaakt. In dit project zal PHP gebruikt gaan worden.

De studielast voor deze bouwsteen bedraagt 84 uur (3 EC).

Uit de verschillende toegepaste methodieken op de Stenden hogeschool, is in verleden gebleken dat Inleiding programmeren in PHP de meest geschikte methode is voor het aanleren van PHP voor een beginnend programmeur. Door evaluaties van studenten is het programma zodanig aangepast dat er meer tijd beschikbaar is voor programmeren in PHP.

Bert Meijerink en Jeroen Pijpker
Emmen, 23 januari 2011

Inhoudsopgave

INHOUDSOPGAVE	5
1 INTRODUCTIE	8
1.1 BOUWSTEEN RATIONALE	8
1.2 COMPETENTIES	9
1.3 BOUWSTEEN THEMA	10
1.4 BOUWSTEEN DOELSTELLINGEN	10
1.5 VOORKENNIS	10
1.6 CONVENTIES	10
1.7 VERSIEBEHEER	10
2 TOETSING	11
2.1 ALGEMEEN	11
2.2 BEOORDELING PRAKTIJKTOETS	11
2.3 ACTIEVE PARTICIPATIE	12
2.4 BEOORDELING WERKCOLLEGES	12
2.5 BOUWSTEEN HERKANSING	12
3 PROGRAMMA	13
3.1 INTRODUCTIE	13
3.1.1 Hoorcolleges	13
3.1.2 Werkcolleges	13
3.2 PROGRAMMAOVERZICHT	14
3.3 WEEKPROGRAMMA'S	15
3.3.1 Introductiecollege	15
3.3.2 Hoorcollege 1 (Hoofdstuk 1, eerste deel en bijlagen 1 t/m 4)	15
3.3.3 Werkcollege 1 (Hoofdstuk 1, activiteiten 1 t/m 3)	15
3.3.4 Hoorcollege 2 (Hoofdstuk 1, tweede deel)	15
3.3.5 Werkcollege 2 (Hoofdstuk 1, activiteiten 4 t/m 9)	16
3.3.6 Hoorcollege 3 (Hoofdstuk 2, eerste deel en bijlage 5)	16
3.3.7 Werkcollege 3 (Hoofdstuk 2, activiteiten 1 t/m 5)	16
3.3.8 Hoorcollege 4 (Hoofdstuk 2, tweede deel)	17
3.3.9 Werkcollege 4 (Hoofdstuk 2, activiteiten 6 t/m 8)	17
3.3.10 Hoorcollege 5 (Hoofdstuk 3)	17
3.3.11 Werkcollege 5 (Hoofdstuk 3, activiteiten 1 t/m 2)	18
3.3.12 Hoorcollege 6 (Hoofdstuk 4)	18
3.3.13 Werkcollege 6 + 7 + 8 (Individuele eindopdracht)	18
3.3.14 Zelfstudie oefentoets	19
4 STRUCTUUR & ORGANISATIE	20
4.1 BOUWSTEEN CONTACT UREN	20
5 LITERATUUR/PROGRAMMATUUR	21
5.1 VERPLICHT LEERSTOF	21
5.2 NASLAGWERK	21

5.3	PROGRAMMATUUR.....	21
5.4	VOORBEELD PROGRAMMA'S.....	21
6	BOUWSTEEN EVALUATIE	22
7	BIJLAGEN.....	23
BIJLAGE 1:	OPZETTEN VAN EEN WEB ONTWIKKELOMGEVING	24
B1.0	General	24
B1.1	Installing Java.....	25
B1.2	Installing Netbeans	25
B1.3	Installing and Configuring the XAMPP Package	27
B1.4	Checking XAMPP Installation.....	30
B1.5	Installing and Enabling the XDebug Debugger.....	31
BIJLAGE 2:	OPZETTEN VAN EEN PROJECT IN NETBEANS	32
B2.0	General	32
B2.1	Debugging your PHP Project.....	34
BIJLAGE 3:	TOELICHTING LEERSTOF HOOFDSTUK 1, EERSTE DEEL	36
BIJLAGE 4:	TOELICHTING LEERSTOF HOOFDSTUK 1, TWEEDE DEEL.....	37
BIJLAGE 5:	TOELICHTING LEERSTOF HOOFDSTUK 2.....	38
BIJLAGE 6:	TOELICHTING LEERSTOF HOOFDSTUK 4.....	40
BIJLAGE 7:	ACTIVITEITEN HOOFDSTUK 1.....	43
B7.1	Activiteit 1	43
B7.2	Activiteit 2.....	43
B7.3	Activiteit 3.....	44
B7.4	Activiteit 4.....	44
B7.5	Activiteit 5.....	44
B7.6	Activiteit 6.....	45
B7.7	Activiteit 7.....	45
B7.8	Activiteit 8.....	45
B7.9	Activiteit 9.....	45
BIJLAGE 8:	ACTIVITEITEN HOOFDSTUK 2.....	46
B8.1	Activiteit 1	46
B8.2	Activiteit 2.....	46
B8.3	Activiteit 3.....	46
B8.4	Activiteit 4.....	46
B8.5	Activiteit 5.....	47
B8.6	Activiteit 6.....	47
B8.7	Activiteit 7.....	47
B8.8	Activiteit 8.....	52
BIJLAGE 9:	ACTIVITEITEN HOOFDSTUK 3.....	53
B9.1	Activiteit 1	53
B9.2	Activiteit 2.....	53
BIJLAGE 10:	INDIVIDUELE EINDOPDRACHT	54
BIJLAGE 11:	CODE CONVENTIES	56
BIJLAGE 12:	SCORING RUBRICS	58
BIJLAGE 13:	OEFENTOETS.....	60
OPGAVE 1	61

OPGAVE 2	62
OPGAVE 3	66
BIJLAGE 1.....	67
BIJLAGE 2.....	68
BIJLAGE 3.....	69

1 Introductie

1.1 Bouwsteen rationale

Programmeren is één van de basiselementen voor de informaticus van dit moment.

Het programmeren in deze bouwsteen zal worden gedaan in de taal PHP. De taal PHP wordt wereldwijd veel gebruikt door webdevelopers.

De ervaring die wordt opgedaan in deze bouwsteen ga je toepassen in het project van de 1e periode: het bouwen van een professionele website.

Kenmerkende beroepssituatie

Karin is na haar HBO informatica opleiding als junior software engineer aan het werk bij WebSystem. Binnen WebSystem is Karin verantwoordelijk voor het complete implementatie traject van content management systemen bij klanten. Binnen deze content management systemen speelt PHP een belangrijke rol.

Een van de projecten waarbij Karin is betrokken is de online webshop van een **EcoGoodies**. Binnen dit project was Karin verantwoordelijk voor het programmeren van de extra gewenste functionaliteiten binnen het content management systeem. Een voorbeeld hiervan is dat de klant **EcoGoodies** een extra module nodig was voor een nieuw concept. Deze module zoekt automatisch uit de bestaande aanbieders van elektriciteit de meeste groene aanbieder voor de regio van de klant.

Doordat WebSystem een relatief kleine onderneming is komt Karin met meerdere facetten van projecten in aanraking. Daardoor kan Karin zich ontwikkelen op meerder vakgebieden.

1.2 Competenties

Binnen deze bouwsteen wordt als beginnend beroepsprofessional gewerkt aan een drietal competenties die betrekking hebben op het analyseren, ontwerpen en realiseren van infrastructuur op niveau 1.

	Analyseren	Adviseren	Ontwerpen	Realiseren	Beheren
Gebruikers interactie					
Bedrijfsprocessen					
Software	<ul style="list-style-type: none"> • Uitvoeren van een eenvoudige requirementsanalyse bestaande uit data-analyse en procesanalyse. • Opstellen van een eenvoudige acceptatietest. (niveau 1)		<ul style="list-style-type: none"> • Met een schematechniek opstellen van een eenvoudig statisch model en ontwerpen van een eenvoudig algoritme. (niveau 1)	<ul style="list-style-type: none"> • Ontwikkeltool bouwen, testen en beschikbaar stellen van een stand-alone single threaded applicatie en een eenvoudige database met bijbehorende queries. (niveau 1)	
Infrastructuur					
Hardware interfacing					

1.3 Bouwsteen thema

Deze bouwsteen vormt de basis voor het maken van een professionele website met (X)HTML/CSS en PHP.

1.4 Bouwsteen doelstellingen

Binnen deze bouwsteen wordt gewerkt aan de volgende doelstellingen:

1. De student kan een casus analyseren;
2. De student toont aan dat hij/zij op de inhoudsgebieden, welke aan de orde komen in hoofdstuk 1 t/m 5 van het PHP boek beheerst en kan toepassen.
3. De student toont aan dat hij/zij een eenvoudig PHP programma kan maken;
4. De student toont aan dat hij/zij kan omgaan met de Netbeans IDE;
5. De student toont aan dat hij/zij de mogelijkheden kent van een moderne programmeer taal, PHP;
6. De student toont aan dat hij/zij een traject kan opzetten voor het ontwikkelen van een eenvoudig PHP programma;
7. De student toont aan dat hij/zij kan programmeren in PHP.

1.5 Voorkennis

Algemene kennis van de computer en MS Windows (op gebruikersniveau) zoals bijvoorbeeld op de Havo bij het vak informatiekunde aan de orde is geweest.

1.6 Conventies

Een paar conventies met betrekking tot lettertypen zijn opgesteld om onderscheid te kunnen maken tussen tekst en programmeercode.

Normale tekst	Verdana. (tekengrootte 10)
Programma code	Courier new. (tekengrootte 10)
Programma output	Courier new. (tekengrootte 10)

Afspraken met betrekking tot naamgeving en indeling van programmacode kun je vinden in bijlage 11.

1.7 Versiebeheer

Versie	Datum	Auteur	Omschrijving
1.5	17-09-2010	B. Meijerink	Opname versiebeheer; Oefentoets toegevoegd.
1.6	23-01-2011	B. Meijerink	Scoring rubrics aangepast.

2 Toetsing

2.1 Algemeen

Om te beoordelen of je de doelstellingen van deze bouwsteen hebt behaald wordt er een individuele eindtoets (praktijktoets) gehanteerd.

Tabel 2.1 geeft een overzicht van de eindtoets met relevante informatie over de normering, punten en credits.

Tabel 2.1 Overzicht toetsing

Test method	Max. points	Norm %	Norm in points	Credits	Deadline	Resit
Praktijktoets	100	55%	55	3 EC		
Total	100	55%	55	3 EC		

2.2 Beoordeling praktijktoets

De eindbeoordeling is een praktijktoets waarbij de student een webpagina moet (af)maken achter de computer. Deze praktijktoets zal dezelfde moeilijkheidsgraad hebben als de behandelde stof en zal binnen de gestelde tijd moeten worden afgerond.

Bij deze eindtoets mag de student gebruik maken van de volgende middelen die hij nodig acht: Boeken, moduleboek en eigen uitwerkingen van opdrachten. **(Andere als genoemde middelen, zoals bijvoorbeeld Internet, zijn niet toegestaan en niet beschikbaar.)**

Om aan de eindtoets te mogen deelnemen moeten alle opdrachten uit de bouwsteen *Inleiding programmeren in PHP* zijn besproken **en afgetekend** op de vrijdag voor de toetsweek. **(Deelname aan de toets is slechts toegestaan indien alle voorgaande studieactiviteiten zijn afgetekend.)**

Alle opdrachten uit de bouwsteen *Inleiding programmeren in PHP* dienen individueel uitgevoerd en besproken te worden met de docent. **Als tijdens de bespreking blijkt dat de student de opdracht niet zelf heeft uitgevoerd, wordt de opdracht met een onvoldoende beoordeeld.**

Voor een herkansing wordt per opdracht eenmalig een vervangende opdracht verstrekt waar dezelfde eisen voor gelden.

De beoordeling van de praktijktoets zal plaatsvinden aan de hand van het beoordelingsformulier (Scoring Rubrics) welke te vinden is in bijlage 12.

Deze bouwsteen wordt met een voldoende beoordeeld als voor de praktijktoets de norm is gehaald.

2.3 Actieve participatie

De student dient alle opdrachten voorafgaand aan de praktijktoets af te ronden in de daarvoor betreffende weken. De mogelijkheid voor het aftekenen van deze opdrachten heeft de student tijdens de ingeroosterde begeleide werkcolleges.

2.4 Beoordeling werkcolleges

Tijdens de werkcolleges zullen de volgende onderdelen beoordeeld worden:

1. De aanwezigheid van de student;
De student moet minimaal bij 75% van de werkcolleges de volledige tijd aanwezig zijn geweest.
2. De te maken opgaven;
De student moet alle opdrachten met een voldoende hebben afgerond.

Punt 2 moet afgerond worden in de week dat het desbetreffende werkcollege op het rooster (spoorboekje) staat.

Al deze 2 onderdelen moeten met een voldoende zijn beoordeeld voor men kan deelnemen aan de afsluitende eindtoets.

2.5 Bouwsteen herkansing

Studenten die niet slagen voor de praktijktoets en/of eindtoets kunnen een herkansing doen voor het onderdeel dat onvoldoende is.

3 Programma

3.1 Introductie

Eén van meest voorkomende beroepen in het werkveld van de ICT is software engineer. De basis voor software engineers (maar ook ander ICT'ers) is een gedegen kennis van programmeertalen.

