

# PyCharm: installatie en gebruik

24 september 2014

## 1 Introductie

Dit document zet je op weg om met PyCharm te werken. De PyCharm ontwikkelomgeving heeft als doel om je leven als Python programmeur gemakkelijker te maken. Het helpt je bij het uitvoeren van je code, het opsporen van fouten, enz. We beschrijven eerst hoe je Python en PyCharm installeert op je eigen machine (in de PC-klassen zijn Python en PyCharm reeds geïnstalleerd), vervolgens overlopen we de basisfunctionaliteit van PyCharm.

## 2 Installatie

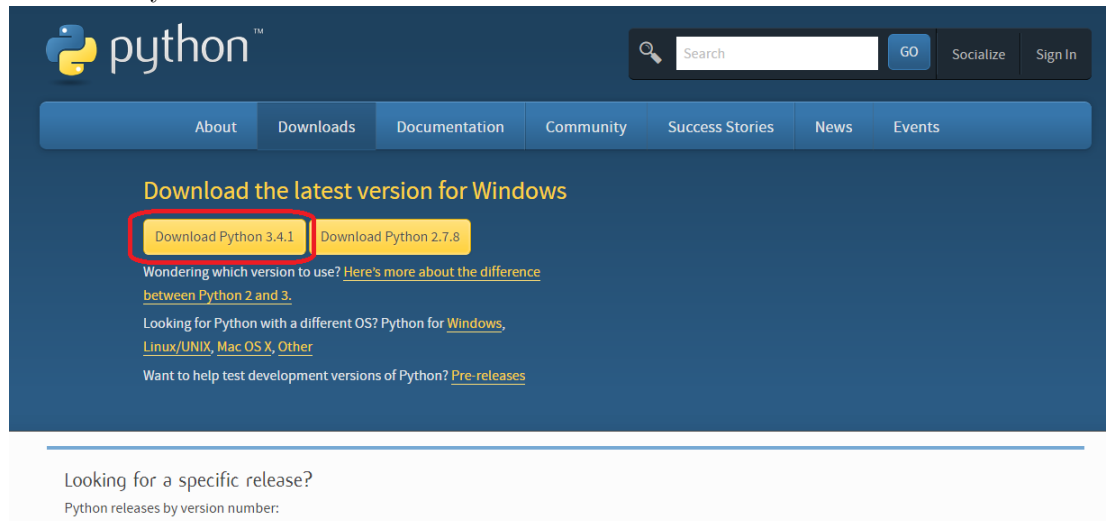
### 2.1 Python interpreter

Om Python code te kunnen uitvoeren op een machine heb je een programma nodig dat Python broncode vertaalt naar machinecode. In deze subsectie overlopen we hoe je dit programma (de Python interpreter) installeert. De installatieprocedures hieronder zijn op een Windows PC uitgevoerd. Voor Mac gebruikers is de installatie gelijkaardig. Linux gebruikers moeten de broncode zelf compileren<sup>1</sup>.

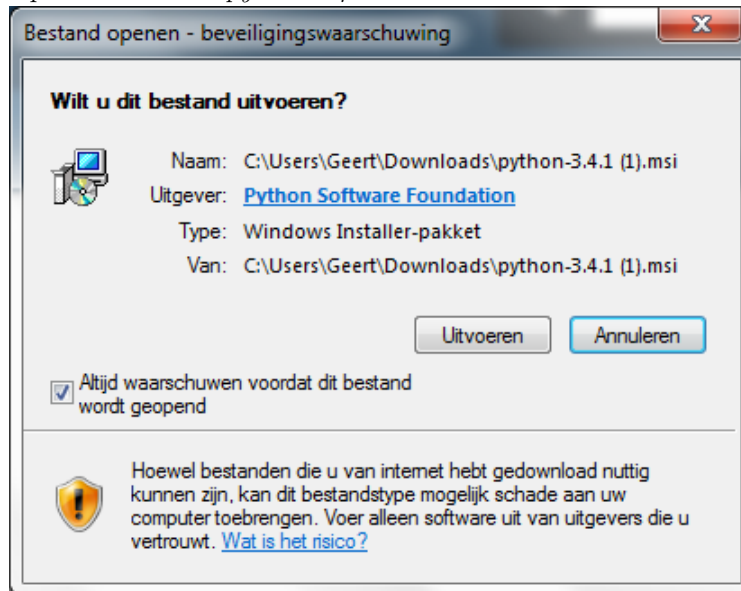
---

<sup>1</sup>[https://www.youtube.com/watch?v=sKiDj0\\_0dCQ](https://www.youtube.com/watch?v=sKiDj0_0dCQ)

