

CodeCombat: tips en tricks

Website

Ga naar de website: <https://programmerenindeklas.weebly.com/6l.html> en klik op CodeCombat. Er opent een nieuwe website en kies daar voor de gele knop: *Ik ben een leerling*.



Je krijgt dan een scherm te zien waar ze vragen naar je *klassencode*. Dit heb je niet nodig. Je gebruikt je eigen *login* die je van de leerkracht kreeg om je aan te melden.

- Heb je nog geen **eigen login gegevens**? Vraag ze aan juf Evelyne door een mailtje te sturen naar evelyne.rotty@coloma.be

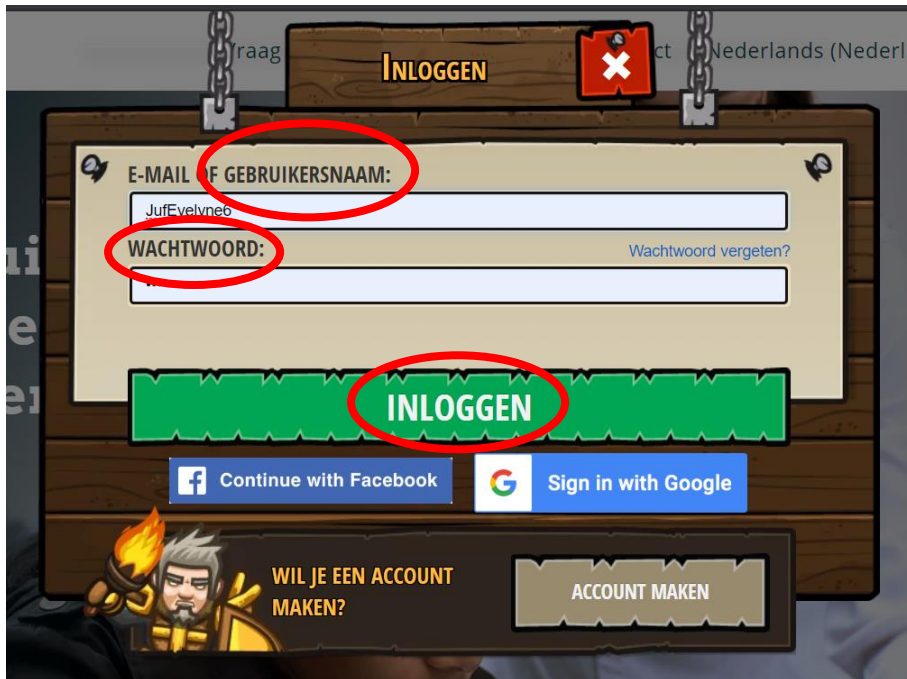
Creëer leerling account

Vul je klassencode in

Vraag je leraar om een klassencode.
Maak je geen deel uit een klas? Maak dan een [Individuele account](#).

Heb je al een account? [Log in](#)

Vul je **gebruikersnaam** juist in. LET OP! Gebruik hoofdletters en cijfers waar nodig.
Onder je gebruikersnaam vul je het **wachtwoord** in.
En klik op de grote groene knop: *inloggen*.



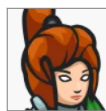
Jouw account

Je komt terecht op je *cursus-pagina*. Hier vind je verschillende zaken terug: je held, de klas, de taal waarin je leert coderen (Python) en je voortgang.

Om te **leren programmeren**, klik je op de groene knop: *doorgaan*.

[Over Ons](#) [Impact](#) [Mijn lessen](#)  [Mijn Account](#)

Welkom op je cursus-pagina!



Huidige Held: Anya

[Verander held](#)

Mijn klassen

6B (Python)

Leerkracht: Juf.Evelyne

[Inleiding tot ComputerTechnologie](#) [bekijk kaart](#) [Bekijk Challenge Levels](#) [bekijk video's](#)

Laatste Level: 4. Vijandelijke Mijn

[Doorgaan](#)

16.7%

DOE MEE MET EEN KLAS

De map

Nu krijg je de map te zien met alle levels die je kan doorlopen tijdens jullie *introductie naar informatica*. Een eerste stap in het leren van programmeren.