Als PHP programmeur ga je (meestal) in teamverband samenwerken in een team van (ervaren) PHP programmeurs en ander vakspecialisten. Je wordt mede verantwoordelijk voor het onderhouden en doorontwikkelen van bestaande pakketten en helpt bij het bouwen en testen van nieuwe applicaties voor zowel intern als extern gebruik.

Deze bouwsteen bevat een aantal (wekelijkse) hoor- en werkcolleges. Tijdens de hoorcolleges zal de docent vaak een demonstratie geven en heeft de student de mogelijkheid om problemen aan te stippen. Na het hoorcollege kan de student zelfstandig aan de slag met de opdrachten.

Tijdens de werkcolleges heeft de student de mogelijkheid om problemen die zich voordoen bij het maken van de opdrachten te bespreken met de docenten. Tevens kunnen dan opdrachten worden afgetekend .

Het geheel zal afgesloten worden met een praktijktoets.

3.1.1 Hoorcolleges

Tijdens de hoorcolleges, welke een interactieve vorm hebben, komen verschillende facetten van het programmeren in PHP aan de orde. Dat kan bijvoorbeeld een introductie van het geheel zijn, of de uitleg van algoritmen.

Regelmatig zal echter een specifiek (programmeer)probleem aan de orde komen en zal de docent de oplossing hiervan stapsgewijs voordoen/uitleggen.

3.1.2 Werkcolleges

Tijdens de werkcolleges heeft de student de mogelijkheid om de onderdelen van de bijbehorende week af te laten tekenen. Het kan vanzelfsprekend eens voorkomen dat een student het werk van een week niet heeft af gekregen en dit kan dan in de week daaropvolgend worden afgetekend. De mogelijkheid bestaat echter niet dat een student aan het eind van de periode nog een groot gedeelte moet afronden.

De werkcolleges dienen vooraf voorbereid te worden.

Voor de werkcolleges geldt een aanwezigheidsplicht.

De docent is voor delen van het werkcollege beschikbaar.

.2 Programmaoverzicht

In onderstaand overzicht is een overzicht te vinden van de wekelijkse activiteiten.

Wk	Taak nr.	Studie activiteit
1	3.3.1	Introductiecollege.
	3.3.2	Hoorcollege 1. (Boek hoofdstuk 1 eerste deel en bijlagen 1 t/m 4.)
	3.3.3	Werkcollege 1. (Hoofdstuk 1, activiteiten 1 t/m 3.)
2	3.3.4	Hoorcollege 2. (Boek hoofdstuk 1, tweede deel)
	3.3.5	Werkcollege 2. (Hoofdstuk 1, activiteiten 4 t/m 9.)
3	3.3.6	Hoorcollege 3. (Boek hoofdstuk 2, eerste deel en bijlage 5.)
	3.3.7	Werkcollege 3. (Hoofdstuk 2, activiteiten 1 t/m 5.)
4	3.3.8	Hoorcollege 4. (Boek hoofdstuk 2, tweede deel.)
	3.3.9	Werkcollege 4. (Hoofdstuk 2, activiteiten 6 t/m 8.)
5	3.3.10	Hoorcollege 5. (Boek hoofdstuk 3.)
	3.3.11	Werkcollege 5. (Hoofdstuk 3, activiteiten 1 t/m 2.)
6	3.3.12	Hoorcollege 6. (Boek hoofdstuk 4 en bijlage 6.)
	3.3.13	Werkcollege 6. (Individuele eindopdracht.)
7	3.3.13	Werkcollege 7. (Individuele eindopdracht.)
8	3.3.13	Werkcollege 8. (Individuele eindopdracht.)
	3.3.14	Zelfstudie, oefentoets.
9		Beoordeling en afronding Inleiding programmeren in PHP opdracht. Eindtoets. Deelname alleen toegestaan als alle voorgaande activiteiten zijn afgetekend.

3.3 Weekprogramma's

3.3.1 Introductiecollege	
Week:	1.
Werkvorm:	Hoorcollege.
Duur:	1 uur.
Lesdoelen:	<ul style="list-style-type: none"> • De student verkrijgt een overzicht van de inhoud van de bouwsteen Inleiding programmeren in PHP. • De student kent de werkwijze en de beoordeling binnen de bouwsteen Inleiding programmeren in PHP.
Inhoud:	Tijdens het introductiecollege krijg je instructie over onder andere de werkwijze/werkvormen, beoordeling, materiaal en inhoud van de bouwsteen Inleiding programmeren in PHP.
Vorbereiden:	
Individuele opdrachten:	

3.3.2 Hoorcollege 1 (Hoofdstuk 1, eerste deel en bijlagen 1 t/m 4)	
Week:	1.
Werkvorm:	Hoorcollege.
Duur:	1 uur.
Lesdoelen:	<ul style="list-style-type: none"> • De student leert een Web ontwikkelomgeving op te zetten; • De student leert de mogelijkheden van de Netbeans ontwikkelomgeving door een demonstratie van de docent; • De student verkrijgt inzicht in de basis (begrippen) van PHP; • De student leert eenvoudige PHP scripts te maken. • De student verkrijgt inzicht in variabelen en constanten;
Inhoud:	Tijdens dit hoorcollege wordt er aandacht besteed aan: <ul style="list-style-type: none"> • De basisbeginselen van PHP; • PHP scripts; • Variabelen (Naming, defining, displaying etc).
Vorbereiden:	Bestudeer theorie hoofdstuk 1 tot "Working with Data Types"(blz. 30). Bestudeer bijlage 1,"Opzetten van een Web ontwikkelomgeving". Bestudeer bijlage 2,"Opzetten van een project in Netbeans". Bestudeer bijlage 3,"Toelichting leerstof hoofdstuk 1, eerste deel". Bestudeer bijlage 4,"Toelichting leerstof hoofdstuk 1, tweede deel".
Individuele opdrachten:	Bestudeer theorie hoofdstuk 1 tot "Working with Data Types"(blz. 30). Bestudeer bijlage 1,"Opzetten van een Web ontwikkelomgeving". Bestudeer bijlage 2,"Opzetten van een project in Netbeans". Bestudeer bijlage 3,"Toelichting leerstof hoofdstuk 1, eerste deel". Bestudeer bijlage 4,"Toelichting leerstof hoofdstuk 1, tweede deel".

3.3.3 Werkcollege 1 (Hoofdstuk 1, activiteiten 1 t/m 3)	
Week:	1.
Werkvorm:	Werkcollege.
Duur:	2 uur.
Lesdoelen:	<ul style="list-style-type: none"> • De student installeert de Netbeans ontwikkelomgeving inclusief aanvullende software; • De student installeert XAMPP; • De student leert eenvoudige scripts schrijven.
Inhoud:	Tijdens dit werkcollege wordt er aandacht besteed aan: <ul style="list-style-type: none"> • Het afdrukken van teksten; • Fouten uit een programma halen; • De student leert variabelen op een correcte manier te gebruiken.
Vorbereiden:	Installeer de Netbeans ontwikkelomgeving en aanvullende software. Installeer XAMPP.
Individuele opdrachten:	Installeer de Netbeans ontwikkelomgeving en aanvullende software. Installeer XAMPP. Voer de activiteiten 1 t/m 3 behorende bij hoofdstuk 1 uit. Deze zijn te vinden in bijlage 7.

3.3.4 Hoorcollege 2 (Hoofdstuk 1, tweede deel)	
Week:	2.
Werkvorm:	Hoorcollege.
Duur:	1 uur.
Lesdoelen:	<ul style="list-style-type: none"> • De student leert werken met data types; • De student verkrijgt inzicht in building expressions; • De student verkrijgt inzicht in operator precedence; • De student leert werken met type casting.
Inhoud:	Tijdens dit hoorcollege wordt er aandacht besteed aan: <ul style="list-style-type: none"> • Datatypes (numeric, boolean etc.); • Array's; • De verschillende operatoren.
Vorbereiden:	Bestudeer theorie hoofdstuk 1 vanaf " Working with Data Types" (blz. 30).
Individuele opdrachten:	Bestudeer theorie hoofdstuk 1 vanaf " Working with Data Types" (blz. 30).

3.3.5 Werkcollege 2 (Hoofdstuk 1, activiteiten 4 t/m 9)	
Week:	2.
Werkvorm:	Werkcollege.
Duur:	2 uur.
Lesdoelen:	<ul style="list-style-type: none"> De student leert operators op een correcte manier te gebruiken.
Inhoud:	Tijdens dit werkcollege wordt er aandacht besteed aan: <ul style="list-style-type: none"> Datatypen en operators en worden hiermee bewerkingen (berekeningen) uitgevoerd.
Vorbereiden:	
Individuele opdrachten:	Voer de activiteiten 4 t/m 9 behorende bij hoofdstuk 1 uit. Deze zijn te vinden in bijlage 7.

3.3.6 Hoorcollege 3 (Hoofdstuk 2, eerste deel en bijlage 5)	
Week:	3.
Werkvorm:	Hoorcollege.
Duur:	1 uur.
Lesdoelen:	<ul style="list-style-type: none"> De student leert werken met functies; De student verkrijgt inzicht de scope van een variabele; De student verkrijgt inzicht in beslissingsstructuren.
Inhoud:	Tijdens dit hoorcollege wordt er aandacht besteed aan: <ul style="list-style-type: none"> Functies; Beslissingsstructuren (if, else etc.).
Vorbereiden:	Bestudeer theorie hoofdstuk 2 tot "Repeating Code" (blz. 95). Bestudeer bijlage 5, "Toelichting leerstof hoofdstuk 2".
Individuele opdrachten:	Bestudeer theorie hoofdstuk 2 tot "Repeating Code" (blz. 95). Bestudeer bijlage 5, "Toelichting leerstof hoofdstuk 2".

3.3.7 Werkcollege 3 (Hoofdstuk 2, activiteiten 1 t/m 5)	
Week:	3.
Werkvorm:	Werkcollege.
Duur:	2 uur.
Lesdoelen:	<ul style="list-style-type: none"> De student leert functies en beslissingsstructuren op een correcte manier te gebruiken.
Inhoud:	Tijdens dit werkcollege wordt er aandacht besteed aan: <ul style="list-style-type: none"> Functies; De verschillende soorten van beslissingsstructuren.
Vorbereiden:	
Individuele opdrachten:	Voer de activiteiten 1 t/m 5 behorende bij hoofdstuk 2 uit. Deze zijn te vinden in bijlage 8.

3.3.8 Hoorcollege 4 (Hoofdstuk 2, tweede deel)	
Week:	4.
Werkvorm:	Hoorcollege.
Duur:	1 uur.
Lesdoelen:	<ul style="list-style-type: none"> • De student verkrijgt inzicht in herhalende code.
Inhoud:	Tijdens dit hoorcollege wordt er aandacht besteed aan: <ul style="list-style-type: none"> • While / do...while statements; • For / foreach statements.
Vorbereiden:	Bestudeer theorie hoofdstuk 2 vanaf "Repeating Code"(blz. 95).
Individuele opdrachten:	Bestudeer theorie hoofdstuk 2 vanaf "Repeating Code"(blz. 95).

3.3.9 Werkcollege 4 (Hoofdstuk 2, activiteiten 6 t/m 8)	
Week:	4.
Werkvorm:	Werkcollege.
Duur:	2 uur.
Lesdoelen:	<ul style="list-style-type: none"> • De student leert herhalende statements op een correcte manier te gebruiken.
Inhoud:	Tijdens dit werkcollege wordt er aandacht besteed aan: <ul style="list-style-type: none"> • Herhalende code.
Vorbereiden:	
Individuele opdrachten:	Voer de activiteiten 6 t/m 8 behorende bij hoofdstuk 2 uit. Deze zijn te vinden in bijlage 8.

3.3.10 Hoorcollege 5 (Hoofdstuk 3)	
Week:	5.
Werkvorm:	Hoorcollege.
Duur:	1 uur.
Lesdoelen:	<ul style="list-style-type: none"> • De student verkrijgt inzicht in strings; • De student leert te werken met strings.
Inhoud:	Tijdens dit hoorcollege wordt er aandacht besteed aan: <ul style="list-style-type: none"> • Construeren van tekst strings; • Werken met Single en Multiple strings.
Vorbereiden:	Bestudeer theorie hoofdstuk 3.
Individuele opdrachten:	Bestudeer theorie hoofdstuk 3.

3.3.11 Werkcollege 5 (Hoofdstuk 3, activiteiten 1 t/m 2)	
Week:	5.
Werkvorm:	Werkcollege.
Duur:	2 uur.
Lesdoelen:	<ul style="list-style-type: none"> • De student leert strings op een correcte manier te gebruiken.
Inhoud:	Tijdens dit werkcollege wordt er aandacht besteed aan: <ul style="list-style-type: none"> • Construeren van tekst strings; • Werken met Single en Multiple strings.
Vorbereiden:	
Individuele opdrachten:	Voer de activiteiten 1 t/m 2 behorende bij hoofdstuk 3 uit. Deze zijn te vinden in bijlage 9.

