

GRN

20-1-2019

Documentatie Informatica

Douwe, Niek, Benjamin, Andries V4

# Inhoudsopgave

Inhoud

[Inhoudsopgave 1](#_Toc535760256)

[Documentatie 2](#_Toc535760257)

[20 december 2018 2](#_Toc535760258)

[27 december 2018 2](#_Toc535760259)

[10 januari 2019 2](#_Toc535760260)

[27 januari 2019 2](#_Toc535760261)

# Docu

## 20 december 2018

* Douwe en Andries maken de code Pizza
* Douwe en Andries maken de code Trapeze
* Benjamin en Niek maken de code Letterbeeld

## Week 51

* Douwe, Benjamin, Niek en Andries verfijnen de code Pizza en beginnen aan de code Proefwerk
* Niek maakt de codes Namenlijst, Driehoek van pascal, Statistiek, Perfect getal en Driehoek (vrijwillig)

## 10 januari 2019

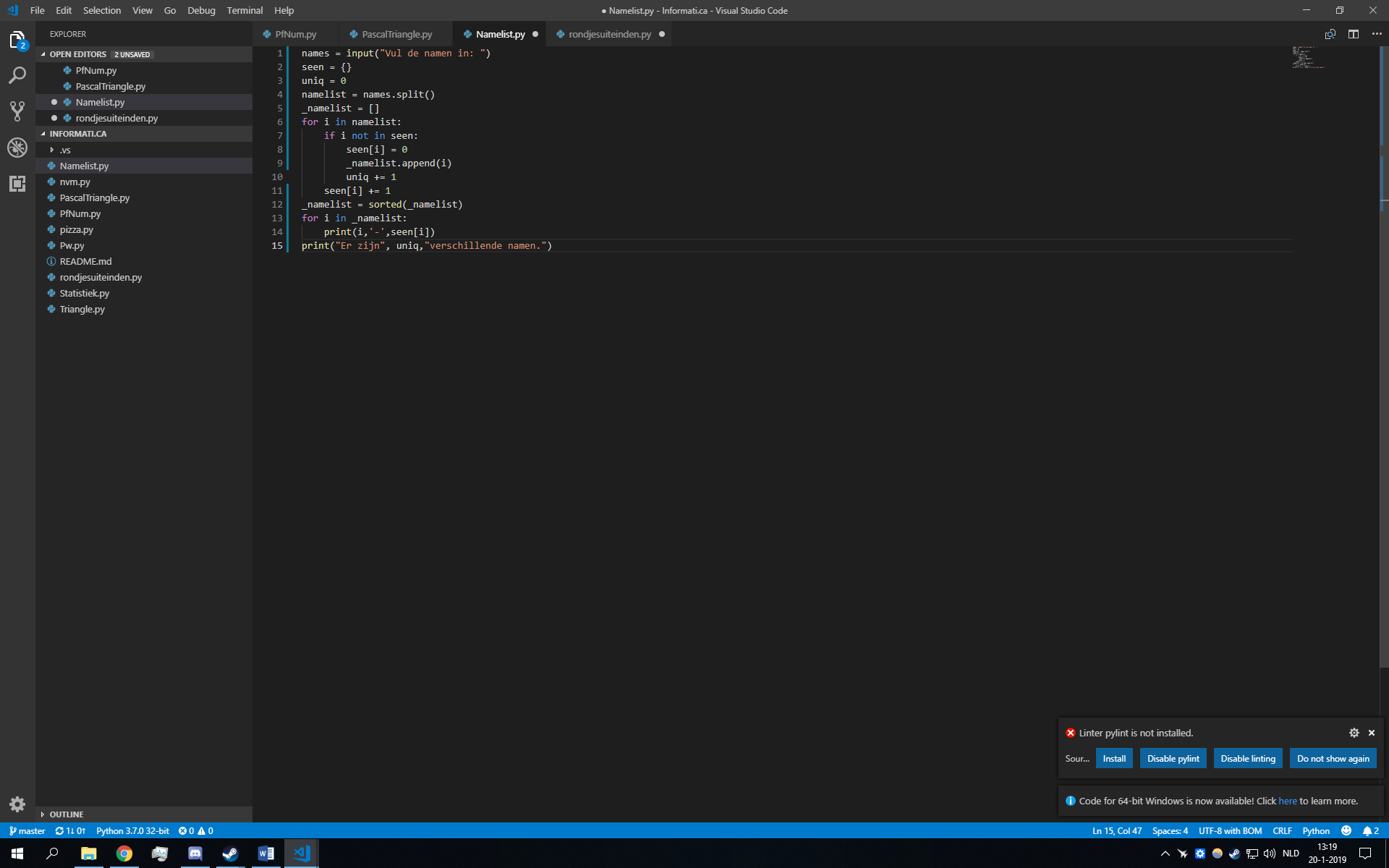
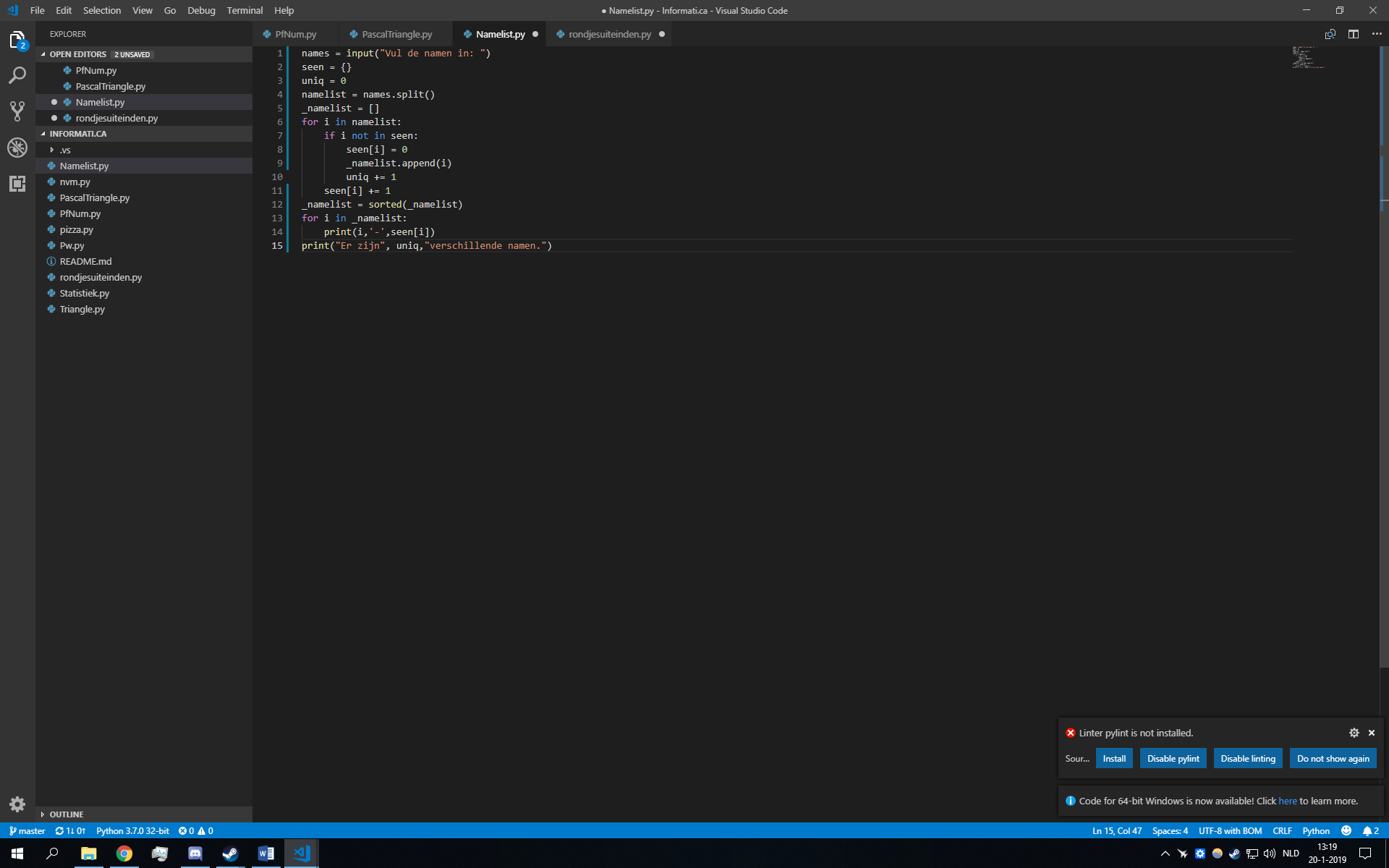
* Douwe en Niek overleggen over de vertaling van de comments in de code (en ook sommige variabelen)
* Benjamin bedacht overige vragen van Proefwerk en in de code gezet met Douwe

