

## Scriptingtalen (C002178)

Wegens Covid19 kan mogelijk afgeweken worden van de onderwijs- en evaluatievormen. Dergelijke afwijkingen zullen via Ufora worden gecommuniceerd.

**Cursusomvang** *(nominale waarden; effectieve waarden kunnen verschillen per opleiding)*

**Studiepunten 6.0**                      **Studietijd 180 u**                      **Contacturen**                      60.0 u

**Aanbodsessies en werkvormen in academiejaar 2021-2022**

A (semester 2)	Nederlands	Gent	werkcollege: PC- klasoefeningen	30.0 u
			hoorcollege	30.0 u

**Lesgevers in academiejaar 2021-2022**

Dawyndt, Peter	WE02	Verantwoordelijk lesgever
----------------	------	---------------------------

**Aangeboden in onderstaande opleidingen in 2021-2022**

	stptn	aanbodssessie
<a href="#">Bachelor of Science in de informatica</a>	6	A

**Onderwijstalen**

Nederlands

**Trefwoorden**

python, javascript, scriptingtalen, dynamische typering

**Situering**

Scriptingtalen zijn geschikt voor scripts: computerprogramma's geschreven om in een specifieke omgeving het uitvoeren van taken te automatiseren, daar waar die anders één voor één door een mens zouden moeten uitgevoerd worden. Scriptingtalen worden vaak geïnterpreteerd in plaats van gecompileerd. Hun primitieven zijn doorgaans elementaire taken of API calls, en de taal laat toe om deze primitieven te combineren tot complexere programma's. Omgevingen die door middel van scripting kunnen geautomatiseerd worden zijn onder meer softwaretoepassingen, webpagina's die door een browser weergegeven worden, shells van besturingssystemen, ingebedde systemen en games. Scriptingtalen kunnen daardoor ook gezien worden als domeinspecifieke talen voor een bepaalde omgeving, en worden in het specifieke geval van het scripten van een softwaretoepassing ook vaak uitbreidingstalen genoemd.

Het spectrum van scriptingtalen strekt zich uit van zeer kleine en doorgedreven domeinspecifieke talen tot hoogniveau programmeertalen voor algemeen gebruik die ook voor scripting kunnen gebruikt worden. Standaardvoorbeelden van scriptingtalen voor specifieke omgevingen zijn onder andere Bash (voor Unix en Unix-achtige besturingssystemen), ECMAScript/Javascript (voor web browsers) en Visual Basic for Applications (voor Microsoft Office toepassingen). Python is een programmeertaal voor algemeen gebruik die ook vaak als uitbreidingstaal gebruikt wordt, terwijl ECMAScript/Javascript nog steeds hoofdzakelijk als scriptingtaal voor web browsers gebruikt wordt, daar waar ze ook steeds vaker als programmeertalen voor algemeen gebruik wordt ingezet.

**Inhoud**

Je leert de mogelijkheden van scriptingtalen verkennen aan de hand van de scriptingtalen Python en JavaScript (ECMAScript). Deze twee scriptingtalen maken van gebruik van dynamische typering, in tegenstelling tot bijvoorbeeld de programmeertaal Java die gebruik maakt van statische typering. Bovendien ervaar je ook hoe Python en JavaScript binnen eenzelfde programmeertaal ideeën uit het objectgericht programmeren combineren met ideeën uit het functioneel programmeren en programma's die inspelen op gebeurtenissen. Naast het werken met de programmeertalen zelf, leer je ze ook toe te passen voor het automatiseren van

taken in specifieke omgevingen.

### **Begincompetenties**

Aantoonbare programmeerervaring in ten minste één procedurele of objectgerichte programmeertaal (bijvoorbeeld C, C++ of Java).

### **Eindcompetenties**

- 1 Beheersen van de scriptingtaal Python.
- 2 Toepassen van de scriptingtaal Python voor het automatiseren van taken in een specifieke omgeving.
- 3 Beheersen van de scriptingtaal JavaScript (ECMAScript).
- 4 Toepassen van de scriptingtaal JavaScript (ECMAScript) voor het automatiseren van taken in een browser.
- 5 Implementeren van een client-server toepassing die gebruik maakt van Python aan de serverzijde (CGI) en JavaScript aan de clientzijde (browser).