3.3.12 Hoorcollege 6 (Hoofdstuk 4)	
Week:	6.
Werkvorm:	Hoorcollege.
Duur:	1 uur.
Lesdoelen:	<ul style="list-style-type: none"> • De student verkrijgt inzicht in formulieren; • De student leert te werken met formulieren.
Inhoud:	Tijdens dit hoorcollege wordt er aandacht besteed aan: <ul style="list-style-type: none"> • Het maken van webformulieren; • Verzenden van de inhoud van formulieren.
Vorbereiden:	Bestudeer theorie hoofdstuk 4. Bestudeer bijlage 6, "Toelichting leerstof hoofdstuk 4".
Individuele opdrachten:	Bestudeer theorie hoofdstuk 4. Bestudeer bijlage 6, "Toelichting leerstof hoofdstuk 4".

3.3.13 Werkcollege 6 + 7 + 8 (Individuele eindopdracht)	
Week:	6 + 7 + 8.
Werkvorm:	Werkcollege.
Duur:	Week 6 2 uren, weken 7 en 8 elk 1 uur.
Lesdoelen:	<ul style="list-style-type: none"> • De student kan een wat grotere programmeeropdracht zelfstandig uitvoeren; • De student kan een gegeven opdracht zelfstandig analyseren.
Inhoud:	Tijdens deze werkcollege's kan de student: <ul style="list-style-type: none"> • vragen stellen met betrekking tot de individuele programmeeropdracht; • de verschillende stappen laten aftekenen; • actief bezig met het programma.
Vorbereiden:	
Individuele opdrachten:	Voer de gegeven individuele opdracht uitdracht uit zoals is aangegeven in bijlage 10.

3.3.14 Zelfstudie oefentoets	
Week:	8
Werkvorm:	Zelfstudie
Duur:	4 uur.
Lesdoelen:	<ul style="list-style-type: none">• Oefenen van het geleerde in deze module;
Inhoud:	Tijdens dit zelfstudiemoment kan de student aan de slag met een oefentoets.
Vorbereiden:	
Individuele opdrachten:	Bestudeer bijlage 13 en maak de opgaven.

4 Structuur & Organisatie

4.1 Bouwsteen contact uren

Het onderstaande schema geeft een overzicht van alle contacturen in deze bouwsteen.

Daarnaast wordt van studenten verwacht dat zij hun eigen (project)bijeenkomsten plannen waar zij kunnen werken aan de opdrachten. Dit geldt tevens voor de tijd die de student nodig heeft om (individuele) opdrachten voor te bereiden en te maken. Tevens geeft dit schema een goed overzicht van de verwachte studiebelasting per student.

Table 4.1: Student contact uren (SCU) per week:

Bouwsteen inleiding programmeren in PHP: Student Contacturen en Student Belastinguren																					
Werkvorm	Aantal groepen	Week 1		Week 2		Week 3		Week 4		Week 5		Week 6		Week 7		Week 8		Week 9		Totaal	
		SCU	SBU	SCU	SBU	SCU	SBU	SCU	SBU	SCU	SBU	SCU	SBU	SCU	SBU	SCU	SBU	SCU	SBU	SCU	SBU
HC	4	2	6	1	3	1	3	1	3	1	3	1	3	0	0	0	0	0	0	7,00	21,00
WC	2	2	6	2	6	2	6	2	6	2	6	2	6	1	3	1	3	0	0	14,00	42,00
ET	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	9	3,00	9,00
ZS	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-
		4	12,00	3	9,00	3	9,00	3	9,00	3	9,00	3	9,00	1	3,00	1	3,00	3	9,00	24,00	72,00

HC = Hoorcollege

PR = Practicum

ET = Eind toets

ZS = Zelfstudie uren

SCU = Student Contact Uren (45 minuten)

SBU = Studie Belasting Uren (60 minuten)

5 Literatuur/programmatuur

5.1 Verplichte leerstof

- Deze module.
- Boek: PHP Programming with MySQL, second edition.
Schrijvers: Don Gosselin, Diana Kokoska and Robert Easterbrooks.
Uitgever: Thomson.
ISBN-10: 0-538-46814-9.
ISBN-13: 978-0-538-46814-5.

5.2 Naslagwerk

- W3Schools: PHP: <http://www.php.net>
- PHP hulp: <http://www.phphulp.nl/>

5.3 Programmatuur

- Netbeans: <http://www.netbeans.org/>

5.4 Voorbeeld programma's

- Voorbeeldprogramma's zijn te vinden onder de course Inleiding programmeren in PHP.

6 Bouwsteen evaluatie

De bouwsteen zal worden geëvalueerd door middel van een vragenlijst aan het eind van de bouwsteen. Deze vragenlijst bevat alle onderdelen van de bouwsteen inclusief organisatorische aspecten, inhoud, kwaliteit van onderwijzend personeel, etc.

Wij willen je vriendelijk verzoeken deel te nemen aan deze evaluatie. De resultaten van deze evaluatie worden gebruikt om de volgende versie van deze bouwsteen te verbeteren.

7 Bijlagen

Bijlage 1: Opzetten van een Web ontwikkelomgeving

B1.0 General

This tutorial shows you how to configure your PHP development environment on the Windows operating system. The easiest way is to install and configure an AMP (Apache, MySQL, PHP) package. This tutorial shows you how to install the XAMPP package. There is also another way to install your Development Environment is to install and configure each component separately.

Contents:

1. Installing Java;
2. Installing Netbeans;
3. Installing and Configuring the XAMPP Package;
4. Checking XAMPP Installation;
5. Installing and Enabling the XDebug Debugger.

To follow this tutorial, you need the following software and resources.

Software or Resource	Version Required
NetBeans IDE	PHP download bundle 6.5
A PHP engine	Version 5. Included in XAMPP-Windows.
A web server	Apache HTTP Server 2.2 is recommended. Included in XAMPP Windows.
A database server	MySQL Server 5.0 is recommended. Included in XAMPP Windows.
A PHP debugger	XDebug 2.0 or later

Typically, development and debugging is performed on a local web server, while the production environment is located on a remote web server. This tutorial lets you set up a local web server. PHP support can be added to a number of local web servers (IIS, Xitami, and so on), but in this tutorial we use Apache HTTP Server. Apache server is included in the XAMPP AMP package used in this tutorial.

B1.1 Installing Java

Java is required to run Netbeans.

- Download the latest Java SDK from <http://java.com/nl/>.



Run the Java installer.

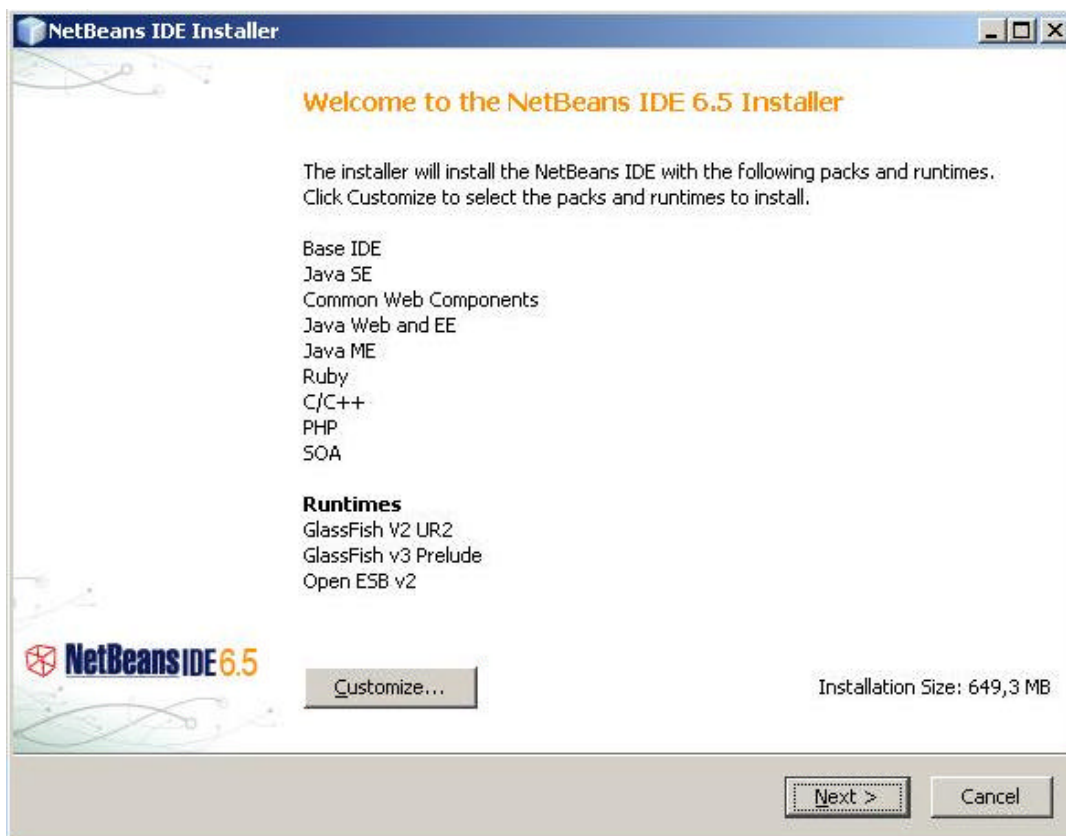
B1.2 Installing Netbeans

Netbeans is a free open-source Integrated Development Environment (IDE) for software developers. Netbeans runs on Windows, Linux, Mac OS X and Solaris.

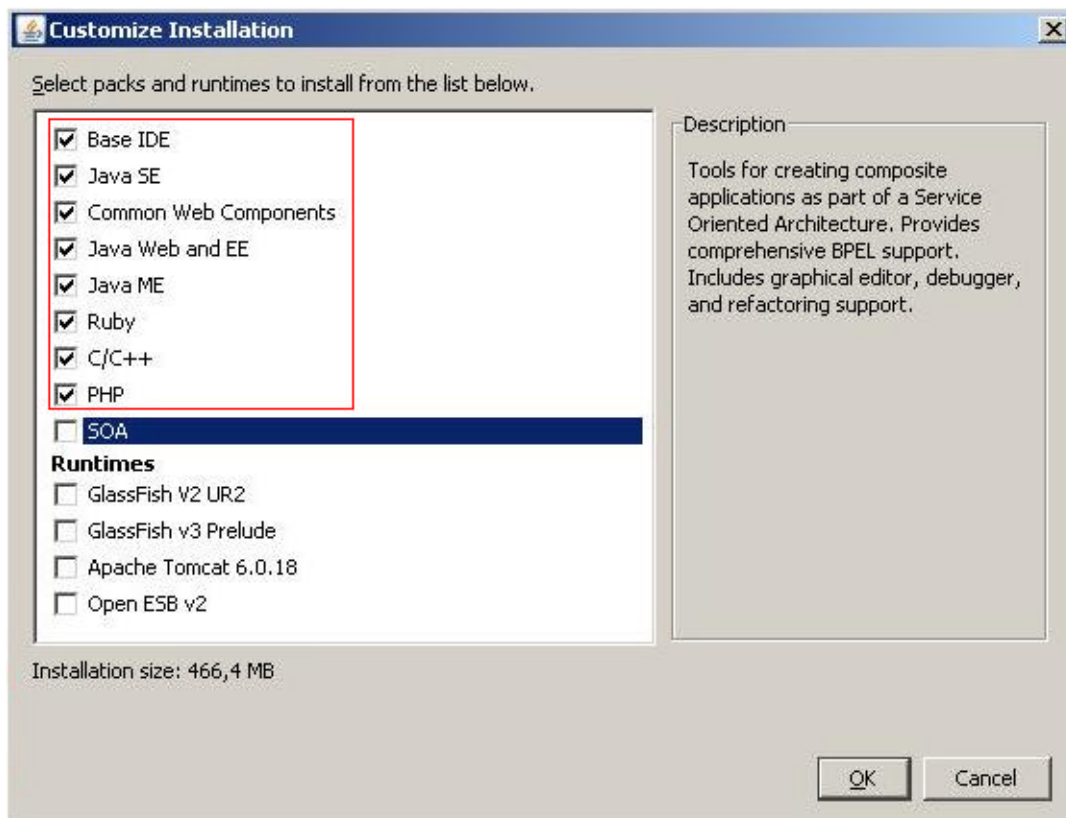
- Download the latest Netbeans build from <http://www.netbeans.org/downloads/index.html>.
You need to download the package with **All** the supported technologies.

Supported technologies *	Java SE	JavaFX	Java	Ruby	C/C++	PHP	All
Java SE	•	•	•				•
JavaFX		•					•
Java Web and EE			•				•
Java ME			•				•
Ruby				•			•
C/C++					•		•
PHP						•	•
SOA							•
Bundled servers							
GlassFish V2 UR2			•				•
GlassFish v3 Prelude				•			•
Apache Tomcat 6.0.18			•				•
	Download Free, 39 MB	Download Free, 63 MB	Download Free, 212 MB	Download Free, 59 MB	Download Free, 24 MB	Download Free, 25 MB	Download Free, 249 MB

- You need to customize the components to install.



- The following components you need to install.