Klik op het eerste level links onderaan: *De kerkers van Kithgard*.



Telkens wanneer je een **nieuw level** vrijspeelt, krijgt het **bolletje** een andere kleur en een **vlag**.



Een **groen bolletje** is een extra oefening om iets nieuw dat je hebt geleerd, nog eens uit te proberen.

Een **paars bolletje met blauwe vlag** staat voor een *Concept of Combo challenge*. In deze levels kan jij jezelf eens testen. Hoe goed kan jij de geleerde programmeertaal gebruiken?

Een level

Je kan eindelijk starten aan je eerste level!

Wanneer het level laadt, zie je meteen de **doelen** die je tijdens dat level zal moeten voltooien. Deze **argumenten** kan je ook in het blauw aan de rechterkant van je scherm terugvinden. Ze starten telkens met een **#**. Ze worden in het Engels **comments** genoemd.



In het eerste level krijg je al wat uitleg van het spel. Lees dit goed!

Links zie je het **speelveld** met tips. **Onderaan** heb je nog een balk met play, muziek op/af, een vergroot- en verkleinglas. Met + zoom je in, met – zoom je uit.

In het **midden** zie je een strook waar de **programmeertaal** staat waartussen je zal kunnen kiezen. Dit noemen we ook wel eens de **methodes**.

Rechts zal jij zelf de programmeertaal moeten typen en **coderegels** schrijven.



De video's die je te zien zal krijgen, kan je niet in het Nederlands bekijken. De uitleg die ze hier geven, kan je ook in deze bundel terugvinden.

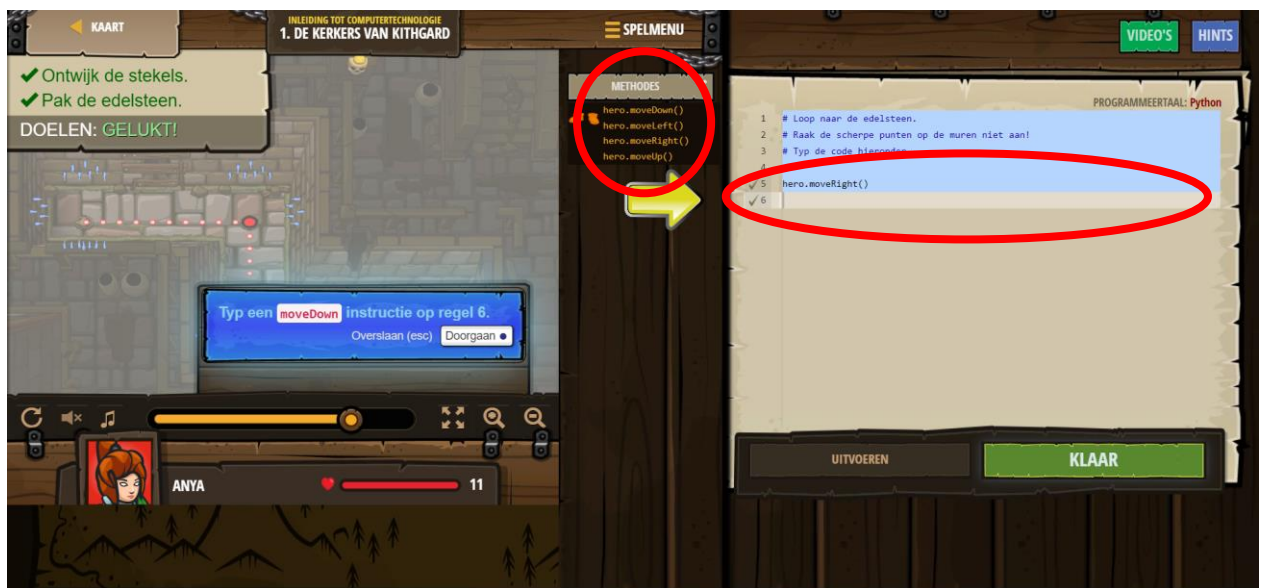
Level 1

De kerkers van Kithgard

De eerste regel werd al ingevuld.