### **Creditcontractvoorwaarde**

Toelating tot dit opleidingsonderdeel via creditcontract is mogelijk mits gunstige beoordeling van de competenties

### **Examencontractvoorwaarde**

Dit opleidingsonderdeel kan niet via examencontract gevolgd worden

### **Didactische werkvormen**

Hoorcollege, zelfstandig werk, werkcollege: PC-klasoefeningen

### **Toelichtingen bij de didactische werkvormen**

Elektronische leeromgeving Ufora (Ufora.ugent.be) wordt gebruikt om eigen inbreng van studenten aan te moedigen en voor het verspreiden van bijkomende achtergrondinformatie en interessante verwijzingen. Opdrachten voor werkcolleges en zelfstandig werk worden verspreid via GitHub (github.ugent.be), kunnen uitgevoerd worden op de interactieve Linux omgeving Helios (helios.ugent.be) en moeten ingediend worden via Indianio (indiano.ugent.be). Voor de programmeeropdrachten kan onmiddellijk geautomatiseerde feedback verkregen worden via de elektronische leeromgevingen Pythia (pythia.ugent.be; Python) en Dodona (dodona.ugent.be; JavaScript).

### **Leermateriaal**

Dit opleidingsonderdeel maakt enkel gebruik van handleidingen die vrij beschikbaar zijn online, aangevuld met videohandleidingen en tips & tricks voor specifieke programmeeropdrachten.

Miller B, Ranim D. How to Think Like a Computer Scientist (gratis beschikbaar online).

Crockford D. Crockford on JavaScript (lessenreeks gratis beschikbaar online).

MDN JavaScript guide (gratis beschikbaar online).

Bootstrap tutorial (gratis beschikbaar online).

Geraamde totaalprijs: €0.

### **Referenties**

Crockford D (2008). JavaScript: The Good Parts. O'Reilly Media. ISBN 978-0-596-51774-8.

Flanagan D (2010). jQuery Pocket Reference. O'Reilly Media. ISBN 978-1-4493-9722-7.

### **Vakinhoudelijke studiebegeleiding**

Via hoor- en werkcolleges aan de computer krijgt de student inzicht in het gebruik van diverse scriptingtalen en wordt hij/zij aangezet om deze talen in de praktijk te brengen door het oplossen van opgelegde automatiseringsproblemen. Tijdens de werkcolleges worden een aantal opgelegde oefeningen en taken beoordeeld. Bijkomende individuele uitleg kan gevraagd worden na afspraak met de verantwoordelijke lesgever of één van de assistenten. Interactieve begeleiding (tussen studenten onderling, en tussen de student en de lesgever) wordt gestimuleerd door gebruik te maken van de elektronische leeromgeving Ufora. Voor de programmeeropdrachten kan onmiddellijk geautomatiseerde feedback verkregen worden via de elektronische leeromgevingen Pythia (pythia.ugent.be; Python) en Dodona (dodona.ugent.be; JavaScript).