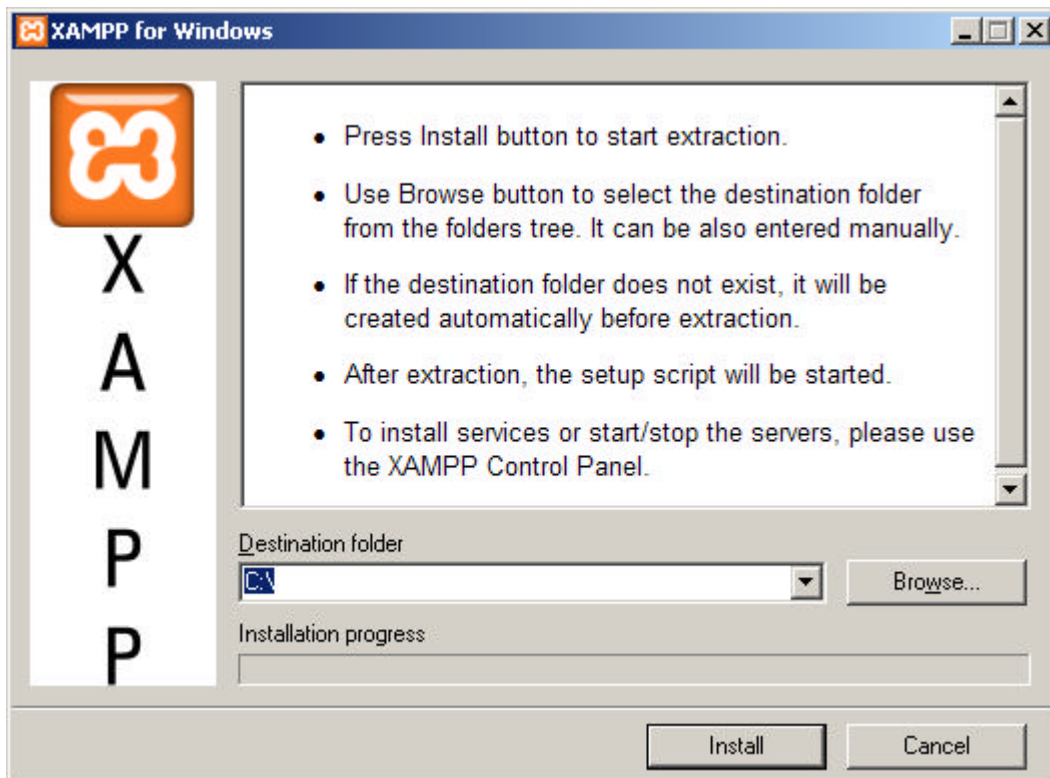


- Now you can go on with installing Netbeans.

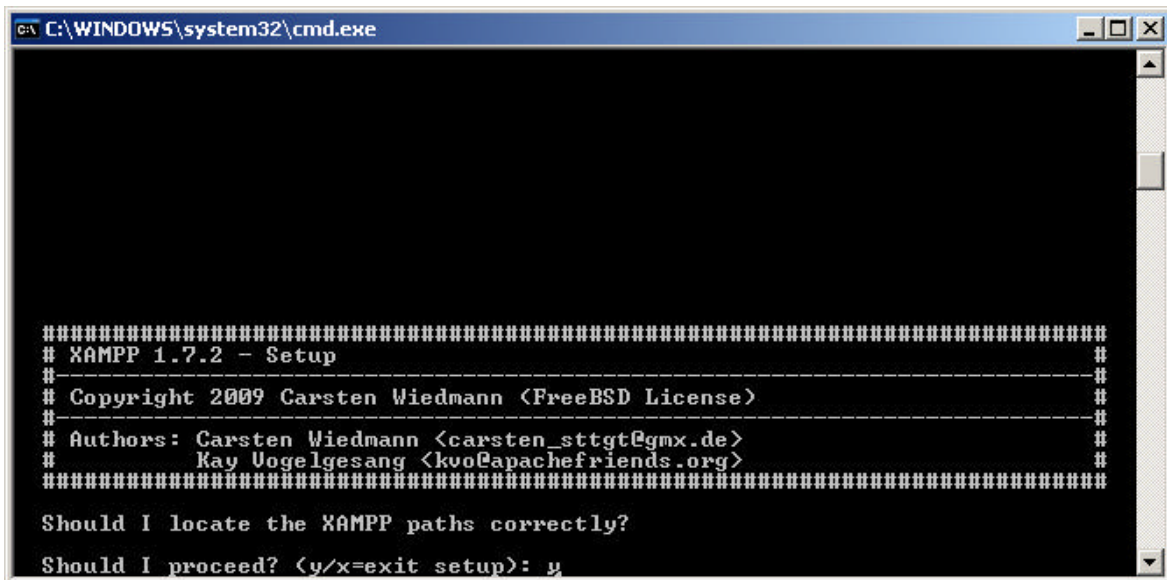
B1.3 Installing and Configuring the XAMPP Package

This section describes how to download, install, and configure the XAMPP package.

- Download the XAMPP package from <http://sourceforge.net/projects/xampp/>.
- When the download is completed, run the file xampp-win32-1.7.2.exe.
Tip: On Microsoft Vista, the User Access Control feature blocks the PHP installer from updating the httpd configuration. Disable UAC while you are installing XAMPP. See Microsoft Support for more information.
- In the destination folder specify the installation location.
- Use the Browse button, if necessary. Then click Install. This instruction suggest the C:/ root directory as the XAMPP location. You can choose any other folder and adjust your further steps accordingly.



- After XAMPP finished installing the following command prompt appears:



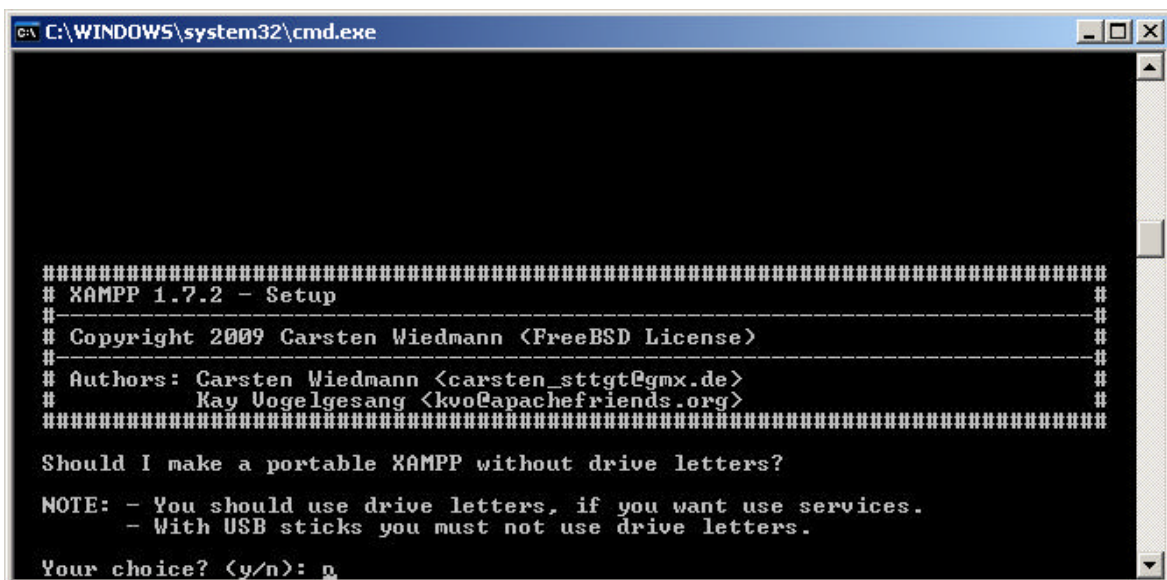
```
C:\WINDOWS\system32\cmd.exe

#####
# XAMPP 1.7.2 - Setup                                     #
#-----#
# Copyright 2009 Carsten Wiedmann <FreeBSD License>      #
#-----#
# Authors: Carsten Wiedmann <carsten_sttgt@gmx.de>       #
#           Kay Vogelgesang <kvo@apachefriends.org>       #
#####

Should I locate the XAMPP paths correctly?

Should I proceed? (y/x=exit setup): y
```

Enter 2 times "y"



```
C:\WINDOWS\system32\cmd.exe

#####
# XAMPP 1.7.2 - Setup                                     #
#-----#
# Copyright 2009 Carsten Wiedmann <FreeBSD License>      #
#-----#
# Authors: Carsten Wiedmann <carsten_sttgt@gmx.de>       #
#           Kay Vogelgesang <kvo@apachefriends.org>       #
#####

Should I make a portable XAMPP without drive letters?

NOTE: - You should use drive letters, if you want use services.
      - With USB sticks you must not use drive letters.

Your choice? (y/n): n
```

Enter "n"


```

C:\WINDOWS\system32\cmd.exe
Should I make a portable XAMPP without drive letters?

NOTE: - You should use drive letters, if you want use services.
      - With USB sticks you must not use drive letters.

Your choice? (y/n): n

relocating XAMPP...
relocate XAMPP base package
relocate Apache
relocate FileZilla FTP Server
relocate Mercury
relocate MySQL
relocate OpenSSL
relocate Perl
relocate PHP
relocate phpMyAdmin
relocate Sendmail
relocate Webalizer
relocate XAMPP Demopage
relocating XAMPP successful.

XAMPP is ready to use.

Press <Return> to continue:

```

```

C:\WINDOWS\system32\cmd.exe

#####
# XAMPP 1.7.2 - Setup                                     #
#-----#
# Copyright 2009 Carsten Wiedmann (FreeBSD License)      #
#-----#
# Authors: Carsten Wiedmann <carsten_sttgt@gmx.de>      #
#           Kay Vogelgesang <kvo@apachefriends.org>      #
#####

I have set the timezone in 'php.ini' and 'my.ini' to "Europe/Paris".

You should correct these values if my guess was wrong.

Press <Return> to continue:

```

```

C:\WINDOWS\system32\cmd.exe

#####
# XAMPP 1.7.2 - Setup                                     #
#-----#
# Copyright 2009 Carsten Wiedmann (FreeBSD License)      #
#-----#
# Authors: Carsten Wiedmann <carsten_sttgt@gmx.de>      #
#           Kay Vogelgesang <kvo@apachefriends.org>      #
#####

1. start XAMPP Control Panel
2. relocate XAMPP
   (current path: C:\xampp)
3. disable HTTPS (SSL)
4. disable Server Side Includes (SSI)
5. enable IPv4 only (current: IPv4/6 (auto))
6. disable mod_perl
7. disable Apache::ASP

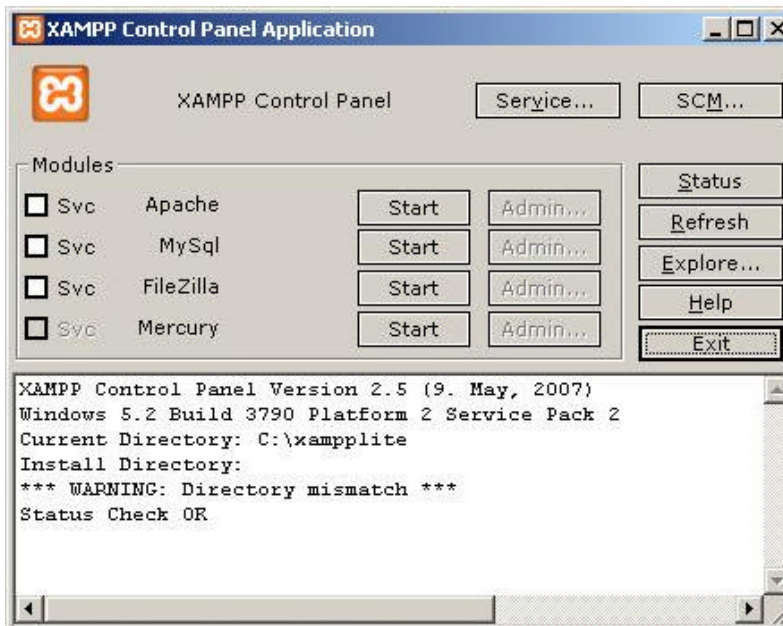
x Exit

Please choose (1-7/x): 1

```

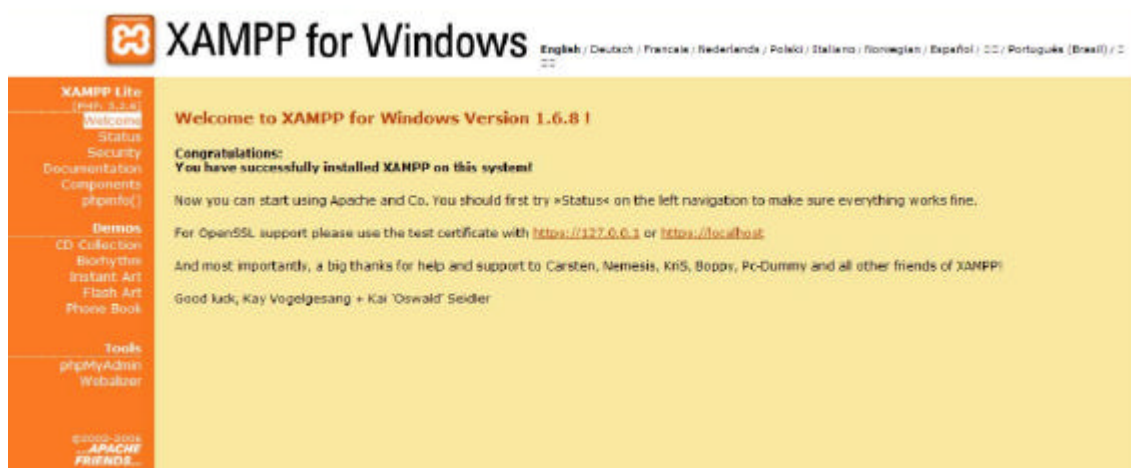
Choose 1 to start the XAMPP Control Panel

- Now the XAMPP Control Panel appears.
Now you can start the Apache webserver and the MySQL services.



