

# Python in de klas

---

## Voor wie is dit materiaal?

---

Dit materiaal is voor iedere leerkracht die in de klas Pythonlessen wil geven. Het materiaal is echt bedoeld voor in een klassikale setting, met veel uitleg en oefenen. Je kunt de lessen zelfs zonder computers met alleen een digibord gebruiken (sla dan deel b over).

## Wat heb ik nodig om dit materiaal te gebruiken?

---

- Een digibord
- Een schift voor iedere leerling
- (Optioneel) wisbordjes (hierover later meer)

## Moet ik zelf veel Python kunnen om deze lessen te geven?

---

Niet echt hoor. De slides leggen alles uit dat je moet weten per les, dus die kun je vooraf lezen en dan ga je het wel redden.

Voor deel b is het wel handig als je iemand in de klas hebt die het een beetje kan, maar dat kan ook goed een leerling zijn, uit jouw klas of uit een hogere klas, of een vader of moeder die in het begin meehelpt.

## Opzet van de lessenserie

---

Het idee van deze lessen is om leerlingen te laten zien dat programmeren zo breed is. Vaak denken mensen bij programmeren aan robots maken en websites en apps, maar je kunt programmeren ook heel goed gebruiken bij 'gewone' schoolvakken. Daarom gaat ieder hoofdstuk van deze serie over een ander vak. Er zijn nu drie modules: Nederlands, Art en Aardrijkskunde. Met opzet is er gekozen voor een de niet-betavakken, juist ook om andere leerlingen aan te spreken dan traditioneel voor programmeren kiezen.

In de modules zelf vind je meer details over de opdrachten in de modules.

# Opzet van de modules

---

Iedere module bestaat uit een aantal lessen. Een les bestaat steeds uit drie stukken: a, b en c.

- a. klassikale les
- b. werkbladen zodat IIn zelf aan de slag kunnen achter de computer
- c. klassikale afsluiting

## Deel a

Deel a is een klassikaal uit te voeren les. Je kunt hiervoor, als je wilt, de meegeleverde slides gebruiken. Voor deel a (en deel c) zijn geen computers nodig, alleen een digibord en schrijfmaterialen voor IIn: schriftjes of wisbordjes.

## Deel b

In deel b gaan IIn zelf aan de slag met werkbladen.

Bovenaan de werkbladen staan steeds doelen voor IIn. Bijvoorbeeld: "deze opdracht wordt afgetekend als er een vierkant is getekend".

Die doelen kun je in principe aftekenen zelfs als je niet zoveel programmeerkennis hebt. Het is immers een kwestie van kijken of er een vierkant staat. Daar waar je iets moet opzoeken de code van een IIn, is ook dat zo uitgelegd dat iedereen het kan aftekenen, bijvoorbeeld "er is een vierkant getekend, en de code bevat 'for'".

Natuurlijk verlopen lessen het allerbest als de leerkracht zelf vaardig is in Python en de bijbehorende problemen en misconcepties maar de realiteit is dat niet alle leerkrachten dat op dit moment zijn. Dat is voor deze serie geen probleem! Ook de te verwachten problemen en vragen van IIn worden steeds uitgelegd in de leerkrachteninstructie per lesdoel.

Nog een paar tips over deel b:

- Sommige leerlingen zullen andere dingen gaan proberen in (of zelfs buiten) Python. Spreek vooraf af of dit wel of niet mag. Exploreren is natuurlijk leuk, maar brengt ook risico's met zich mee, namelijk:
  - Hele goede leerlingen demotiveren klasgenoten
  - Leerlingen leiden anderen af met bijv. tekeningen van gezichtjes
- Zelf dingen uitproberen in Python leidt tot verwarring en frustratie

- Sommige lesdoelen hebben "(extra)" in de titel. Die zijn voor snellere of meer ervaren leerlingen. Die kunnen op de volgorde in de lessenserie waar ze staan gedaan worden, of op het einde van alle lessen. Net hoeveel je wilt dat ze leerlingen uit elkaar gaan lopen qua waar ze zijn.

## Deel c

Deel c is weer klassikaal uit te voeren les. Dat kan heel kort zijn, 5 of 10 minuutjes om even samen terug te kijken op het individuele deel. De laatste twee slides van deel a, waarin de leerdoelen herhaald worden, kun je hiervoor gebruiken.

Ik vraag dan per leerdoel of Iln het idee hebben dat ze het ook echt kunnen.

## Algemene tips!

Een aantal algemene tips rondom de lessenserie:

- Reken op ongeveer 2 klokuren voor delen a, b en c bij elkaar: ca 45 minuten voor a, 45 min voor b en 10 minuten voor c, plus tussenin tijd om van de klassikale opzet naar de computers te gaan.
- Als je wisbordjes hebt, raad ik zeker aan die te gebruiken! Een van de uitdagingen van een programmeerles is dat leerlingen soms de mooiste dingen maken, zonder dat ze het goed snappen! Ze proberen dingen tot dat het werkt, of zoeken dingen op op YouTube. Helemaal geen probleem natuurlijk, maar soms hebben ze dan veel minder begrip dan jij denkt. Daarom is het goed dat begrip regelmatig te toetsen, juist ook omdat jij als Ik misschien zelf ook nog niet alle ins en outs van Python ken. Ik programmeer zelf al 20 jaar en zelfs ik snap niet altijd alles wat Iln in elkaar knutselen!
- Als je nieuwe codes laat zien, lees ze dan hardop voor, en laat leerlingen dat ook regelmatig doen. Uit ons eigen onderzoek blijkt dat Iln dan beter onthouden hoe ze moeten programmeren.
- Bijvoorbeeld: `pen.forward(10)` is "pen punt forward rond haakje openen 10 rond haakje sluiten"
- Wees ook heel consequent in het benoemen van karakters in de code. Dus "(" is niet een haakje, maar rond haakje. Op de slides proberen we dat ook zo consequent mogelijk te doen. Als je wilt, kun je onze poster met karakters in je lokaal ophangen.
- Sommige leerlingen hebben weinig ervaring met computeren. Het vinden van zaken op

het toetsenbord kan al lastig zijn, zoals de min of de 'tab'. Ook vaardigheden als knippen en plakken met ctrl-c en ctrl-v ontbreken regelmatig, en ook de ctrl-z voor ongedaan maken kennen ze niet altijd. OF ze weten niet hoe ze het moeten invoeren en proberen steeds heel precies ctrl en z tegelijk in te drukken ipv eerst de ctrl en dan de z erbij. Dit zijn geen leerdoelen van deze serie lessen maar de kans is groot dat je er wel tegenaan loopt.

- Stip steeds even de betekenis en goede uitspraak van de Engelse woorden in de code aan. Ik hoor nog regelmatig leerlingen worstelen met wat die woorden nou zijn. Dit wordt in het materiaal ook benoemd maar uitspraak zeker is ook handig als leerkrachten dat benadrukken. En soms betekenen woorden ook net iets anders in Python. In de tekentool heb je bijv. het commando 'left'. Dat betekent natuurlijk links, maar in deze code betekent het "draai linksom", dus niet "ga naar links". Dat is ook belangrijk om dat even te delen met leerlingen.