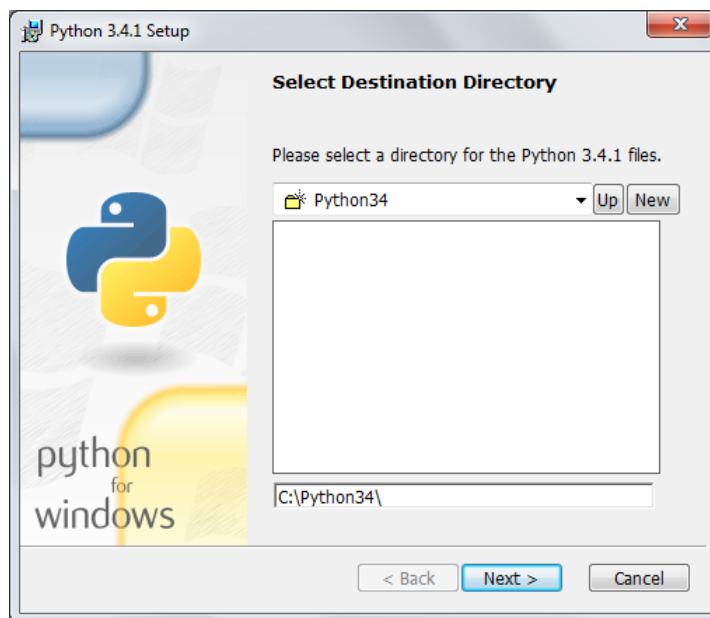
1. Surf naar <https://www.python.org/download> en download de laatste versie van Python 3.

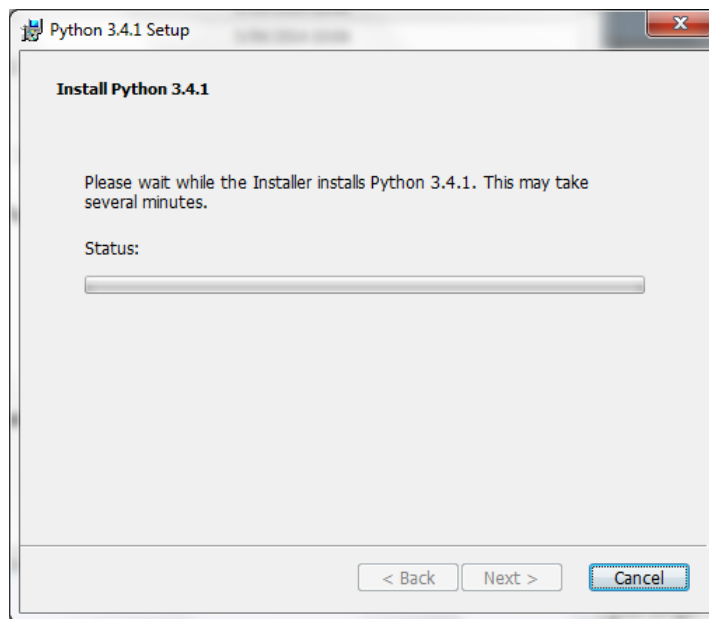


2. Open het bestand *python-3.4.1.msi*.



3. Volg de installatiestappen.







## 2.2 PyCharm

Als de installatie van de Python interpreter gelukt is, kan je in principe beginnen programmeren: je kan code schrijven in een standaard tekstverwerker zoals kladblok en code uitvoeren via commandline. Maar dit is niet erg handig. Typisch programmeert men vanuit een ontwikkelingsomgeving. In deze cursus gebruiken we PyCharm. De installatie verloopt als volgt:

1. Surf naar <http://www.jetbrains.com/PyCharm/download/> en download de *Community Edition*.

## Download PyCharm

[Windows](#)  
[Mac OS X](#)  
[Linux](#)

[See what's new in PyCharm 3.4 »](#)

### Professional Edition **Free 30-day trial**

- Full-featured IDE for Python & Web development
- Supports Django, Flask, Google App Engine, Pyramid, web2py
- JavaScript, CoffeeScript, TypeScript, CSS, Cython, Template languages and more
- Remote development, Databases and SQL support, UML & SQLAlchemy Diagrams

 [Download Professional](#)

### Community Edition **FREE**

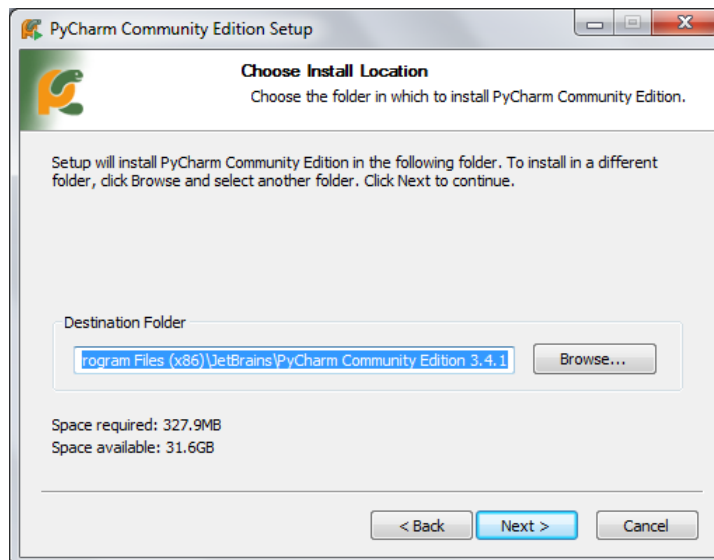
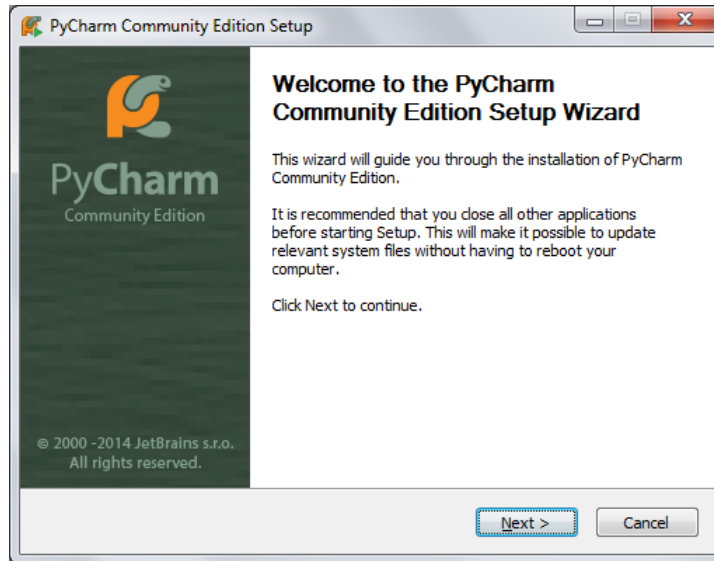
- Lightweight IDE for Python development only
- Free, open-source, [Apache 2 license](#)
- Intelligent Editor, Debugger, Refactorings, Inspections, VCS integration
- Project Navigation, Testing support, Customizable UI, Vim key bindings