### **Evaluatiemomenten**

periodegebonden en niet-periodegebonden evaluatie

#### **Evaluatievormen bij periodegebonden evaluatie in de eerste examenperiode**

Openboekexamen, vaardigheidstest

#### **Evaluatievormen bij periodegebonden evaluatie in de tweede examenperiode**

Openboekexamen, vaardigheidstest

#### **Evaluatievormen bij niet-periodegebonden evaluatie**

Werkstuk, vaardigheidstest

#### **Tweede examenkans in geval van niet-periodegebonden evaluatie**

Examen in de tweede examenperiode is niet mogelijk

#### **Toelichtingen bij de evaluatievormen**

De punten voor het opleidingsonderdeel Scriptingtalen worden voor 35% (7/20) berekend op basis van niet-periodegebonden evaluatie (dagelijks werk) en voor 65% (13/20) op basis van periodegebonden evaluatie (examen). Er zijn in totaal drie niet-periodegebonden evaluatiemomenten die samen de score van de niet-periodegebonden evaluatie bepalen. Voor de **eerste niet-periodegebonden evaluatie (Python)** krijgen de studenten een reeks van 44 Python oefeningen opgelegd. Op basis van de behandelde programmeertechnieken worden deze oefeningen onderverdeeld in 10 reeksen. De eerste oefening van elke reeks is telkens een variant op het werken met ISBN-nummers. Een voorbeeldoplossing van deze oefening is gegeven op Ufora, en in een bijhorende instructievideo wordt uitgelegd hoe we tot deze voorbeeldoplossing gekomen zijn. Met deze ISBN-oefening zetten we uiteen hoe de nieuw aangebrachte programmeertechniek van de oefeningenreeks in de praktijk kan gebracht worden. Met deze voorbereiding kunnen de studenten zelf aan de slag om de programmeertechniek in de praktijk te brengen via het oplossen van de andere opgelegde oefeningen uit de reeks. Studenten moeten hun oplossingen voor de opgelegde oefeningen van elke reeks (inclusief de ISBN-oefening) indienen via het online leerplatform Dodona tegen vooraf vastgelegde deadlines (telkens op vrijdag om 22:00 in de week van het werkcollege gewijd aan de oefeningenreeks). Studenten kunnen gebruik maken van het overzicht op Dodona om op te volgen voor welke oefeningen ze reeds een correcte oplossing hebben ingediend. Dodona geeft immers een handig overzicht van de opgelegde oefeningen uit de opeenvolgende reeksen, de bijhorende deadlines en de tijd waarop voor het eerste een correcte oplossing voor de oefening werd ingediend.

De eerste niet-periodegebonden evaluatie (Python) wordt afgesloten met een evaluatiemoment tijdens het eerste werkcollege dat volgt op de deadline voor het indienen van de oplossingen van de laatste oefeningenreeks. Bij deze evaluaties krijgen de studenten twee uur de tijd om twee nieuwe Python programmeeroefeningen op te lossen. Hierbij kunnen ze gebruik maken van het online leerplatform Dodona om de correctheid van hun oplossingen na te gaan. De ingediende oplossingen van deze evaluatie-oefeningen worden echter met de hand nagekeken door de lesgever en zijn begeleiders, en beoordeeld op correctheid, gebruikte programmeerstijl, keuzes gemaakt bij het gebruik van de verschillende programmeertechnieken en de kwaliteit van de oplossingsmethode. De moeilijkheidsgraad van deze evaluatie-oefeningen is van hetzelfde niveau als deze van de opgaven die tijdens de periodegebonden evaluatie (het examen) worden voorgelegd. Bovendien volgt dit evaluatiemoment dezelfde procedure die gebruikt wordt tijdens de periodegebonden evaluatie, zodat studenten op basis van hun ervaring eventueel hun werkmethode kunnen bijsturen in voorbereiding naar het examen.

Voor de **tweede niet-periodegebonden evaluatie (JavaScript)** krijgen de studenten een reeks van JavaScript oefeningen opgelegd. Met deze oefeningen kunnen de studenten zelf aan de slag om hun kennis en vaardigheden van JavaScript in de praktijk te brengen. Studenten moeten hun oplossingen voor de opgelegde oefeningen van elke reeks indienen via het online leerplatform Dodona tegen vooraf vastgelegde deadlines (telkens op vrijdag om 22:00 in de week van het werkcollege gewijd aan de oefeningenreeks). Studenten kunnen gebruik maken van de online leeromgeving Dodona om de correctheid van hun oplossingen na te gaan en onmiddellijk geautomatiseerde feedback te krijgen op hun ingediende oplossingen.

De tweede niet-periodegebonden evaluatie (JavaScript) wordt afgesloten met een evaluatiemoment tijdens het eerste werkcollege dat volgt op de deadline voor het indienen van de oplossingen van de laatste oefeningenreeks. Bij deze evaluaties krijgen de studenten twee uur de tijd om twee nieuwe JavaScript programmeeroefeningen op te lossen. Hierbij kunnen ze gebruik maken van het online leerplatform Dodona om de correctheid van hun oplossingen te testen. De ingediende oplossingen van deze evaluatie-oefeningen worden echter met de hand nagekeken door de lesgever en zijn begeleiders, en beoordeeld op correctheid, gebruikte

programmeerstijl, keuzes gemaakt bij het gebruik van de verschillende programmeertechnieken en de kwaliteit van de oplossingsmethode. De moeilijkheidsgraad van deze evaluatie-oefeningen is van hetzelfde niveau als deze van de opgaven die tijdens de periodegebonden evaluatie (het examen) worden voorgelegd. Bovendien volgt dit evaluatiemoment dezelfde procedure die gebruikt wordt tijdens de periodegebonden evaluatie, zodat studenten op basis van hun ervaring eventueel hun werkmethode kunnen bijsturen in voorbereiding naar het examen.