Schrijf telkens regel per regel, laat geen openingen.

Naar beneden gaan = `hero.moveDown()`
Naar boven gaan = `hero.moveUp()`
Naar rechts gaan = `hero.moveRight()`
Naar links gaan = `hero.moveLeft()`



Programmeertaal ingevoerd? Klik op *uitvoeren*.

Gelukt? Klik op *klaar*.

Zorg eerst voor een *goed plan*. Weet waar je naartoe wilt gaan.

Bugs zijn foutjes die in je code zijn geslopen. Dit is heel normaal bij programmeren. Daarom is het belangrijk geregeld een test uit te voeren om na te gaan of je code wel goed werkt.

Level 2

Schat in het diepe

Kijk goed naar het voorbeeld aan de rechterkant. Dit toont je het plan dat je moet volgen. Later zal je zelf een plan moeten bedenken.

TIP: De haakjes na een code, bijvoorbeeld `hero.moveUp()` kan je gebruiken om eenzelfde beweging meermaals uit te voeren. Je wilt bijvoorbeeld dat jouw hero twee plaatsen naar boven gaat, dan gebruik je de code: `hero.moveUp(2)`

Enkel de levels waarin een nieuwe moeilijkheid zit, worden hieronder uitgelegd. De andere levels zou je met de nodige oefening zelf moeten kunnen oplossen.

Level 3

Schaduwwachter

TIP: Een regel vergeten? Ga bij de vorige coderegel staan en klik op je *Enter-toets*. Zo kan je eenvoudig een extra regel invoeren.

a. *Tegen de Kith in*

Let op! De ogers zullen zich tijdens het uitvoeren van je code verplaatsen. Gelukt? Nu kan je de eerste *Challenge* spelen.

b. *Kruipgangen van Kithgard*

TIP: Soms helpt het om op je stappen terug te gaan.

Level 4

Vijandelijke mijn

Vanaf nu moet je goed op het aantal regels code letten. Bij level vier is het maximum bijvoorbeeld 5 regels. Lees de doelen telkens goed, voordat je het level start.

Let op! De blauwe knipperende cirkels zijn bommen. Deze moet je ontwijken.

Challenge

Lange stappen

TIP: Kijk goed naar de **argumenten**. Daar moet je eerst iets aanpassen.

Level 5

Ware namen

Iemand aanvallen = **hero.attack(target)**

In dit level leer je een vijand aanvallen. Kijk goed naar het voorbeeld. Daar tonen ze jou hoe je twee keer de oger Brak kan aanvallen. Dit hoef je dus zelf niet meer te programmeren.

Let op! Hoofdletter gebruiken bij het begin van een naam.

Level 6

Cel commentaar

Je hero iets laten zeggen = **hero.say(message)**

TIP: Kijk bij de argumenten welk wachtwoord je moet gebruiken.

Level 8

De gevangene

Zorg ervoor dat je het plan goed volgt. Schrijf elk deel van de code onder de juiste argumenten.

TIP: Verplaats je hero ver genoeg van de bewaker Two voordat jij hem aanvalt. Patrick, de gevangene, kan met zijn pijl en boog de oger al aanvallen vanop een afstand.

Level 9

Vuurdansen

Iets herhalen = **while True loop**

Deze is al iets moeilijker om te leren gebruiken. De **while True loop** gebruik je om een bepaalde beweging, of meerdere bewegingen na elkaar, te herhalen. Zo lang je iets in de **while True loop** typt, zullen deze codes zich blijven herhalen.

Let op! Je start met de regen **while True:** . Dan ga je naar de volgende regel en zet je **4 spaties**, anders werkt het niet. Nu typ je regel per regel de code die moet herhaald worden.



```
1 # 'Loops' zijn een betere manier om iets te doen dat steeds moet worden herhaald.
2
3 while True:
4     # Voeg hier acties die moeten worden herhaald.
5     hero.moveRight(2)
6     hero.moveUp(2)
7
8
```

VIDEO'S HINTS HERSTARTEN

PROGRAMMEERTAAL: Python

4 spaties

(Dit is de oplossing van level 10)

Challenge

Loop pakuis MOEILJK!