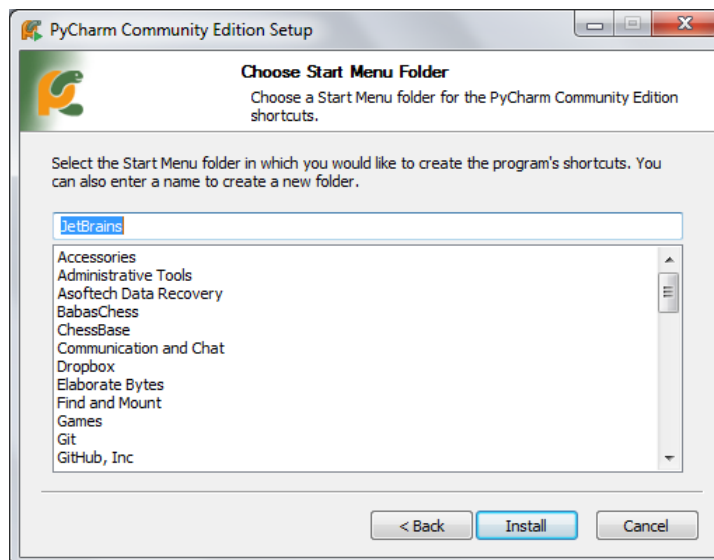
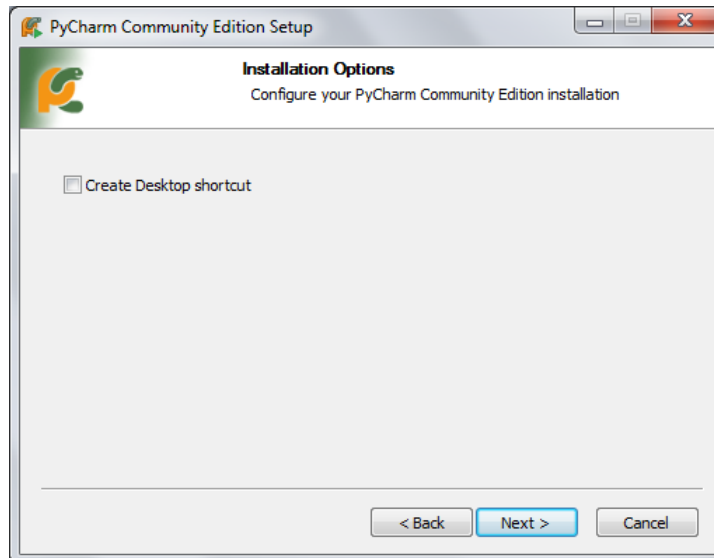
 [Download Community](#)

2. Open het bestand *PyCharm-community-3.4.1*.



3. Volg de installatiestappen. Je kan alles op de standaardwaarden laten staan.







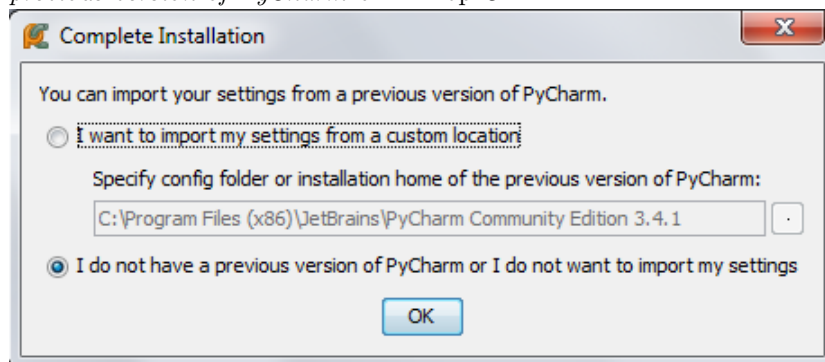


PyCharm is nu geïnstalleerd.