De **derde niet-periodegebonden evaluatie** omvat een opdracht waarbij de studenten een client-server toepassing moeten implementeren die gebruik maakt van Python aan de serverzijde (CGI) en JavaScript aan de clientzijde (browser). De oplossing voor deze opdracht moet ingediend worden tegen een vooraf vastgelegde deadline, en wordt beoordeeld op basis van correctheid, keuzes die genomen werden bij het ontwerp en de implementatie, en gebruiksvriendelijkheid van de oplossing.

De score van de niet-periodegebonden evaluaties wordt bepaald volgens de formule  $s * c / a$ . Hierbij staat  $s$  voor de score die de studenten behaald hebben op basis van hun ingediende evaluatie-oefeningen (uitgedrukt op 20),  $c$  staat voor het aantal opgelegde oefeningen die correct werden ingediend tegen de wekelijkse deadlines, en  $a$  staat voor het aantal opgelegde oefeningen. Een student die bijvoorbeeld 16/20 behaald heeft voor zijn evaluatie-oefeningen en alle 44 opgelegde oefeningen correct heeft ingediend voorafgaand aan de wekelijkse deadlines, krijgt voor die evaluatiereeks een score van  $16 * 44/44 = 16$  op 20. Als die student echter nog steeds 16/20 had behaald voor zijn evaluatie-oefeningen, maar slechts 30/44 opgelegde oefeningen correct had ingediend voorafgaand aan de wekelijkse deadlines, dan houdt die student slechts een score van  $16 * 30/44 = 10.9$  op 20 over.

Studenten krijgen zo snel mogelijk na een evaluatie per email hun score toegestuurd. Tijdens het daaropvolgende werkcollege krijgen de studenten ook de oplossingen die ze ingediend hebben tijdens de evaluatie terug, voorzien van feedback die aangeeft waar er verbeterpunten zijn die ze kunnen meenemen bij het verder oplossen van oefeningen of naar het examen. De niet-periodegebonden evaluaties kunnen niet hernomen worden tijdens de tweede examenperiode. Voor de tweede examenperiode wordt enerzijds een score berekend rekening houdend met de punten behaald voor de niet-periodegebonden evaluatie (dus punten dagelijks werk gequoteerd op 7 samen met het examen gequoteerd op 13, zoals dat ook geldt voor de eerste examenperiode) en wordt anderzijds ook een score berekend zonder rekening te houden met de punten behaald voor de nietperiodegebonden evaluatie (examenresultaat rechtstreeks gequoteerd op 20). De eindscore voor de tweede examenperiode is het maximum van de voorgaande twee berekeningen.

Tijdens de **periodegebonden evaluatie (examen)** krijgen de studenten 4 uur om vier programmeeroefeningen op te lossen: twee Python-oefeningen en twee JavaScript oefeningen. Voor de Python oefeningen kunnen de studenten opnieuw gebruik maken van de Pythia omgeving om hun oplossingen in te dienen en feedback te krijgen op de correctheid en de programmeerstijl ervan. Voor de JavaScript oefeningen kunnen de studenten opnieuw gebruik maken van de Dodona omgeving om hun oplossingen in te dienen en feedback te krijgen op de correctheid ervan. Voor het bepalen van de examenscore worden de ingediende oplossingen evenwel opnieuw met de hand nagekeken en beoordeeld.

### Eindscoreberekening

De punten worden voor 35% (7/20) berekend op basis van niet-periodegebonden evaluaties (dagelijks werk) en voor 65% (13/20) op basis van een periodegebonden evaluatie (examen). Er zijn in totaal drie niet-periodegebonden evaluaties die samen de score van de niet-periodegebonden evaluatie bepalen.

De niet-periodegebonden evaluaties kunnen niet hernomen worden tijdens de tweede examenperiode. Voor de tweede examenperiode wordt enerzijds een score berekend rekening houdend met de punten behaald voor de niet-periodegebonden evaluatie (dus punten dagelijks werk gequoteerd op 7 samen met het examen gequoteerd op 13, zoals dat ook geldt voor de eerste examenperiode) en wordt anderzijds ook een score berekend zonder rekening te houden met de punten behaald voor de niet-periodegebonden evaluatie (examenresultaat rechtstreeks gequoteerd op 20). De eindscore voor de tweede examenperiode is het maximum van de voorgaande twee berekeningen.