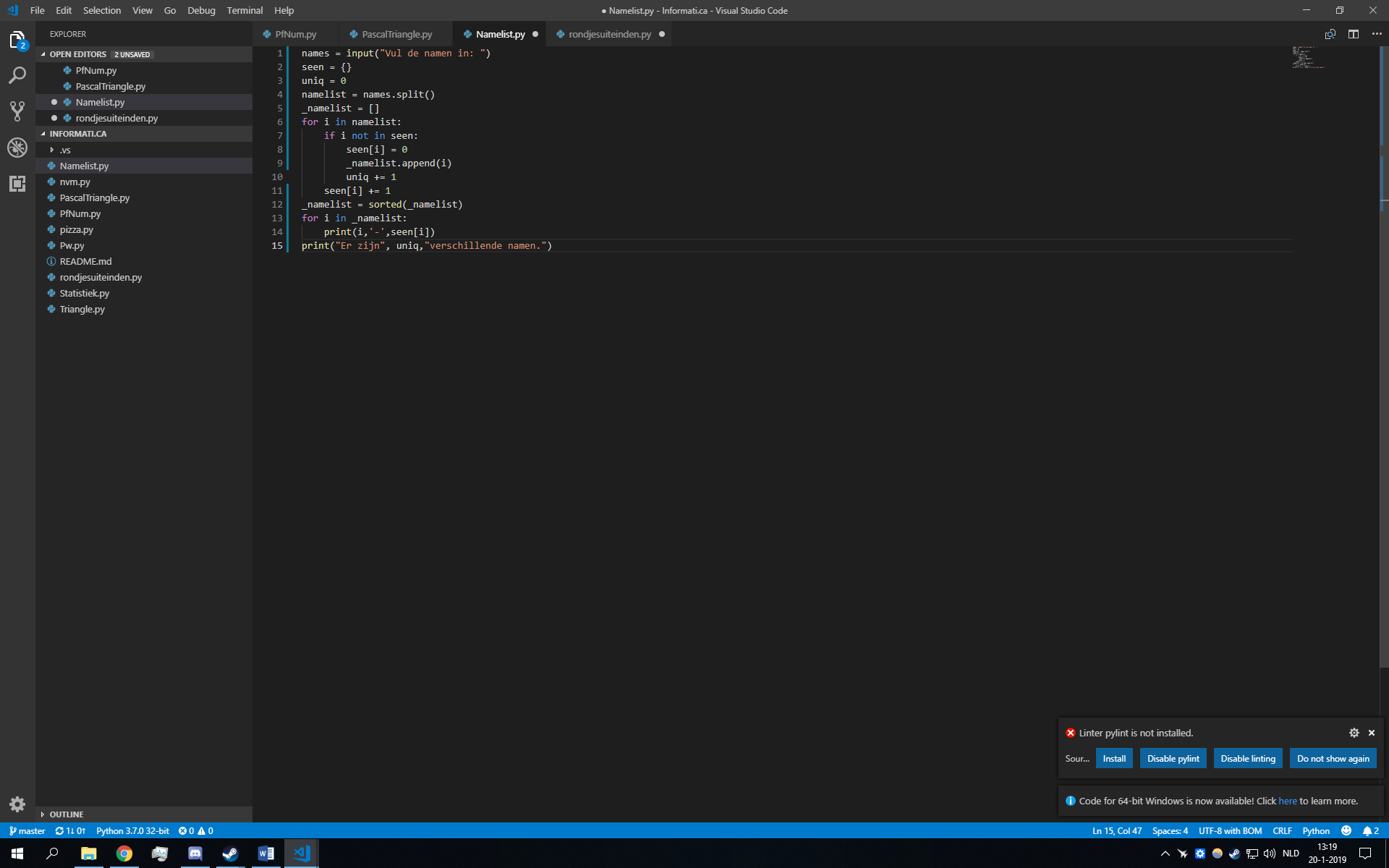
## 20 januari 2019

* Douwe en Niek maken de laatste ‘finishing touches’ aan de codes
* Samen met iedereen de documentatie gemaakt voor de codes
* Alle codes zijn te vinden [op deze website](http://www.github.com/niekaukes/informati.ca)