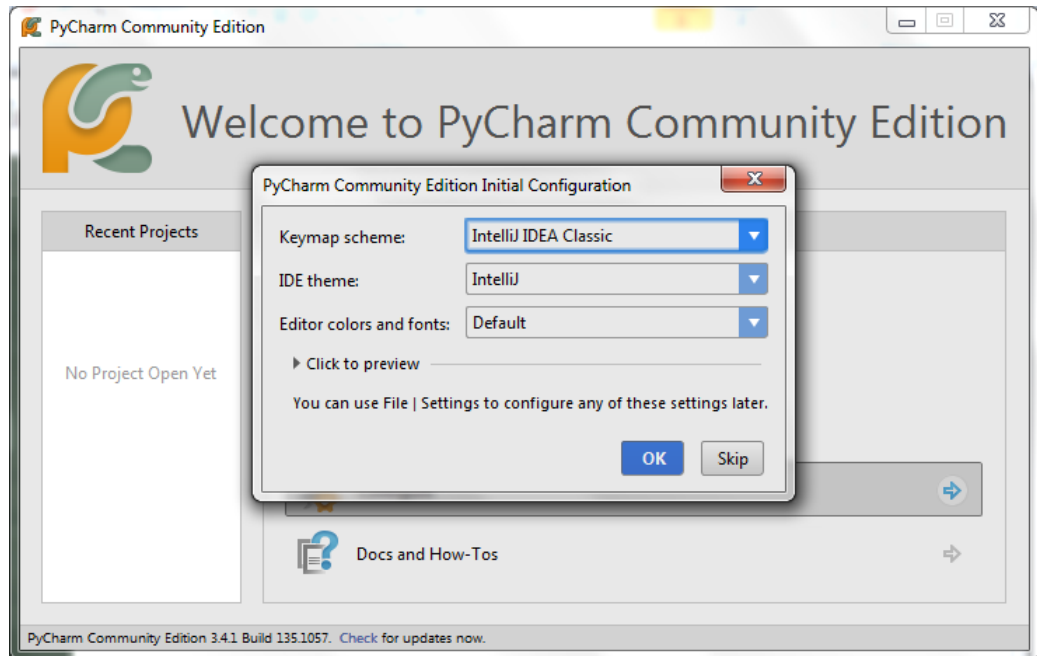
### 3 Configuratie PyCharm

De eerste keer dat je PyCharm opent moet je enkele instellingen configureren. Dit geldt ook voor de installaties van Pycharm op de computers in de pc-klassen.

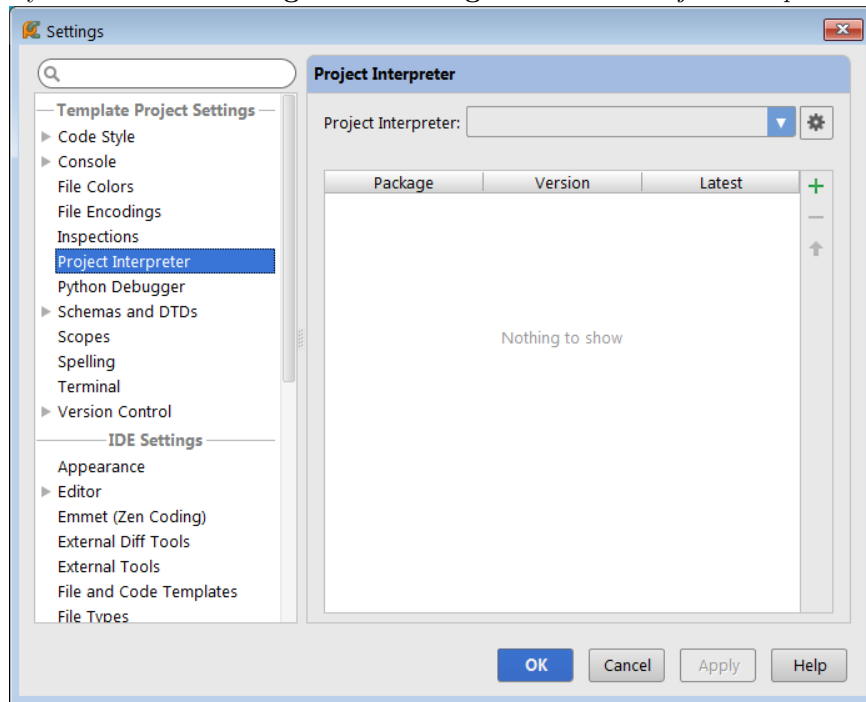
1. Open PyCharm (je vindt Pycharm in C:\Program Files (x86)\JetBrains\PyCharm Community Edition\bin\pycharm.exe). Onderstaand dialoogvenster verschijnt. Laat staan op *I do not have a previous version of PyCharm* en klik op *OK*.



