Cursus Programmeren in Python – Docentenhandleiding (versie Basthon)

# Doelstelling

Het lesmateriaal is speciaal ontwikkeld voor het nieuw examenprogramma (vanaf schooljaar 2019). Als onderdeel van het leren imperatief programmeren is er ook aandacht voor het omzetten van een ontwerp (o.a. stroomdiagram) naar een gestructureerd programma vooraan. In het lesmateriaal is er ook expliciete aandacht voor het herkennen en gebruiken van standaardalgoritmen (oftewel plannen). Python is gekozen als programmeertaal, allereerst omdat het door zijn laagdrempelige syntax makkelijk te lezen en leren is. Daarnaast kent Python veel mogelijkheden zoals o.a. koppelen aan databases, webapplicaties, object-georiënteerd programmeren, machine learning, etc. Ook is Python een populaire taal in het bedrijfsleven en de wetenschappelijke wereld.

Nadat je de cursus hebt doorlopen, heb je voldoende kennis in huis om een (eenvoudig) spelletje te bouwen.

**De verwerkingstijd voor deze cursus is minimaal 40 uur.**

* Ongeveer 25 uur is voldoende om de basistheorie te behandelen. Dit kun je afsluiten met een of twee (tussentijdse)toetsen of een PO. Voor het PO is ongeveer 15 uur extra nodig.
* Voor de verdieping is ongeveer 5 uur nodig. Dit kun je afsluiten met een toets of een PO. Voor het PO is ongeveer 15 uur nodig.

# Leeromgeving

Zowel de theorie als het maken van programmeer opdrachten zijn volledig geïntegreerd in de browser (met behulp van Basthon, Jupyter notebooks). Voor het lesmateriaal (met uitzondering van onderwerp 14 – tekstbestanden) is er dus geen aparte programmeeromgeving of software nodig.

In de leeromgeving kan de leerling bovenaan bij 'Bestand' met 'Bewaar notebook als..' hun werk opslaan. Met 'Open' kunnen ze het een volgende keer weer inladen. Dit bestand kan ook ingeleverd worden bij de docent. Hiervoor kun je op het Informatica Actief website een inleveropdracht maken.

# Programmeeromgeving voor PO’s

Voor de PO’s en onderwerp 14 (tekstbestanden) kan gebruik gemaakt worden van Visual Studio Code (gratis beschikbaar voor Windows, Mac en Linux). Met Visual Studio Code kunnen leerlingen gebruik maken van versiebeheer (GitHub) en gelijktijdig samenwerken (LiveShare). Zie voor het gebruik van Visual Studio Code de bijbehorende handleiding die in de docentenkamer van Informatica-Actief:

* *VSC handleiding IA.docx*: Handleiding voor het installeren en instellen van Visual Studio Code en Python.
* *LiveShare in VSC handleiding IA.docx:* Handleiding voor het instellen en gebruiken van Live Share waarmee leerlingen gelijktijdig (‘live’) op afstand in het zelfde document kunnen werken.
* *GitHub in VSC handleiding IA.docx:* Handleiding voor het instellen en gebruiken van GitHub voor versiebeheer en het samenwerken aan dezelfde code.

# Opzet van de cursus

De opzet van deze cursus ziet er per onderwerp als volgt uit:

* Uitleg filmpjes: aan de hand van korte filmpjes wordt de belangrijkste theorie behandeld;
* Theorie en voorbeeldcode: de theorie is aangevuld met voorbeeldcode die je kunt runnen waardoor je beter begrip van de werking van Python kunt krijgen;
* Opdrachten: na elke theorieblok volgen een of meer opdrachten waarmee je je de theorie eigen maakt. Modeluitwerkingen zijn gegeven waarmee je je eigen oplossing kan controleren;
* Afsluitende Opdrachten: na elk hoofdstuk volgen afsluitende opdrachten waarmee je kunt nagaan of je het onderwerp voldoende hebt bestudeerd.

Het onderwerp kan afgesloten worden met één of meer (tussentijdse) toetsen, één of twee PO's, of een combinatie daarvan.

# Relatie met examenprogramma

De inhoud van deze cursus levert een bijdrage aan de volgende eindtermen:

**Domein A: Vaardigheden**

**Domein B: Grondslagen**

* Subdomein B1: **Algoritmen**
  + De kandidaat kan een oplossingsrichting voor een probleem uitwerken tot een algoritme, daarbij standaardalgoritmen herkennen en gebruiken, en de correctheid en efficiëntie van digitale artefacten onderzoeken via de achterliggende algoritmen.
* Subdomein B2: **Datastructuren**
  + De kandidaat kan: meerdere (abstracte) datastructuren herkennen, onderling vergelijken en beoordelen op toepasbaarheid.