TIP: Gebruik enkel de **while True loop**. Vind je de oplossing niet? Kijk dan eens naar de beweging van de kleine hero die af en toe voorbij komt. Volg het patroon dat hij maakt.

Level 15

Bekende vijand

Variabele gebruiken = ... = “...”

Variabelen kan je gebruiken om een code te schrijven waarbij je geen naam nodig hebt.

Eerst heb je geleerd om een de slechte oger Brak aan te vallen met deze code: `hero.attack("Brak")`

Met een variabele heb je de naam van de oger niet meer nodig in je code, door deze eerst met een ander woord te linken: `vijand1 = "Brak"`

Level 16

Meester der namen

In dit level leer je de variabele op een tweede manier gebruiken. Nu ken je zelfs de namen niet meer van de ogers. Link daarom eerst de vijanden aan een code.

Gebruik daarvoor: `enemy1 = self.findNearestEnemy()`

Het spel heeft voor jou het eerste deel van de code al geschreven. De eerste vijand die je zal tegenkomen (NearestEnemy = de dichtstbijzijnde vijand) heeft de naam: enemy1 gekregen.

Ze vallen de vijand twee keer aan en schrijven de coderegels: `enemy2 = self.findNearestEnemy()`

Nu heeft de tweede vijand, waarvan we de naam niet kennen, een waarde gekregen: enemy2.

```
1 # Jouw held kent de namen van deze vijanden niet!
2 # De bril geeft jou de findNearestEnemy (vindDichtstbijzijndeVijand) methode.
3
4 # Wijs het resultaat van hero.findNearestEnemy() toe aan de variabele enemy1:
5 enemy1 = hero.findNearestEnemy()
6 # enemy1 verwijst nu naar de dichtstbijzijnde vijand. Gebruik de variabele om aan te vallen!
7 hero.attack(enemy1)
8 hero.attack(enemy1)
9
10 # Enemy1 is nu verslagen, dus als je hero.findNearestEnemy() nog een keer aanroept, zul je de nieuwe d
11 enemy2 = hero.findNearestEnemy()
12 hero.attack(enemy2)
13 hero.attack(enemy2)
14
15 # Wijs het resultaat van hero.findNearestEnemy() toe aan de variabele enemy3:
16 enemy3 = hero.findNearestEnemy()
17
18 # Val nu aan met de enemy3 variabele:
19 hero.attack(enemy3)
20 hero.attack(enemy3)
21
```

(Dit is de oplossing van level 16)

Level 17

Het laatste Kithhof.

Start met de **while True loop**. Haal deze eerst leeg door het woordje 'pass' te verwijderen. Laat je hero zich eerst verplaatsen zodat hij voor de vijand staat. Pas dan gebruik je de variabele: **enemy = self.findNearestEnemy()**, gevolgd door je aanval op de vijand.

TIP: Onder **while True:** moet je 8 regels schrijven.

Level 18

Poorten van Kithgard

Muur bouwen = **hero.buildXY("fence", x-as, y-as)**

In dit level leer je een muur te bouwen om de vijanden tegen te houden. Een deel van de muur werd voor jou al gebouwd, maar dit is niet genoeg. Je zal zelf nog twee stukken van de muur moeten bouwen op de botten-kruisen.



Ga met je muis op de onderste botten staan. Je krijgt dan te zien op welke coördinaten je een muur kan bouwen. In dit geval zie je: **x: 36, y: 30**.

Deze twee getallen vul ik in, in mijn code: **hero.buildXY("fence", 36, 30)**

Vergeet ook niet het derde deel van de muur te zetten. Doe dit volgens dezelfde stappen.

Wakka Maul

Als je tot aan het laatste level bent geraakt, kan je tegen elkaar spelen. Je kan als oger of als mens spelen. Probeer de tegenpartij te slim af te zijn!