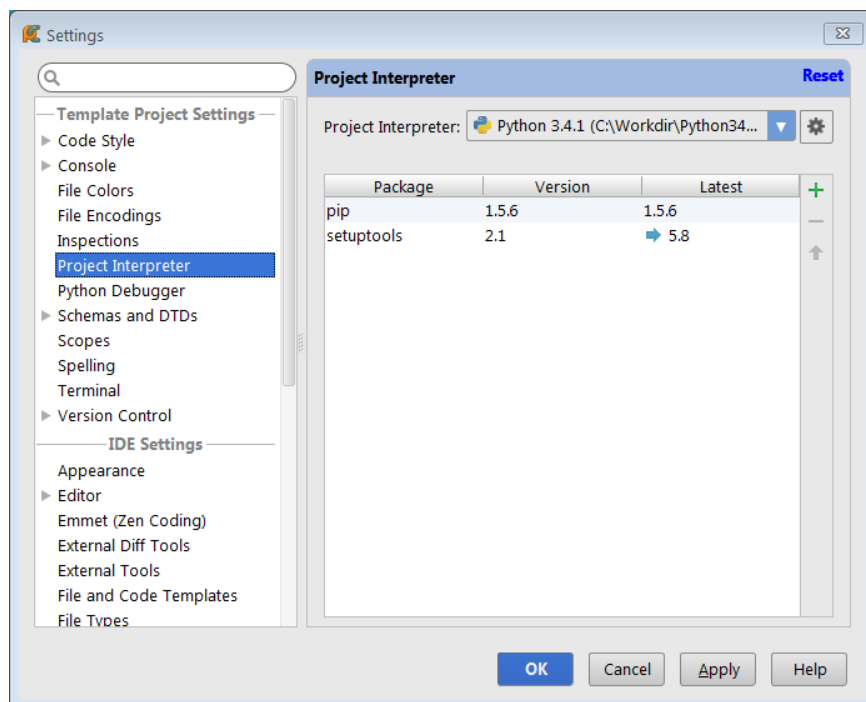
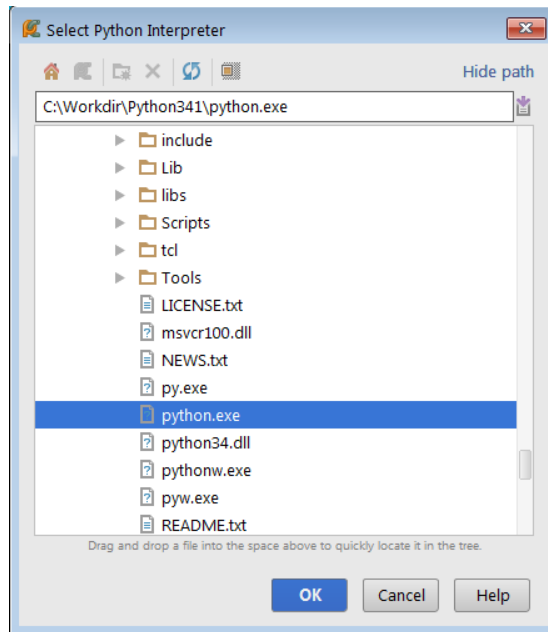
2. Onderstaand dialoogvenster verschijnt. Hier kan je de sneltoetsen en het uitzicht van PyCharm wijzigen. Indien je vroeger in een andere ontwikkelomgeving hebt gewerkt kan je hier de sneltoetsen overnemen. Klik op *OK*.



3. Als laatste stap stellen we PyCharm in met de juiste versie van Python. Ga naar **Configure** → **Settings** en selecteer *Project Interpreter*.



Klik op het tandwielje en klik vervolgens op *Add Local*. Geef het pad naar de Python 3 interpreter in: voor de pc-lassen is dit `C:\Workdir\Python34\python.exe`. Als je PyCharm op je eigen (Windows) computer installeert is dit pad normaal `C:\Python34\python.exe`. Wanneer de interpreter is toegevoegd, klik op *OK*. PyCharm is nu klaar voor gebruik.




## 4 PyCharm gebruiken


In deze sectie overlopen we de basisfunctionaliteit van PyCharm.

**Project openen** Een project open je via **File**→**New Project....** De eerste keer dat je PyCharm opent, krijg je mogelijk het Quick Start menu. In dat geval klik je gewoon **Create New Project** om een nieuw project aan te maken.

Opgelet: als je de computers van de pc-klassen gebruikt, wil PyCharm je project zetten in de map `\stud-\home.icts.kuleuven.be\s0211197\PycharmProjects`. Dit geeft problemen omdat je project zich dan op de netwerk harde schijf bevindt, terwijl PyCharm op de lokale schijf is geïnstalleerd. Je maakt je project dus best aan in de map `C:\Users\jouw-r-nummer\PycharmProjects`.

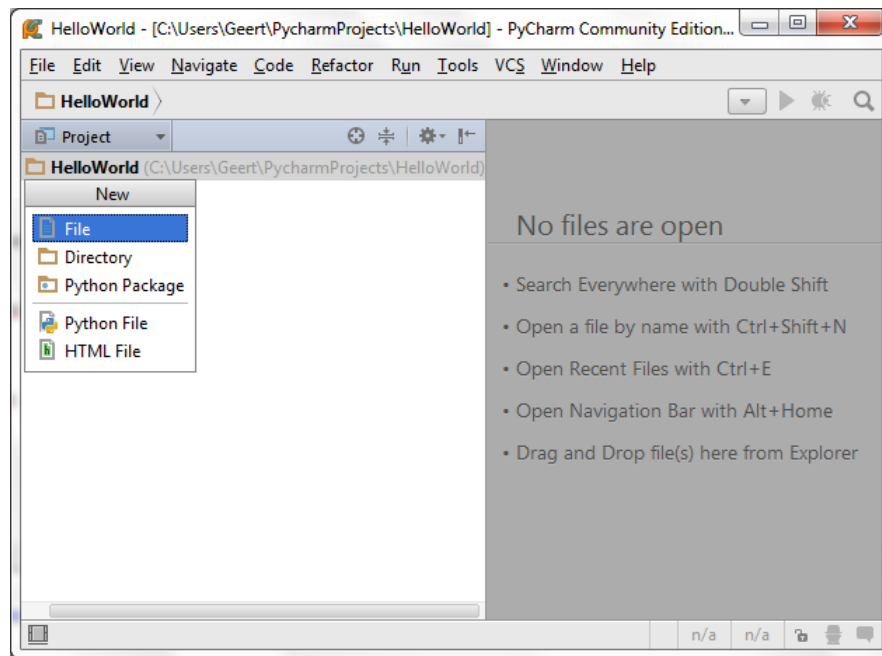
**Wisselen tussen vensters** De PyCharm interface bestaat uit verschillende vensters: Project, Run, Debug, TODO, etc. Je kan vensters zichtbaar maken of laten verdwijnen door met je muis op volgend icoon te staan  (of door de snelkoppelingen te leren: alt+1, alt+2, etc.).