# Documentatie

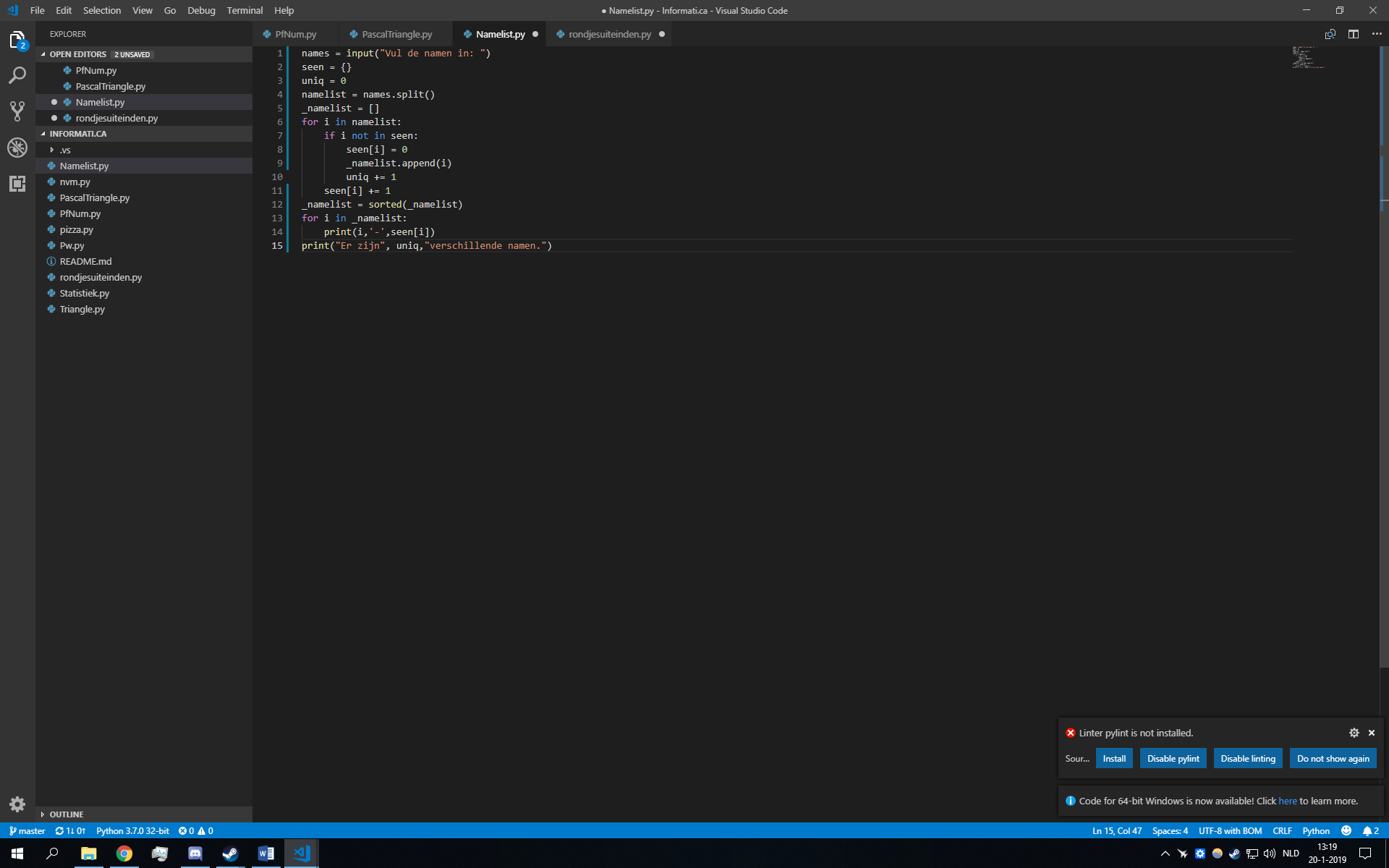
Code voor de namenlijst