Domein C: Informatie

* Subdomein C3: **Representeren**
  + De kandidaat kan voor een verzameling (geïdentificeerde) gegevens die in een programma verwerkt moeten worden, een geschikte datastructuur kiezen op grond van overwegingen van elegantie, efficiëntie en implementeerbaarheid.

**Domein D: Programmeren**

* Subdomein D1: **Ontwikkelen**
  + De kandidaat kan, voor een gegeven doelstelling, programmacomponenten ontwikkelen in een imperatieve programmeertaal, daarbij programmeertaalconstructies gebruiken die abstractie ondersteunen, en programmacomponenten zodanig structureren dat ze door anderen gemakkelijk te begrijpen en te evalueren zijn.
* Subdomein D2: **Inspecteren en aanpassen**
  + De kandidaat kan structuur en werking van gegeven programmacomponenten uitleggen, en zulke programmacomponenten aanpassen op basis van evaluatie of veranderde eisen.

# Toetsing

## Schriftelijke toetsen

Met een schriftelijke toets kan er per leerling beoordeeld worden waar ze staan, en welke misconcepten of hiaten er mogelijk zijn. Daarna kunnen vervolg stappen genomen worden om het leren te bevorderen. Schriftelijke toetsen kunnen ook formatief ingezet worden, waarbij het cijfer niet meetelt voor het PTA.

Er zijn drie (tussentijdse) toetsen beschikbaar:

* Toets 1: onderwerpen 1 t/m 7 (variabelen, keuzes, eigen functies, while).
* Toets 2: onderwerpen 1 t/m 10, met nadruk op de laatste onderwerpen 5 t/m 10 (eigen functies, for en while loops).
* Toets 3: onderwerpen 1 t/m 14, met nadruk op de verdiepende onderwerpen 11, 12 en 14 (geneste loops, strings en bestanden).

## Praktische opdrachten (PO’s)

Met praktische opdrachten kunnen leerlingen uitgedaagd worden om een creatieve oplossing te ontwerpen en ontwikkelen voor een grotere, minder afgebakende opdracht. Ze oefenen hiermee meer vaardigheden vanuit domein A, waaronder samenwerken en het gebruik van ondersteunende tools (bijvoorbeeld versie beheer), alsook domein B waarbij ze zelf een ontwerp moeten opstellen.

Er zijn twee PO’s bijgeleverd (inclusief beoordelingsmodel):

* PO Galgje:
  + Tijdspad: na onderwerp 10 – plannen met lijsten.
  + Kennis: Deze is gebaseerd op het gebruik maken van **lijsten** van letters (en dus niet Strings), omdat hiermee het toepassen (en aanpassen) van **standaard plannen** (voor het doorlopen van lijsten) geoefend wordt en dus ook getoetst kan worden.
  + Relatie met examenprogramma:
    - Subdomein A8: Ontwerpen en ontwikkelen: programma ontwerp maken, omzetten naar een werkend programma en reflecteren op gemaakte keuzes.
    - Subdomein A12: Informatica-instrumentarium hanteren: programmeeromgeving & versiebeheer
    - Subdomein B1: Algoritmen: Programma ontwerp waarbij gebruik wordt gemaakt van standaardalgoritmen
    - Domein D: Programmeren: Ontwikkelen, evalueren en verbeteren.
  + Beoordeling: zie beoordelingsmodel in de omschrijving van PO Galgje.
  + Aandachtspunt: Er zijn veel uitwerkingen van Galgje online beschikbaar. Door leerlingen te verplichten om gebruik te maken van lijsten (in plaats van strings) wordt fraude bemoeilijkt.
  + Uitbreidingen: Er zijn meerdere uitbreidingen en verdiepingen mogelijk om leerlingen mee uit te dagen
* PO Zeeslag:
  + Tijdspad: na onderwerp 11 – geneste loops.
  + Kennis: Leerlingen maken gebruik van andere **datastructuren** (geneste lijsten) en complexere **algoritmen.**
  + Relatie met examenprogramma:
    - Zelfde als PO Galgje, met aansluitend:
    - Subdomein B2: Datastructuren & Subdomein C3: Representeren: Keuze van datastructuur (geneste lijsten).
    - Subdomein B1: Algoritmen: Keuze/ontwerp van algoritme heeft bij deze opdracht zichtbaar effect op efficiëntie en elegantie van de code.
  + Er worden uitvoerige tips en aanwijzigen gegeven. Als docent mag je ervoor kiezen om deze weg te laten om de leerlingen meer uit te dagen.
* Andere ideen voor POs zijn mijnenveger, lingo, tekst-based-game, yahtzee, wachtwoord generator, bingo, woordzoeker

.