**Bestanden toevoegen aan je project** Selecteer je project en navigeer vervolgens naar **File**→**New....** Je code schrijf je in *Python Files*.


**Code uitvoeren** Klik met rechtermuisknop op het bestand dat je wil uitvoeren en klik vervolgens op *Run*. Onderaan verschijnt het Run venster. Dit venster kan je gebruiken voor standaard invoer en uitvoer. Zo zullen je print statements hier verschijnen. Je kan de uitvoering stoppen m.b.v. het stop icoon:  (kan o.a. handig zijn wanneer je een oneindige lus hebt geschreven).

**Syntax errors** Wanneer je ongeldige Python code schrijft, zal PyCharm dit aangeven door de code rood te onderstrepen. Wanneer je de cursor op de code zet, toont PyCharm een rood lampje met suggesties om de error op te lossen en verschijnt er links onderaan een foutmelding (zie Figuur 1).





**Debugger** Wanneer je code niet doet wat je verwacht, is het vaak nuttig om de uitvoering van je programma op een bepaalde regel te stoppen om de toestand te inspecteren (i.e., de waarden van de variabelen). Dit doe je m.b.v. een debugger:

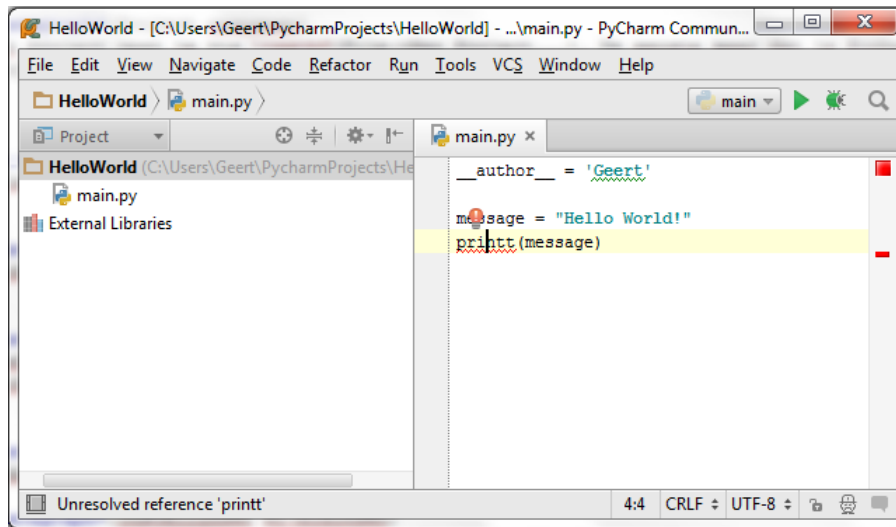


Geef aan op welke regel(s) je de uitvoering wil pauzeren. Dit doe je door een *breakpoint* zetten op een regel: klik in de grijze balk links van een regel. Een rood bolletje verschijnt.

Klik met rechtmuisknop op het bestand dat je wil debuggen en klik vervolgens op *Debug*. Onderaan verschijnt nu het debug venster (zie Figuur 2). In het *Variables* paneel zie je alle variabelen die in scope zijn met hun waarden op de regel die gedebugd wordt (je kan naar deze regel springen door op volgend icoon te klikken ). De variabelen zijn gerangschikt op alfabetische volgorde. Naast het *Variables* paneel vind je het *Watches* paneel. Hier kan je de waarden van expressies die je zelf invoert inspecteren.

Wanneer de debugger gestopt is aan een breakpoint heb je verschillende opties:

- continue execution  : voer de rest van de code uit (eventueel tot aan een volgend breakpoint)
- step over  : ga naar de volgende regel
- step into  : indien de huidige regel een functie aanroept, spring je naar het begin van die functie
- step out  : na een oproep van step into, ga terug naar de regel die de functie oproep deed



Figuur 1: PyCharm signaleert onmiddellijk je syntax errors.

**Instellingen wijzigingen** Instellingen wijzigen doe je via **File**→**Settings**. Voorlopig zal je hier niet zoveel aan willen wijzigen. Het kan wel handig zijn om regelnummering aan te zetten. Dit doe je door bij **Editor**→**Appearance** *Show line numbers* aan te vinken (zie Figuur 3).