B1.4 Checking XAMPP Installation

- Run your browser and enter the following URL: <http://localhost>. The XAMPP welcome page opens:



- To ensure that the Apache and MySQL servers have been installed as system services, restart your operating system, run the browser, and enter the <http://localhost> URL again. The XAMPP welcome page includes a menu in the left margin through which you can check the status of XAMPP components and run `phpinfo()`, among other useful features. `phpinfo()` returns a screen with configuration information about your XAMPP components.

B1.5 Installing and Enabling the XDebug Debugger

Because XAMPP Lite does not contain the XDebug debugger, you need to install and set up the XDebug as a component.

- Download XDebug 2.0 from <http://www.xdebug.org/download.php>;
- Copy the XDebug dll to a new folder called c:\xdebug;
- To attach XDebug to the PHP engine, locate the `php.ini` file and add the following lines to it:

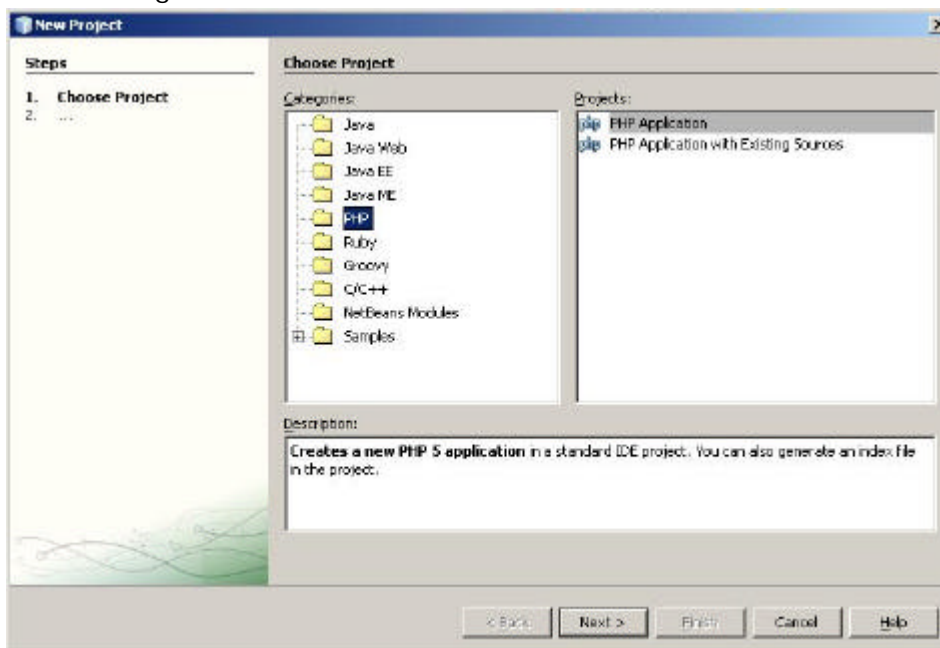
```
[XDebug]
zend_extension = "C:\xampp\php\ext\php_xdebug.dll"
xdebug.remote_enable=on
xdebug.remote_host=localhost
xdebug.remote_port=9000
xdebug.remote_handler=dbgp
```


Bijlage 2: Opzetten van een project in Netbeans

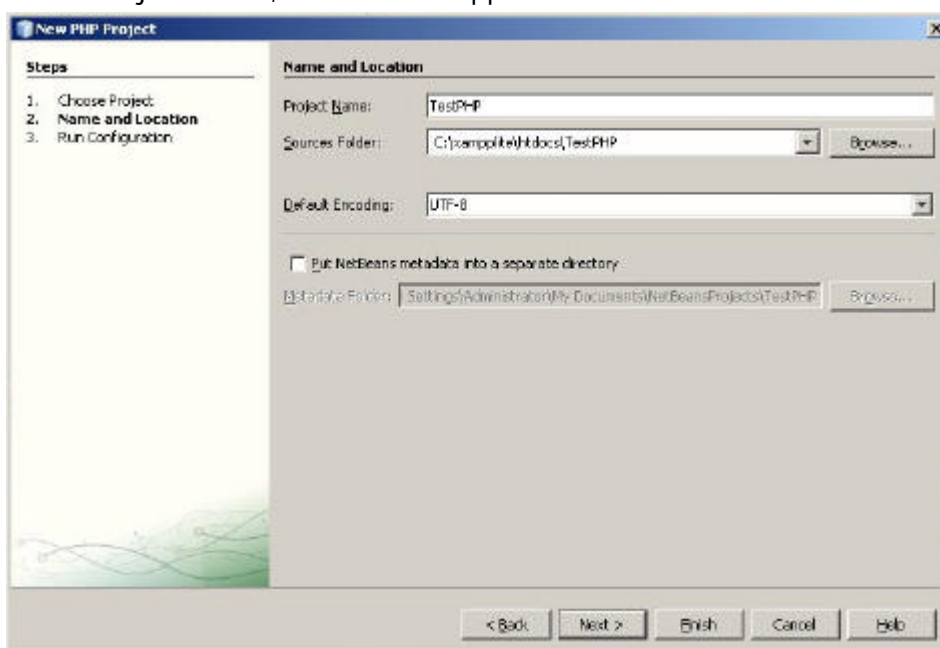
B2.0 General

To start PHP development in the NetBeans IDE for PHP, you first need to create a project. A project contains the information on the location of the project files and the way you want to run and debug your application (run configuration).

- Start the IDE, switch to the Projects windows, and choose File ? New Project. The Choose Project panel opens.
- In the Categories list, choose PHP.



- In the Projects area, choose PHP Application and click Next.

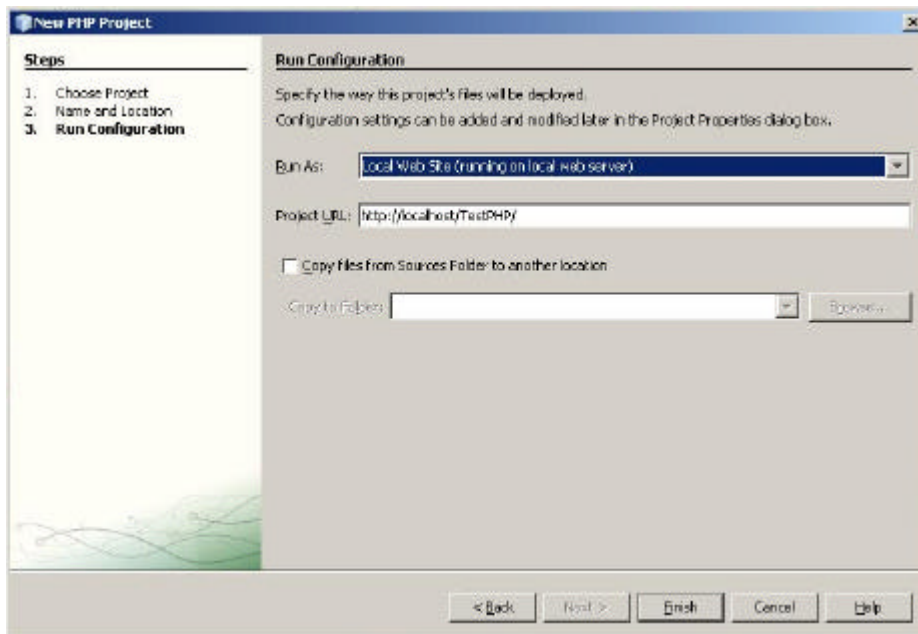


- In the Project Name text field, enter TestPHP. In the Sources Folder field, browse for your PHP document root (c:\xampplite\htdocs\), and create a subfolder there called TestPHP. The document root is the folder where the web server looks for files to open in the browser. The document root is specified in the web server configuration file. On Xampp, the document root is XAMPP_HOME/htdocs.

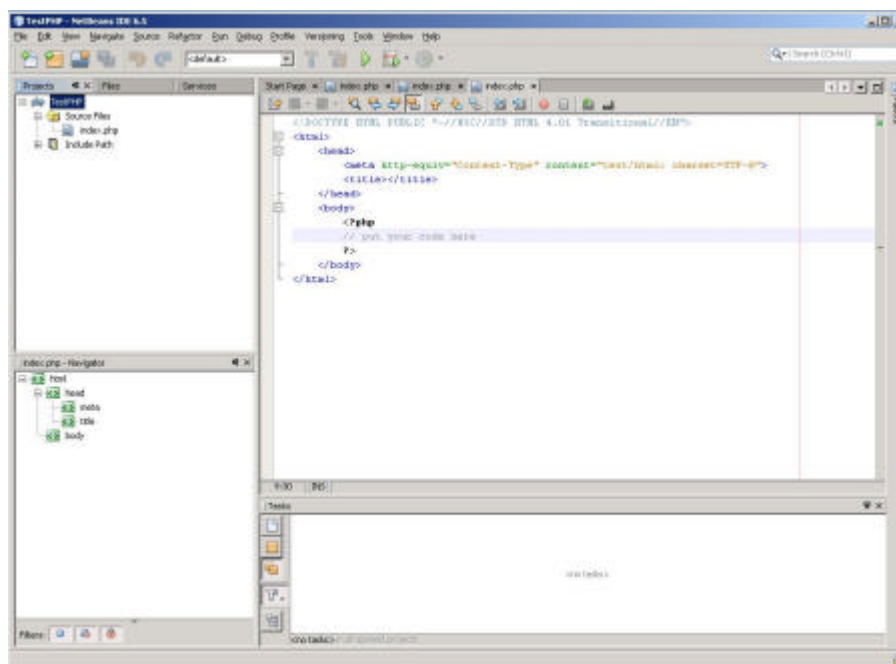
Leave all other field with their default values. Click Next. The Run Configuration windows opens.

In the Run As drop-down list, select Local Web Site. The project will run on your local Apache server. Your other options are to run the project remotely via FTP and to run it from the command line.

Leave the Project URL at default.



- Click Finish. The IDE creates the project.



B2.1 Debugging your PHP Project

A debugger is a program that lets programmers execute an application one step at a time. It typically provides functions to stop and start a program at selected point in the source code, and to examine the values of variables.

A debugger is a software tool that can helps in examining how an application executes. It can be used to find bugs.

First you need to create a new project

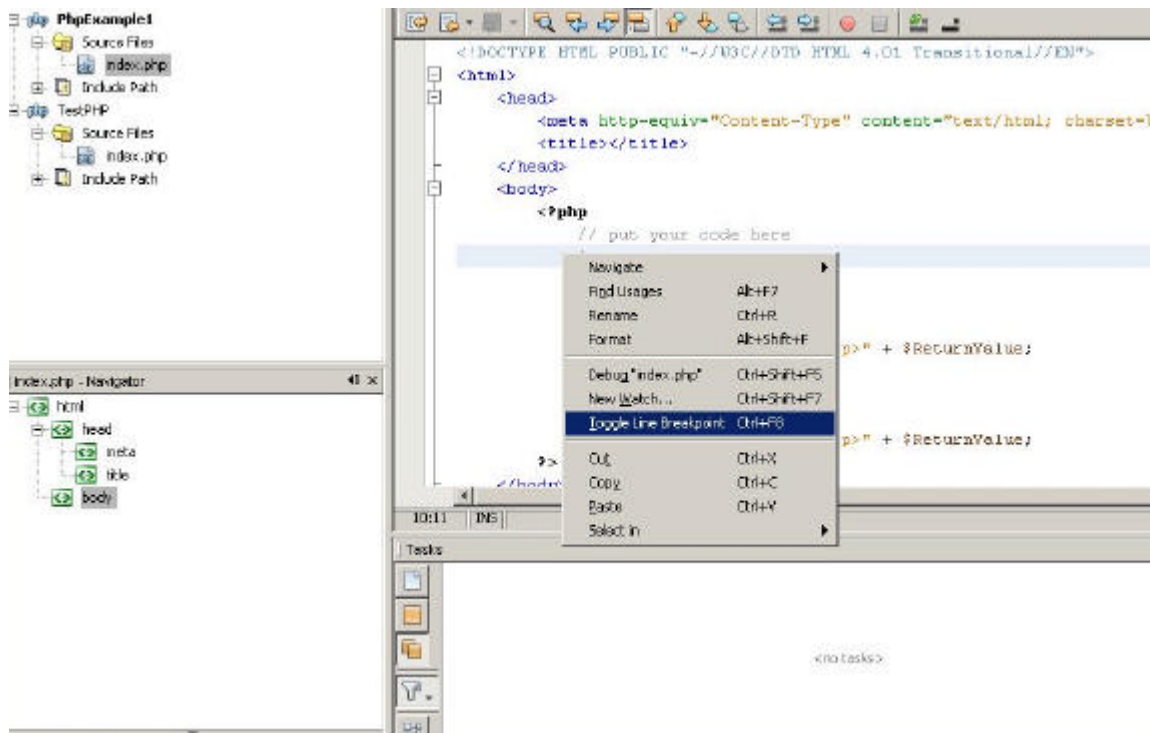
Edit the source code of the php file as follow:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
                                         charset=UTF-8">