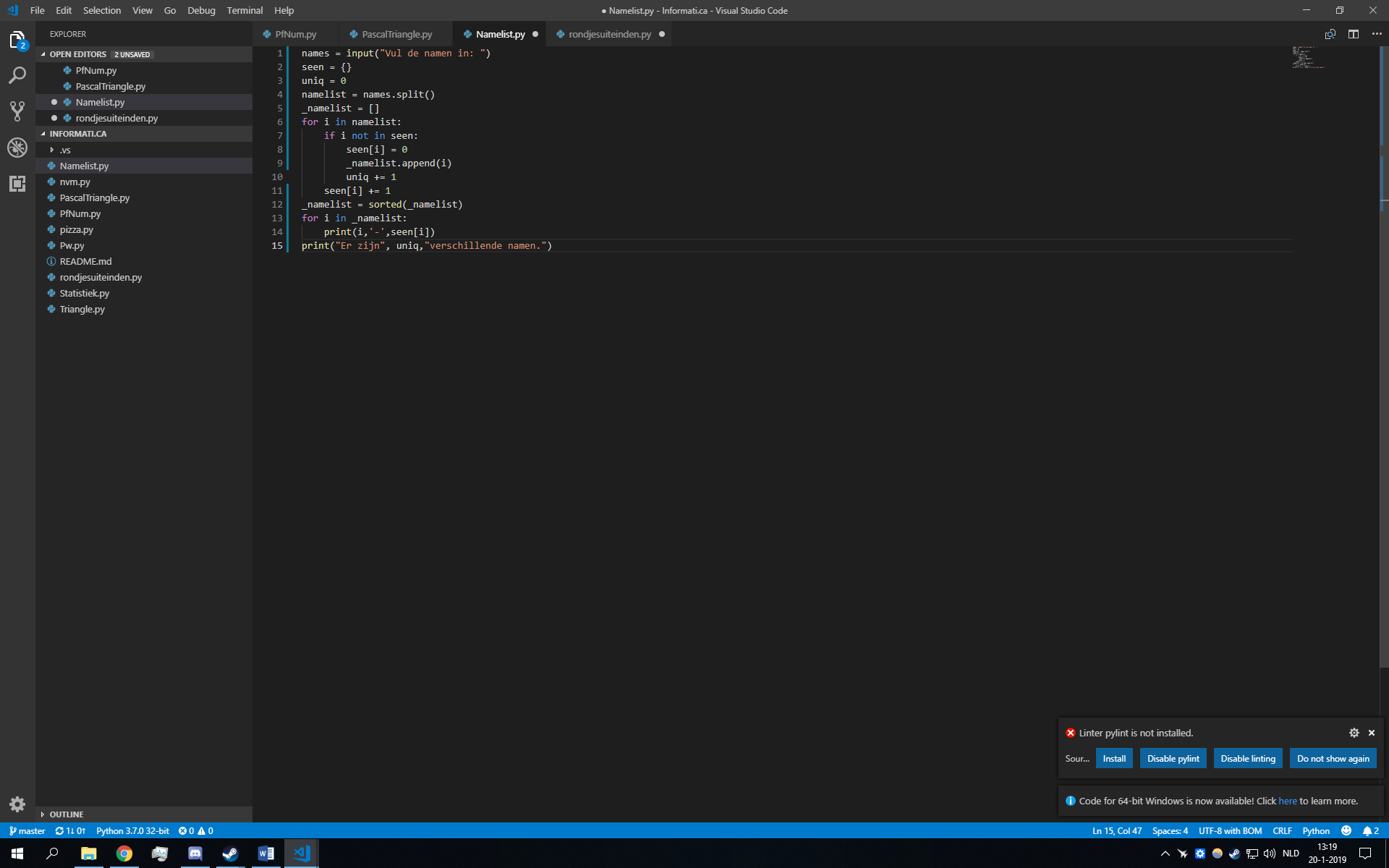
Deze regels maken variabelen aan voor de code