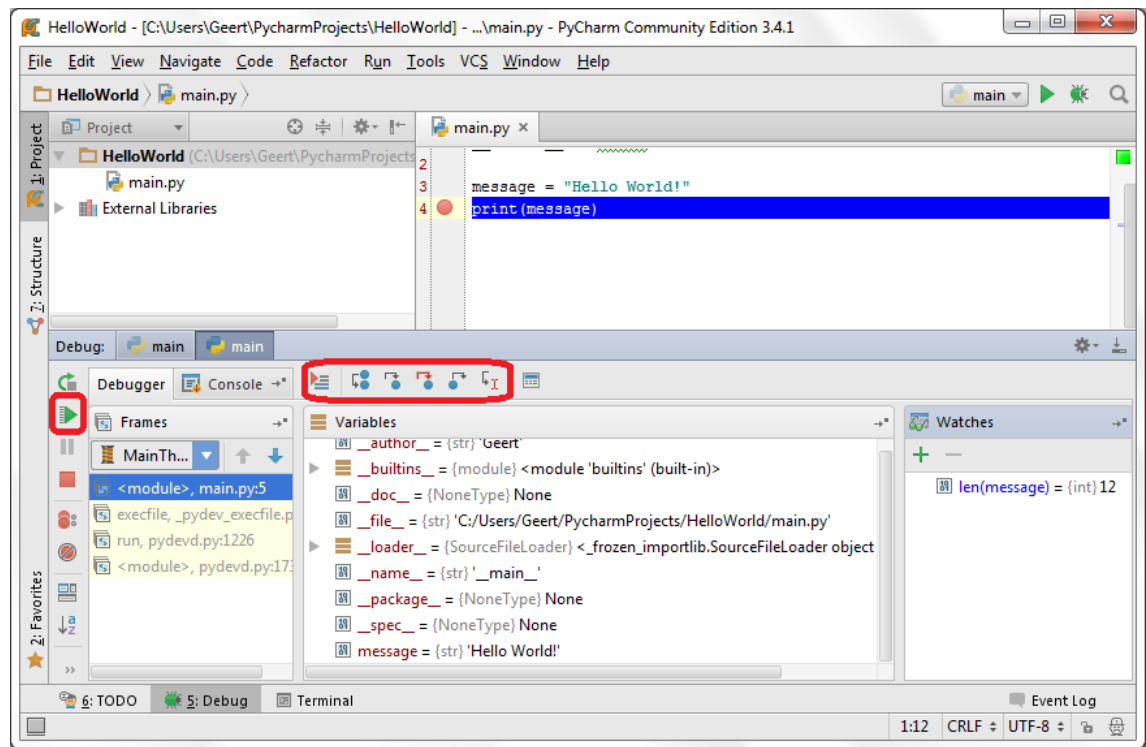
## 5 Graphics module

Het boek *Python for Everyone* definieert een eigen *graphics* module. Door gebruik te maken van de module kan je een grafische interface maken voor je programma zonder dat je zelf pixel per pixel moet definiëren.

De installatie verloopt als volgt:

1. Surf naar <http://bcs.wiley.com/he-bcs/Books?action=resource&bcsId=8084&itemId=1118626133&resourceId=31810> en download de graphics module.
2. Kopieer de module naar een plaats die Python weet te vinden. Wanneer je in je code naar een andere module refereert, zal Python deze module zoeken in een aantal vooraf bepaalde mappen. Om de graphics module te kunnen gebruiken, moet deze in één van deze mappen staan.

Door onderstaand stukje code uit te voeren kan je achterhalen in welke mappen Python zal zoeken voor jouw machine.



Figuur 2: De debugger in PyCharm.

```
import sys
print(sys.path)
```

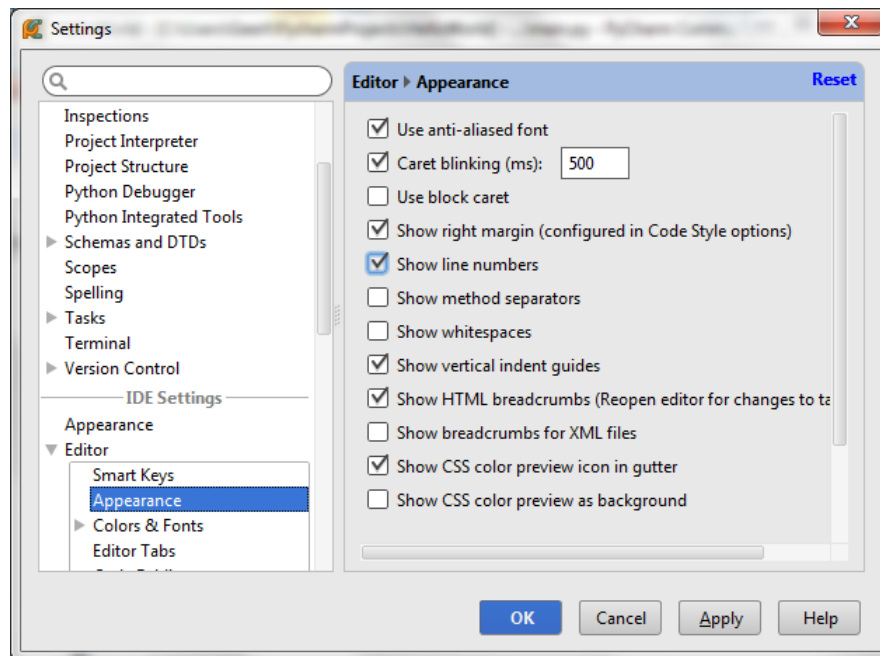
Als je de code uitvoert, zal een lijst met verschillen bestandslocaties geprint worden. Bv.:

```
'C:\\Users\\Geert\\PycharmProjects\\HelloWorld\\',
'C:\\Windows\\system32\\python34.zip',
'C:\\Python34\\DLLs', 'C:\\Python34\\lib',
'C:\\Python34', 'C:\\Python34\\lib\\site-packages'
```