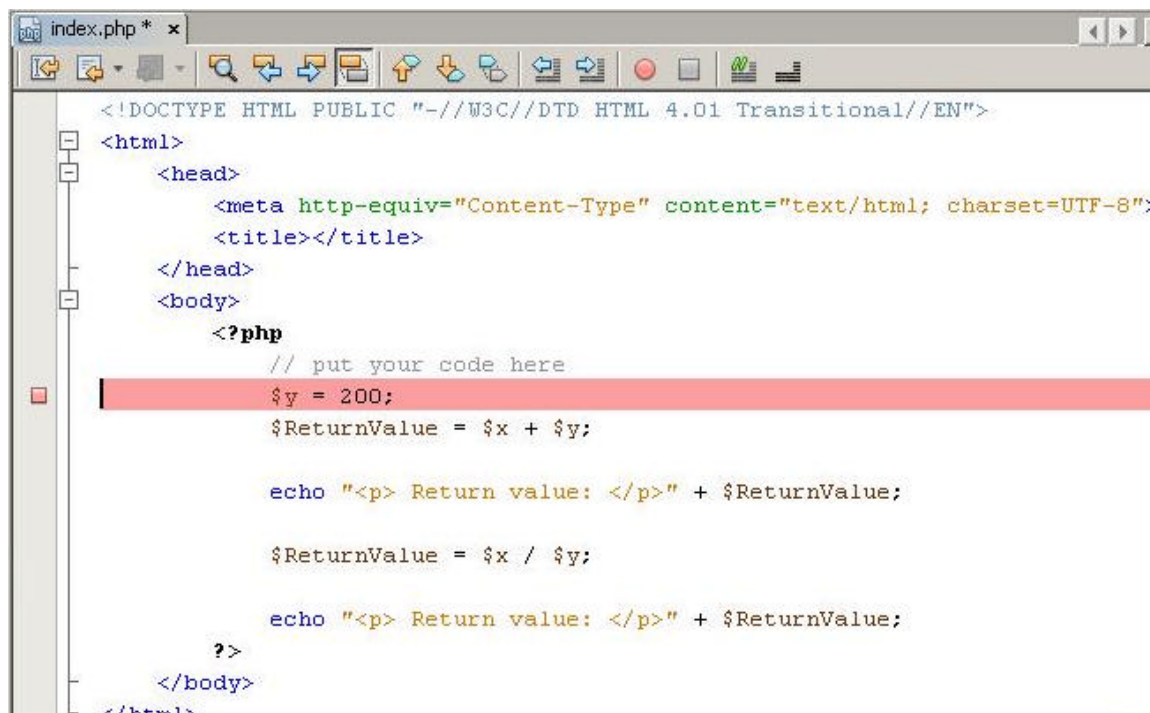
    <title></title>
  </head>
  <body>
    <?php
      // put your code here
      $x = 100;
      $y = 200;
      $ReturnValue = $x + $y;
      echo "<p> Return value: </p>" + $ReturnValue;
      $ReturnValue = $x / $y;
      echo "<p> Return value: </p>" + $ReturnValue;
    ?>
```

Setting breakpoints

To start our investigation, we set a breakpoint. A breakpoint is a flag attached to a line of source code that will stop the execution of the code when it is reached. You can set a breakpoint in the Netbeans editor, selecting the appropriate line in our case line 10, and by clicking the right mouse button and then select "Toggle Line Breakpoint".



After you insert a breakpoint you may notice the line background is changed to red.



Once you have set the breakpoint, invoke you can start debugging by pressing.



Bijlage 3: Toelichting leerstof hoofdstuk 1, eerste deel

Het is voor ons docenten (en voor jezelf) handig als je bovenaan elke PHP-pagina een blokje commentaar opneemt waarin onder anderen je naam, het hoofdstuk en het nummer van de opdracht komen te staan.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<?php
/*
 * Filename      :
 * Assignment    :
 * Created       :
 * Description    :
 * Programmer    :
 *
 */
?>

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html;
                                     charset=iso-8859-1" />
    <title></title>
  </head>

  <body>
  </body>
</html>
```

Bijlage 4: Toelichting leerstof hoofdstuk 1, tweede deel

Op blz. 23 wordt iets gezegd over de naamgeving van variabelen. In tegenstelling tot wat in het boek staat kiezen wij voor:

De naam van een variabele begint altijd met een kleine letter (meteen na het \$teken). Als de variabele uit meer woorden bestaat dan beginnen de volgende woorden steeds met een hoofdletter.

Voorbeelden:

```
$teller  
$mijnWoonplaats  
$totaalAantalStudenten
```

In het codefragment op blz. 26 staan dubbele \$ tekens. Dit moet er uiteraard slechts één zijn.

Bij blz. 43:

Het is ook mogelijk om een complete integer-deling te maken (waarbij het resultaat wordt afgekapt op een geheel getal). We gebruiken hiervoor de function *floor()*.

```
$divisionResult = floor(15/6); // levert 2
```

Bijlage 5: Toelichting leerstof hoofdstuk 2

Op blz. 87 en volgend wordt het if-else statement behandeld. De voorbeelden geven misschien de indruk dat alle html-code met behulp van echo moet worden geschreven.

Maar het kan ook anders; we gebruiken dan mixed html en php.

In onderstaand voorbeeld komt een link te staan die alleen voor ingelogde gebruikers is te volgen:

```
<?php
    if($userIngelogd)
    {
?>
        <a href="toonforum.php" title="ga naar forum">forum</a>
<?php
    }
    else
    {
?>
        <p title="alleen voor ingelogde gebruikers">Forum</p>
<?php
    }
?>
```

Op blz. 96 en volgend wordt het while- en het do-while statement behandeld. Bij elk van de voorbeelden gaat het om een zogenaamde tellende lus (die ook heel eenvoudig met een for-statement is te programmeren). Daarom volgt hier nog een voorbeeld met een niet tellende lus:

Stel we hebben een array `$woorden[]`; dit array bevat een groot aantal woorden. Een variabele `$woord` bevat een enkel woord. We moeten uitzoeken of de inhoud van `$woord` voorkomt in het array `$woorden`. Indien dit het geval is moet de index beschikbaar komen, zo niet dan de waarde -1.

```
function zoekWoord($woord, $woorden)
{
    $gevonden = $false;
    $i = 0;

    while($i < count($woorden) && !gevonden)
    {
        if($woord == $woorden[$i])
        {
            $gevonden = true;
        }
        else
        {
            $i++;
        }
    }

    if($gevonden)
    {
        return $i;
    }
}
```

```
    else
    {
        return -1;
    }
}
```


Bijlage 6: Toelichting leerstof hoofdstuk 4

In hoofdstuk 4 wordt een voorbeeld van een form gegeven. De verwerking van dat formulier vindt in een andere pagina plaats. Het kan ook anders (is meer in gebruik): het formulier en de verwerking daarvan in dezelfde pagina. Schematisch ziet het er als volgt uit:

```
if(!op de submit knop gedrukt)
{
    // eerste keer dat we deze pagina binnenkomen

    plaats formulier op het scherm
}
else
{
    // tweede keer dat we deze pagina binnenkomen,
    // dus formulier is al ingevuld

    verwerk formulier
}
```

We geven een volledig uitgewerkt voorbeeld: *calculateAge.php* (deze file is te vinden in Blackboard). De regelnummers staat normaal niet in een php-pagina; ze zijn hier toegevoegd om een ander te kunnen uitleggen.

In deze pagina wordt een geboortedatum ingetoetst. Met behulp van deze datum en huidige datum wordt de leeftijd van de betreffende persoon berekend (en op het scherm gezet). De pagina is ingedeeld volgens het hierboven gegeven schema. Het formulier en de verwerking vinden plaats binnen dezelfde pagina.

In regel 20 wordt afgetest of `$_POST["submit"]` bestaat. Dat is de eerste keer niet het geval (verderop meer hierover). De regels 21 t/m, 39 worden daarom uitgevoerd en de regels 41 t/m 61 niet.

In regel 25 wordt aangegeven dat de variabelen worden verzonden via de `$_POST` array (en niet zoals in het voorbeeld van het boek via de `$_GET`-array (in de url)). Verder is in regel 25 aangegeven dat vanuit deze pagina *calculateAge.php* nogmaals moet worden aangeroepen.

Als het textfield is ingevuld en op de knop is geklikt dan wordt de waarde die is ingevoerd in het textfield dus verzonden als `$_POST["dateOfBirth"]`. Maar ook de waarde van de knop wordt verzonden: `$_POST["submit"]`.

Als we de pagina voor de tweede keer binnenkomen wordt in regel 20 weer afgetast of `$_POST["submit"]` bestaat. Dat is dus nu het geval. Nu worden de regels 41 t/m 61 uitgevoerd. Hier wordt de leeftijd berekend en naar het scherm geschreven.

```
01 <?php
02 /*
03  * Filename      : calculateAge.php
04  * Project       : Introduction course php programming
05  * Created       : 7-7-2009
07  * Description   : Calculate age with date of birth and today's date
08  * Programmer    : Jeroen Pijpker
```

```

09  */
10  ?>
11
12  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
13  <html>
14  <head>
15      <meta http-equiv="Content-Type" content="text/html;
                                     charset=UTF-8">
16      <title>Calculate Age</title>
17  </head>
18  <body>
19      <?php
20          if(!isset($_POST["submit"]))
21          {
22              // we still have to fill in the form
23          ?>
24
25          <form method="post" action="calculateAge.php">
26              <table>
27                  <tr>
28                      <td>Date of birth (ddmmyyyy) :</td>
29                      <td><input type="text"
                                     name="dateOfBirth"></td>
30                  </tr>
31                  <tr>
32                      <td>&nbsp;</td>
33                      <td><input type="submit"
                                     name="submit"
                                     value="Calculate"></td>
34                  </tr>
35              </table>
36          </form>
37
38          <?php
39          }
40          else
41          {
42              $dateOfBirth      = $_POST["dateOfBirth"];
43              $dayOfBirth       = floor($dateOfBirth /
1000000);
44              $temp              = $dateOfBirth % 1000000;
45              $monthOfBirth     = floor($temp/10000);
46              $yearOfBirth      = $temp % 10000;
47              $dayNow           = Date("j");
48              $monthNow         = Date("n");
49              $yearNow          = Date("Y");
50
51              if(($monthNow < $monthOfBirth) || ($monthNow ==
                                     $monthOfBirth && $dayNow < $dayOfBirth))
52              {
53                  $age = $yearNow - $yearOfBirth - 1;
54              }
55              else
56              {
57                  $age = $yearNow - $yearOfBirth;
58              }
59
60              echo "<p>The age is: $age</p>";

```

```
61         }  
62     ?>  
63 </body>  
64 </html>
```

Bijlage 7: Activiteiten hoofdstuk 1

B7.1 Activiteit 1

Je moet een Web pagina maken (dat gebruik maakt van `print()` statements) welke informatie toont van Stenden.

1. Maak een nieuw PHP project aan in Netbeans;
2. Voeg een standaard PHP script delimiters toe aan document body. Voeg een commentaar blok toe aan de code section wat de datum van vandaag en jouw naam bevat;
3. Gebruik `print()` statements om een tekst af te drukken zoals is te zien in de figuur hieronder. Wees er zeker van dat dat het formaat van de tekst gelijk is



B7.2 Activiteit 2

Maak een nieuw PHP project aan door middel van Netbeans en schrijf een PHP code blok dat gebruik maakt van `echo()` statements om jouw naam, adres en geboortedatum af te drukken op het scherm.

Voeg een commentaar blok toe aan de code section wat de datum van vandaag en jouw naam bevat;

Beschrijf alles zodanig dat het conform strict DTD is.

B7.3 Activiteit 3

Maak een nieuw PHP project aan door middel van Netbeans.

In dit project moet je gebruik maken van de code welke je hieronder kunt vinden.

Deze code bevat een aantal fouten welke jij moet opsporen en oplossen.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title></title>
  </head>
  <body>
    <?php
      /*
      Your name
      Today's date
      */
      echo <h1>California Occuptional Guide</h2>";
      echo ("<h2>Locksmiths</h2>
      echo "<p>A locksmith installs, services, and repairs various
      types of locks and ... ."
    >
  </body>
```

B7.4 Activiteit 4

Maak een nieuw PHP Project in Netbeans om graden Celsius om te rekenen naar graden Fahrenheit. De formule hiervoor is: $^{\circ}\text{F} = \frac{9}{5} * ^{\circ}\text{C} + 32$.

Gebruik hiervoor de onderstaande code:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title></title>
  </head>
  <body>
    <?php
      $tempCelsius = 20;
      echo "<p>Temperatuur in Graden celsius $tempCelsius °C. </p>";
      $tempFahrenheit = ((9 / 5) * $tempCelsius) + 32;
      echo "<p>Temperatuur in Fahrenheit $tempFahrenheit °F. </p>";
    ?>
  </body>
</html>
```

B7.5 Activiteit 5

Maak een nieuw PHP Project in Netbeans om Fahrenheit om te rekenen naar graden Celsius. De formule hier is: $^{\circ}\text{C} = (^{\circ}\text{F} - 32) * \frac{5}{9}$

B7.6 Activiteit 6

Zet een bedrag in Euro's en centen om naar centen.

B7.7 Activiteit 7

Zet een bedrag in centen om naar Euro's en centen.

B7.8 Activiteit 8

De int variabele datum bevat een datum in de vorm *ddmmjjjj*.

Voorbeeld: de datum 14 april 2010 wordt opgeslagen als *14042010*.