(r. 1-5)

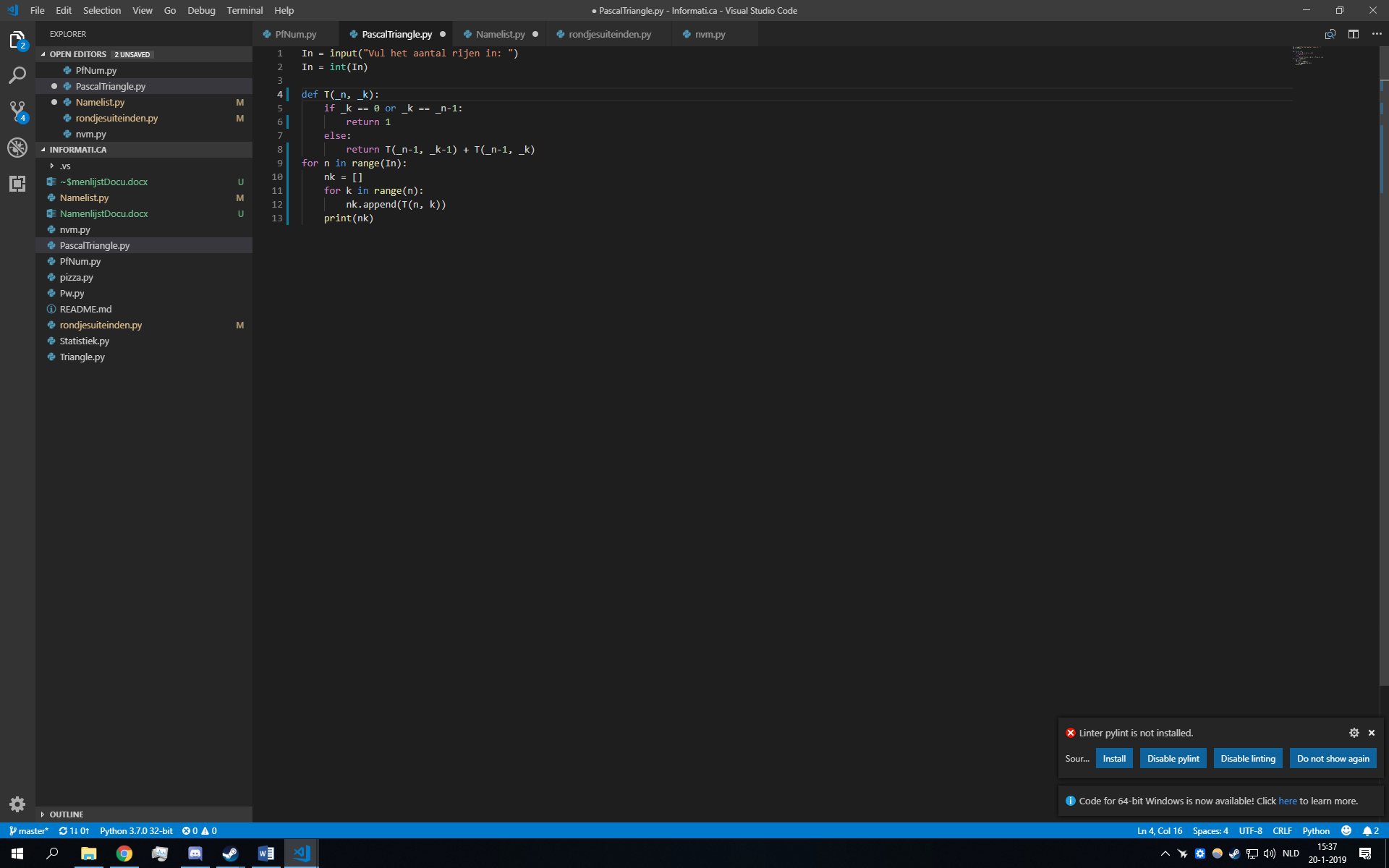


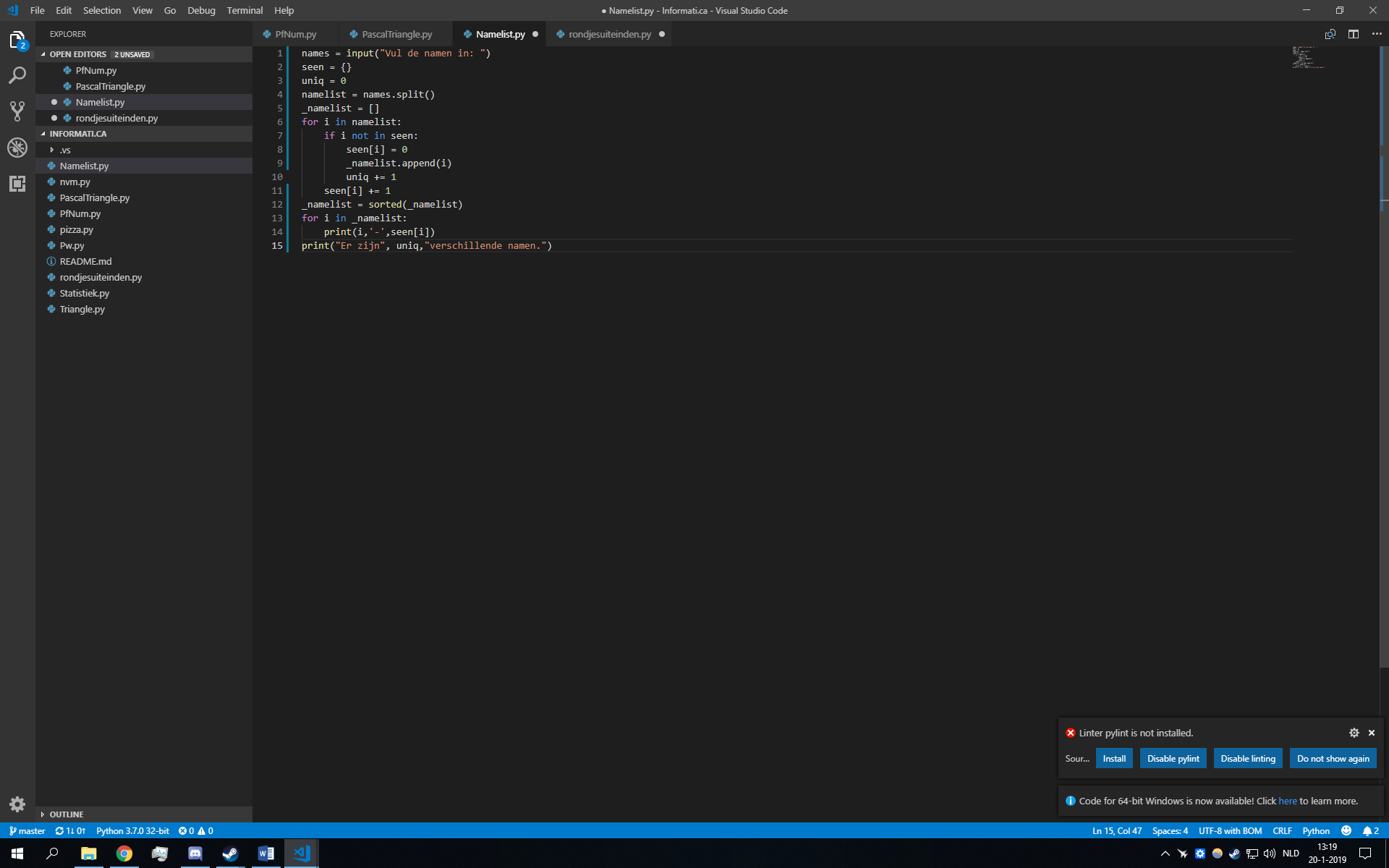
In deze regels zit een loop met namenlijst waarin i de momentele waarde is. Hierin wordt de dictionary seen gebruikt om duplicates uit de namenlijst te halen. Ook voegt hij het aantal keer gezien toe aan seen. Unieke waarden worden ook toegevoegd aan \_namelist, die later wordt gebruikt. (r. 6-11)