De bestandslocatie `C:\\Users\\Geert\\PycharmProjects\\HelloWorld` is de plaats waar de code werd uitgevoerd (typisch de map van je project). Om ervoor te zorgen dat de externe module gedeeld kan worden over verschillende projecten, kopiëer je de module naar één van de *andere* mappen. `C:\\Python34\\lib` (of een gelijkaardige map) is een goede kandidaat. Wanneer de graphics module zich in deze map bevindt, is de installatie voltooid.

3. Test je installatie door onderstaande code uit te voeren:





Figuur 3: Instellingen wijzigingen in PyCharm.

```
from graphics import GraphicsWindow

window = GraphicsWindow(400,200)
canvas = window.canvas()
canvas.setBackground("red")
window.wait()
```

Met het eerste statement laat je Python weten dat je gebruik wil maken van *GraphicsWindow* uit de *graphics* module. Als alles goed gaat, wordt er een rood venster geopend wanneer je de code uitvoert.

## 6 Oefeningen

### 6.1 Tic-tac-toe grid

Schrijf een programma dat onderstaande grid afdruckt. Vermijd hierbij het gebruik van dergelijke statements: `print("+ - - + - +")`, maar ken de strings eerst toe aan variabelen.

Voorbeeld:

```
C:\Python34\python.exe C:/Users/Geert/PycharmProjects/Sessiel/tic-tac-toe_grid.py
+---+---+
|   |   |   |
+---+---+
|   |   |   |
+---+---+
|   |   |   |
+---+---+

Process finished with exit code 0
```

## 6.2 Briefpapier formaat omzetting : inches naar millimeter

Amerikaans briefpapier heeft een formaat van 8.5 x 11 inch. Schrijf een programma dat dit formaat in millimeter berekent. Eén inch is 25.4 millimeter. Maak gebruik van constanten.

Voorbeeld:

```
C:\Python34\python.exe C:/Users/Geert/PycharmProjects/Sessiel/letter_dimensions.py
USA letter dimension in millimeters: 215.9 x 279.4

Process finished with exit code 0
```

## 6.3 Begroeting

Schrijf een programma dat de voor- en achternaam van de gebruiker vraagt en vervolgens een begroeting voor de gebruiker print.

Voorbeeld:

```
C:\Python34\python.exe C:/Users/Geert/PycharmProjects/Sessiel/greeter.py
Enter your first name: Marc
Enter your last name: Peeters
Welcome Marc Peeters!
```

## 6.4 Inwendig product

Schrijf een programma dat het inwendig product tussen vectoren in twee dimensies berekent. Het inwendig product tussen twee vectoren  $\mathbf{u} = (x_1, y_1)$  en  $\mathbf{v} = (x_2, y_2)$  is gedefinieerd als volgt:

$$\mathbf{u} \cdot \mathbf{v} = x_1x_2 + y_1y_2$$

Voorbeeld:

```
C:\Python34\python.exe C:/Users/Geert/PycharmProjects/Sessie1/inner_product.py
Enter x-coordinate of the first vector: 2.2
Enter y-coordinate of the first vector: 3.3
Enter x-coordinate of the second vector: 2
Enter y-coordinate of the second vector: 1
Inner product is: 7.7

Process finished with exit code 0
```

## 6.5 Omzetting van seconden

Schrijf een programma dat de gebruiker vraagt een aantal seconden in te geven en deze seconden vervolgens omzet naar het formaat: aantal dagen, aantal uren, aantal minuten, aantal seconden.

**Hint:** integer deling (//) en de modulo operator (%) komen hierbij van pas.

Voorbeeld:

```
C:\Python34\python.exe C:/Users/Geert/PycharmProjects/Sessie1/seconds_converter.py
Time in seconds: 86491
1 days, 0 hours, 1 minutes, 31 seconds

Process finished with exit code 0
```

## 6.6 Carthesische coördinaten naar polaire coördinaten

Schrijf een programma dat carthesische coördinaten  $(x, y)$  naar polaire coördinaten  $(r, \theta)$  omzet. De omzettingformules hiervoor zijn:

$$r = \sqrt{x^2 + y^2}$$

$$\theta = \tan^{-1}(x/y)$$

**Hint:** je zal voor je programma gebruik moeten maken van voorgedefinieerde functies in de *math* module. Voor  $\tan^{-1}$  moet je de functie *atan* gebruiken.

Voorbeeld:

```
C:\Python34\python.exe C:/Users/Geert/PycharmProjects/Sessie1/cartesian_to_polar.py
Enter the x-coordinate: 2.5
Enter the y-coordinate: 2.5
The radius is: 3.5355339059327378
Theta is: 45.0

Process finished with exit code 0
```

## 7 Extra oefeningen uit het boek

Hier zijn nog enkele extra oefeningen uit het boek:

P2.2, P2.3, P2.7, P2.8, P.11, P2.16, P.17, P.18, P2.20, P.21