Maak een nieuw PHP project in Netbeans waarmee de int-variabelen *dag*, *maand* en *jaar* met de juiste gegevens uit datum worden gevuld. (Maak gebruik van % en /)

B7.9 Activiteit 9

Zoek op de site <http://nl.php.net> uit wat de functie `time()` doet.

Bijlage 8: Activiteiten hoofdstuk 2

B8.1 Activiteit 1

Schrijf een PHP programma waarmee de som van de getallen 1 t/m 100 wordt berekend en getoond.

B8.2 Activiteit 2

Er moeten programmafragmenten gemaakt worden voor een cijferadministratie die aan een cijfer een tekst koppelt.

Er kunnen alleen hele cijfers (int-waarden) ingevoerd worden.

De volgende koppelingen moeten er zijn:

Cijfer	Uitvoer
< 1	"Ongeldig cijfer".
1, 2, 3	"Zeer slecht".
4, 5	"Onvoldoende".
6, 7	"Voldoende".
8	"Goed".
9	"Zeer goed".
10	"Uitmuntend".
>10	"Ongeldig cijfer".

Maak hiervoor 2 programmafragmenten, één met een if (-else) en één met een case structuur. Zorg er voor dat beide zo efficiënt mogelijk werkt.

B8.3 Activiteit 3

Schrijf 3 verschillende functies waarmee het volgende patroon naar het scherm wordt geschreven. Je moet gebruik maken van een for-lus, while en do-while lus.

```
*****
```

B8.4 Activiteit 4

Schrijf 3 verschillende functies waarmee het volgende patroon naar het scherm wordt geschreven. Je moet gebruik maken van een geneste for-lus, while en do-while lus.

```
*****
*****
*****
*****
*****
```

B8.5 Activiteit 5

Schrijf 3 verschillende functies waarmee het volgende patroon naar het scherm wordt geschreven. Je moet gebruik maken van een geneste for-lus, while en do-while lus.

```
*
**
***
****
*****
*****
*****
*****
*****
*****
*****
```

B8.6 Activiteit 6

Schrijf 3 verschillende functies waarmee het volgende patroon naar het scherm wordt geschreven. Je moet gebruik maken van een geneste for-lus, while en do-while lus.

```
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
```

B8.7 Activiteit 7

De docent wijst van onderstaande opdrachten één opdracht toe aan elke student.

Schrijf 3 verschillende functies waarmee het volgende patroon naar het scherm wordt geschreven. Je moet gebruik maken van een geneste for-lus, while en do-while lus.

Opdracht 1

```
      *
     * *
    * * *
   * * * *
  * * * * *
 * * * * * *
* * * * * * *
* * * * * * *
* * * * * * *
```

Opdracht 2

```
* * * * * * * *
 * * * * * * *
  * * * * * * *
   * * * * * * *
    * * * * * * *
     * * * * * *
      * * * * *
       * * * *
        * * *
         * *
          *
```

Opdracht 3

```
* * * * * * * *
 * * * * * * *
  * * * * * *
   * * * * *
    * * * *
     * * *
      * *
       *
```

Opdracht 4

```
*
 * * *
* * * *
* * * * *
```


Opdracht 13

-	+	+	+	+	+	+	+
+	-	+	+	+	+	+	+
+	+	-	+	+	+	+	+
+	+	+	-	+	+	+	+
+	+	+	+	-	+	+	+
+	+	+	+	+	-	+	+
+	+	+	+	+	+	-	+
+	+	+	+	+	+	+	-

Opdracht 14

-	-	-	-	-	-	-	-
+	-	-	-	-	-	-	-
+	+	-	-	-	-	-	-
+	+	+	-	-	-	-	-
+	+	+	+	-	-	-	-
+	+	+	+	+	-	-	-
+	+	+	+	+	+	-	-
+	+	+	+	+	+	+	-
+	+	+	+	+	+	+	+

Opdracht 15

[illegible]

Opdracht 16

[illegible]

Opdracht 17

-	-	-	-	-	-	+
-	-	-	-	-	+	+
-	-	-	-	+	+	+
-	-	-	-	+	+	+
-	-	-	+	+	+	+
-	-	+	+	+	+	+
-	+	+	+	+	+	+
+	+	+	+	+	+	+
-	+	+	+	+	+	+
-	-	+	+	+	+	+
-	-	-	+	+	+	+
-	-	-	-	+	+	+
-	-	-	-	-	+	+
-	-	-	-	-	-	+
-	-	-	-	-	-	-

Opdracht 18

+	-	-	-	-	-	-
+	+	-	-	-	-	-
+	+	+	-	-	-	-
+	+	+	+	-	-	-
+	+	+	+	+	-	-
+	+	+	+	+	+	-
+	+	+	+	+	+	+
+	+	+	+	+	+	-
+	+	+	+	+	+	-
+	+	+	+	+	-	-
+	+	+	+	+	-	-
+	+	+	-	-	-	-
+	+	-	-	-	-	-
+	-	-	-	-	-	-

Opdracht 19

```

1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6

```

Opdracht 20

```

1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1

```

Opdracht 21

$$\begin{array}{cccc} & & & 1 \\ & & 2 & 1 \\ & 3 & 2 & 1 \\ 4 & 3 & 2 & 1 \end{array}$$

Opdracht 22

```

1 2 3 4 5 6
  1 2 3 4 5
    1 2 3 4
      1 2 3

```

5 4 3 2 1
6 5 4 3 2 1

1 2
1

Opdracht 23

```
1 2 3 4 5 6
  2 3 4 5 6
    3 4 5 6
      4 5 6
        5 6
          6
```

Opdracht 24

```
0 1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9 0
2 3 4 5 6 7 8 9 0 1
3 4 5 6 7 8 9 0 1 2
4 5 6 7 8 9 0 1 2 3
5 6 7 8 9 0 1 2 3 4
6 7 8 9 0 1 2 3 4 5
7 8 9 0 1 2 3 4 5 6
8 9 0 1 2 3 4 5 6 7
9 0 1 2 3 4 5 6 7 8
```

B8.8 Activiteit 8

Gegeven is de volgende array `$fruits = ("melon", "banana", "orange", "apple", "apricot")`;

Het is de bedoeling om deze array te printen met een `foreach` lus.

Bijlage 9: Activiteiten hoofdstuk 3

B9.1 Activiteit 1

Ontwikkel een PHP pagina waarmee in een textarea een stuk tekst (woorden, gescheiden door spaties) kan worden ingevoerd.

Als resultaat worden alle woorden uit de tekst getoond (één woord per regel), waarbij in de woorden van tenminste vier letters de eerste en laatste letter zijn verwisseld.

B9.2 Activiteit 2

Ontwikkel een PHP pagina waarmee in een textarea een stuk tekst (woorden, gescheiden door spaties) kan worden ingevoerd.

Als resultaat worden alle woorden uit de tekst getoond (één woord per regel), waarbij in elk woord alle hoofdletters zijn omgezet naar kleine letters, behalve als alleen de eerste letter van het betreffende woord een hoofdletter is.

Voorbeeld: Herman Archer LIVEs in nEW YORK wordt dan Herman Archer lives in new york.

Bijlage 10: Individuele eindopdracht

Het is de bedoeling om een website met PHP te ontwikkelen. Deze website gaat over jezelf.

Het uiterlijk van de website moet afhankelijk van de weersomstandigheden in de provincie Drenthe.

Deze website moet een aantal elementen bevatten:

- Home;
- My Location (gebruik maken van een google map);
- My Weather;
- My Interests;
- Info Form.

Hoe gaan we dit oplossen:

Er is op Blackboard (in de course Bouwsteen Inleiding in programmeren in PHP) een ZIP file te vinden met de naam **setGoogleMapLocation**. In deze ZIP file is de benodigde PHP code te vinden voor methode 1. Voor methode 2 is op dezelfde locatie een file te vinden met de naam **getWeatherTempHoogeveen.php**.

Methode 1:

setGoogleMapLocation(\$street, \$place, \$windowText) – deze methode kan worden gebruikt om een google maps image op te nemen in je website.

Methode 2:

getWeatherStringHoogeveen() – deze methode returned een string met informatie over het weer in de regio Drenthe.

De return string bevat 4 elementen:

- Plaats;
- Temperatuur;
- Conditie nummer;
- Conditie;

Betekenis conditienummer:

Nr	Conditie	Nr	Conditie
0	tornado	22	smoky
1	tropical storm	23	blustery
2	hurricane	24	windy
3	severe thunderstorms	25	cold
4	thunderstorms	26	cloudy
5	mixed rain and snow	27	mostly cloudy (night)
6	mixed rain and sleet	28	mostly cloudy (day)
7	mixed snow and sleet	29	partly cloudy (night)
8	freezing drizzle	30	partly cloudy (day)
9	drizzle	31	clear (night)
10	freezing rain	32	sunny
11	showers	33	fair (night)
12	showers	34	fair (day)
13	snow flurries	35	mixed rain and hail
14	light snow showers	36	hot
15	blowing snow	37	isolated thunderstorms
16	snow	38	scattered thunderstorms
17	hail	39	scattered thunderstorms
18	sleet	40	scattered showers
19	dust	41	heavy snow
20	foggy	42	scattered snow showers

21	haze		
----	------	--	--

Bijlage 11: Code conventies

1. Naming

1.1 Use meaningful names.

Use descriptive names for all identifiers (names of classes, variables and methods). Avoid ambiguity. Avoid abbreviations.
Simple mutator methods should be named `setSomething(...)`.
Simple accessor methods should be named `getSomething(...)`.
Accessor methods with Boolean return values are often called `isSomething(...)`, for example `isEmpty()`.

1.2 Class names start with a capital letter.

1.3 Class names are singular nouns.

1.4 Methods and variable names start with lowercase letters.

All the – class, method and variable names – use capital letters in the middle to increase readability of compound identifiers, e.g. `numberOfItems`.

1.5 Constants are written in UPPERCASE.

Constants occasionally use underscores to indicate compound identifiers:
`MAXIMUM_SIZE`

|

2. Layout

2.1 One level of indentation is four spaces.

2.2 All statements within a block are intended one level.

2.3 Braces for classes and methods are alone on one line.

The braces for class and method blocks are on separate lines and are at the same indentation level, for example:

```
public int getAge()  
{  
    statements  
}
```

2.4 Also for all other blocks, are alone on one line.

The braces for all other blocks are on separate lines and are at the same indentation level, for example:

```
while(condition)  
{  
    statements  
}
```

```
if(condition)  
{  
    statements  
}  
else  
{  
    statements  
}
```

2.5 Always use braces in control structures

Braces are used in if-statements and loops even if the body is only a single statement.

2.6 Use a space before the opening brace of a control structure's block.

2.7 Use a space around operators.

2.8 Use a blank line between methods (and constructors).

Use blank lines to separate logical blocks of code. This means at least between methods, but also between logical parts within a method.

Bijlage 12: Scoring rubrics

De toets heeft een aantal gebieden waarop hij beoordeeld zal worden namelijk: Correctheid v.d. code, specificaties, efficiëntie, Code conventies, leesbaarheid en documentatie.

Daarnaast krijgt de student als bonus 10 punten zodat het maximum aantal te behalen punten voor de toets 100 is. Om het eindcijfer te berekenen dient dit puntenaantal door 10 gedeeld te worden.

De verdeling van bovengenoemde punten zal verdeeld worden over de opgaven. Het gewicht dat een opgave meetelt wordt aangegeven op de toets.

De bonuspunten zullen per opgave worden toegekend.

Eindbeoordeling toets van student:			met studentnummer:		
Onderdeel	Punten opgave 1	Punten opgave 2 (indien van toepassing)	Punten opgave 3 (indien van toepassing)	Punten opgave 4 (indien van toepassing)	Totaal aantal punten onderdeel
Totaal (totaal maximaal 90)					A
Bonuspunten (totaal 10)					B
				Totaal A+B	
				Delen door 10	
				Eindcijfer	

Voor het maken van de eindbeoordeling zal gebruik worden gemaakt van de beoordelingsformulieren welke op de volgende pagina zijn te vinden.

Beoordelingsformulieren van Student:

studentnummer:

Beoordelingsformulier opgave:			
Onderdeel	Opmerkingen	Maximum aantal punten	Behaalde punten
Algemeen			
Bonus			

Beoordelingsformulier opgave:			
Onderdeel	Opmerkingen	Maximum aantal punten	Behaalde punten
Algemeen			
Bonus			

Bijlage 13: Oefentoets**Naam Student:****Klas:****SECTOR TECHNIEK
GEGEVENSBLAD**

CODERING/VAK	:	IPPHP
KLAS	:	I-1
DATUM	:	week 44
TIJD	:	180 minuten
(DEEL)TENTAMEN/HERKANSING	:	(DEEL)TENTAMEN
DOCENT(EN)	:	B. Meijerink
AANTAL OPGAVEN	:	3
BIJLAGEN	:	3
TOEGESTANE HULPMIDDELEN	:	-
TOEGESTANE LITERATUUR	:	Alles
OPGAVEN INLEVEREN	:	ja

Attentie:

De toets is een praktijktoets waarvan de software aan het eind van de toets ingeleverd moet worden.

Je naam en studentnummer moet in het commentaar bovenaan elke programma komen te staan. **Ontbreken naam en/of studentnummer op één of meerdere uitwerkingen dan resulteert dat in het cijfer 1.**

NORMERING:

Normering zal plaatsvinden aan de hand van de scoring rubrics zoals aangegeven in het bouwstenenboek IPPHP. Van de 100 te behalen punten kun je 38 punten halen met opgave 1, 38 punten halen met opgave 2 en 24 punten halen met opgave 3.

Opgave 1

In bijlage 1 is de gedeeltelijk programmacode te vinden welke bij opgave 1 hoort. (Let op! Deze bijlage 1 is een onderdeel van deze voorbeeld toets en de hier bedoelde bijlage 1 is in bijlage 13 van dit bouwsteenboek te vinden.)

In deze programmacode is een string opgenomen met een datum om je programma te testen. **Jouw programma moet zodanig gemaakt worden dat het werkt voor elke datum (Invoerformat DD-MM-JJJJ).**

Jij moet het programma afmaken zodat de uitvoer wordt gegenereerd welke te zien is in afbeelding 1. Deze uitvoer is gegenereerd aan de hand van de gegeven string maar zal natuurlijk anders zijn als je het met een andere string uitvoert.

De uitvoer bevat de volgende elementen:

- Textelement: De persoon is geboren op:
- Datumelement: 23 Juli 2010 (Let op: maand is nu in tekst weergegeven);
- Periode-element: in de zomerperiode. De maanden maart, april en mei vormen de voorjaarsperiode, de maanden juni, juli en augustus vormen de zomerperiode, de maanden september, oktober en november vormen de herfstperiode en de maanden december, januari en februari vormen de winter periode.

Schrijf de resterende code om bovenstaande uit te kunnen voeren. **Er moet minimaal 1 switch structuur en minimaal 1 if- else structuur in zitten.**

(vergeet niet je naam en studentnummer in de code op te nemen.)



Afbeelding 1

Opgave 2

In bijlage 2 is de gedeeltelijk programmacode te vinden welke bij opgave 2 hoort. (Let op! Deze bijlage 2 is een onderdeel van deze voorbeeld toets en de hier bedoelde bijlage 1 is in bijlage 13 van dit bouwsteenboek te vinden.)


Bij deze opgave moet je een formulier (afbeelding 2) maken welke de wegenbelasting berekend voor een voertuig afhankelijk van het type voertuig. (Er wordt alleen naar gewicht en type voertuig gekeken. Voor een motor is de te betalen belasting 0,60 Euro maal het gewicht, voor een personenauto is de te betalen belasting 0,55 Euro maal het gewicht en voor een vrachtwagen is de te betalen belasting 0,50 Euro maal het gewicht.

Wordt dit formulier ingevuld zoals te zien op afbeelding 3, dan gedrukt op de knop **Bereken belasting**, dan moet het scherm er uit komen te zien zoals te zien op afbeelding 4. (Uitvoer bij normaal gebruik.)

Wordt dit formulier ingevuld zoals te zien op afbeelding 5 (programma is opnieuw opgestart), dan gedrukt op de knop **Bereken belasting**, dan moet het scherm er uit komen te zien zoals te zien op afbeelding 6. (Je hebt geen correct gewicht ingevoerd.)

Wordt dit formulier ingevuld zoals te zien op afbeelding 7 (programma is opnieuw opgestart), dan gedrukt op de knop **Bereken belasting**, dan moet het scherm er uit komen te zien zoals te zien op afbeelding 8. (Je bent het voertuigtype vergeten in te voeren.)

Schrijf de resterende code om bovenstaande uit te kunnen voeren. (**vergeet niet je naam en studentnummer in de code op te nemen.**)



The screenshot shows a Mozilla Firefox browser window with the title 'Belasting berekening - Mozilla Firefox'. The address bar shows 'http://localhost/Oefentoets0-2/index.p'. The page content is titled 'Bepalen Wegenbelasting'. Below the title, there is a text input field with the placeholder text 'Geef het gewicht van het voertuig op in kilogrammen.' Below this field are three radio buttons labeled 'Motor', 'Personenwagen', and 'Vrachtwagen'. At the bottom of the form is a button labeled 'Bereken Belasting'. The status bar at the bottom of the browser window shows 'Klaar'.

Afbeelding 2



Belasting berekening - Mozilla Firefox

Bestand Bewerken Beeld Geschiedenis Bladwijzers Extra Help

http://localhost/Oefentoets0-2/index.p

Meest bezocht Aan de slag Laatste nieuws

Bepalen Wegenbelasting

Geef het gewicht van het voertuig op in kilogrammen.

1000

☐ Motor

☒ Personenwagen

☐ Vrachtwagen

Bereken Belasting

Klaar

Afbeelding 3



Belasting berekening - Mozilla Firefox

Bestand Bewerken Beeld Geschiedenis Bladwijzers Extra Help

http://localhost/Oefentoets0-2/index.p

Meest bezocht Aan de slag Laatste nieuws

Bepalen Wegenbelasting

Te betalen wegenbelasting is 550 Euro.

☐ Motor

☐ Personenwagen

☐ Vrachtwagen

Bereken Belasting

Klaar

Afbeelding 4, uitvoer van afbeelding 3



Belasting berekening - Mozilla Firefox

Bestand Bewerken Beeld Geschiedenis Bladwijzers Extra Help

http://localhost/Oefentoets0-2/index.p

Meest bezocht Aan de slag Laatste nieuws

Belasting berekening

Bepalen Wegenbelasting

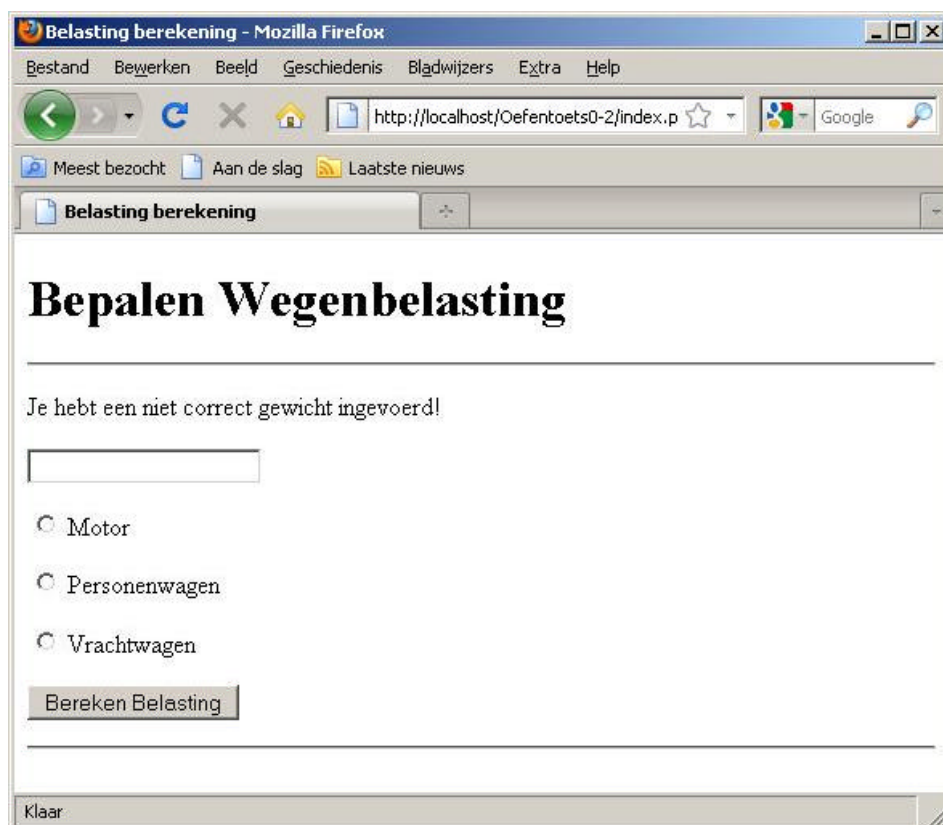
Geef het gewicht van het voertuig op in kilogrammen.

☐ Motor
☐ Personenwagen
☒ Vrachtwagen

Bereken Belasting

Klaar

Afbeelding 5, geen gewicht ingevoerd



Belasting berekening - Mozilla Firefox

Bestand Bewerken Beeld Geschiedenis Bladwijzers Extra Help

http://localhost/Oefentoets0-2/index.p

Meest bezocht Aan de slag Laatste nieuws

Belasting berekening

Bepalen Wegenbelasting

Je hebt een niet correct gewicht ingevoerd!

☐ Motor
☐ Personenwagen
☐ Vrachtwagen

Bereken Belasting

Klaar

Afbeelding 6, uitvoer van afbeelding 5



Belasting berekening - Mozilla Firefox

Bestand Bewerken Beeld Geschiedenis Bladwijzers Extra Help

http://localhost/Oefentoets0-2/index.p

Meest bezocht Aan de slag Laatste nieuws

Belasting berekening

Bepalen Wegenbelasting

Geef het gewicht van het voertuig op in kilogrammen.

2000

☐ Motor

☐ Personenwagen

☐ Vrachtwagen

Bereken Belasting

Klaar

Afbeelding 7, geen voertuigtype aangegeven



Belasting berekening - Mozilla Firefox

Bestand Bewerken Beeld Geschiedenis Bladwijzers Extra Help

http://localhost/Oefentoets0-2/index.p

Meest bezocht Aan de slag Laatste nieuws

Belasting berekening

Bepalen Wegenbelasting

Je bent het voertuigtype vergeten in te voeren.

☐ Motor

☐ Personenwagen

☐ Vrachtwagen

Bereken Belasting

Klaar

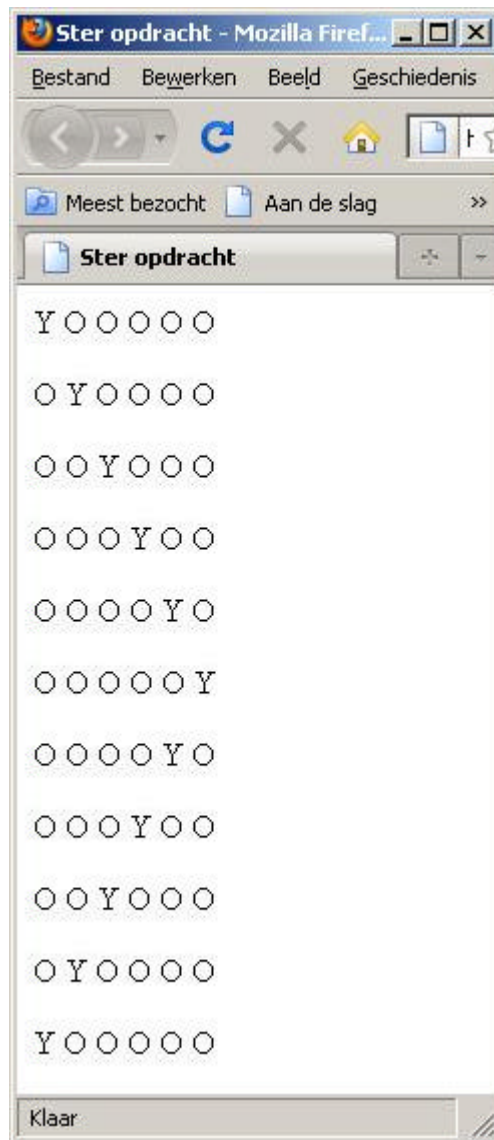
Afbeelding 8, uitvoer van afbeelding 7

Opgave 3

In bijlage 3 is de gedeeltelijk programmacode te vinden welke bij opgave 3 hoort. (Let op! Deze bijlage 3 is een onderdeel van deze voorbeeld toets en de hier bedoelde bijlage 1 is in bijlage 13 van dit bouwsteenboek te vinden.)

Bij deze opgave moet je door middel van een geneste lus structuur (je mag kiezen uit for, while of do/while) een afdruk maken zoals is te zien in afbeelding 9.

Schrijf de resterende code om bovenstaande uit te kunnen voeren. **(vergeet niet je naam en studentnummer in de code op te nemen.)**



Afbeelding 9

Bijlage 1

```
<!--  
    Naam:  
    Studentnummer:  
-->  
  
<!DOCTYPE html PUBLIC  
    "-//W3C//DTD XHTML 1.0 Strict//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
    <head>  
        <title>Stringbewerking</title>  
        <meta http-equiv="content-type"  
            content="text/html; charset=iso-8859-1" />  
    </head>  
    <body>  
        <?php  
            $invoerString = "23-07-2010"; // dagen < als 10 moet 0  
                                           // voor staan. Voorbeeld: 01.  
  
            /*  
            *  
            *      Door student in te vullen.  
            *  
            */  
  
        ?>  
    </body>  
</html>
```

Bijlage 2

```
<!--
  Naam:
  Studentnummer:
-->

<!DOCTYPE html PUBLIC
  "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Belasting berekening</title>
    <meta http-equiv="content-type"
      content="text/html; charset=iso-8859-1" />
  </head>
  <body>
    <h1>Bepalen Wegenbelasting</h1><hr />

    /*
     *
     *      Door student in te vullen.
     *
    */

  </body>
</html>
```

Bijlage 3

```
<!--
<!--
    Naam:
    Studentnummer:
-->

<!DOCTYPE html PUBLIC
    "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>Ster opdracht</title>
        <meta http-equiv="content-type"
            content="text/html; charset=iso-8859-1" />
    </head>
    <body>
        <?php

            /*
             *
             *      Door student in te vullen.
             *
             */

        ?>
    </body>
</html>
```