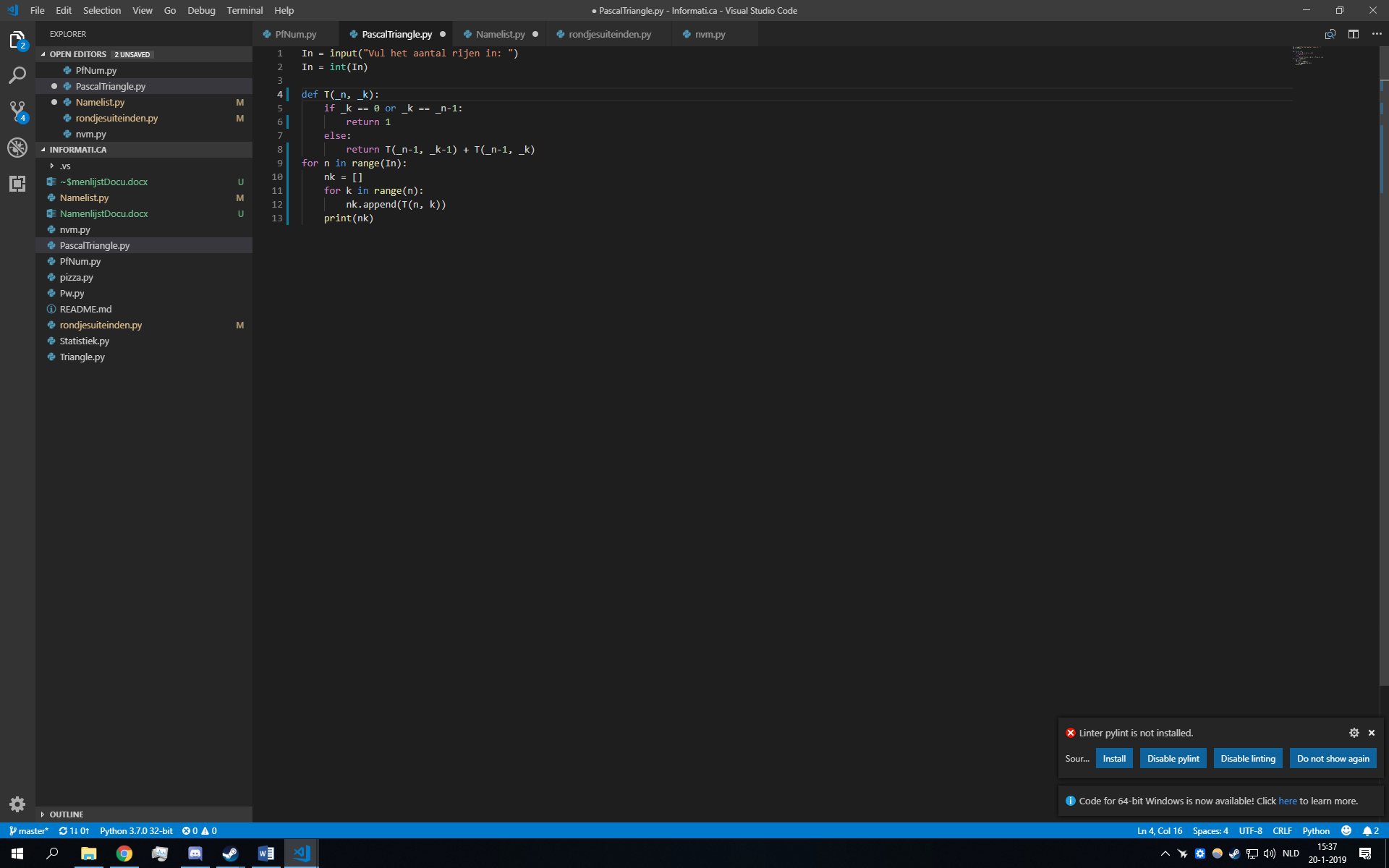
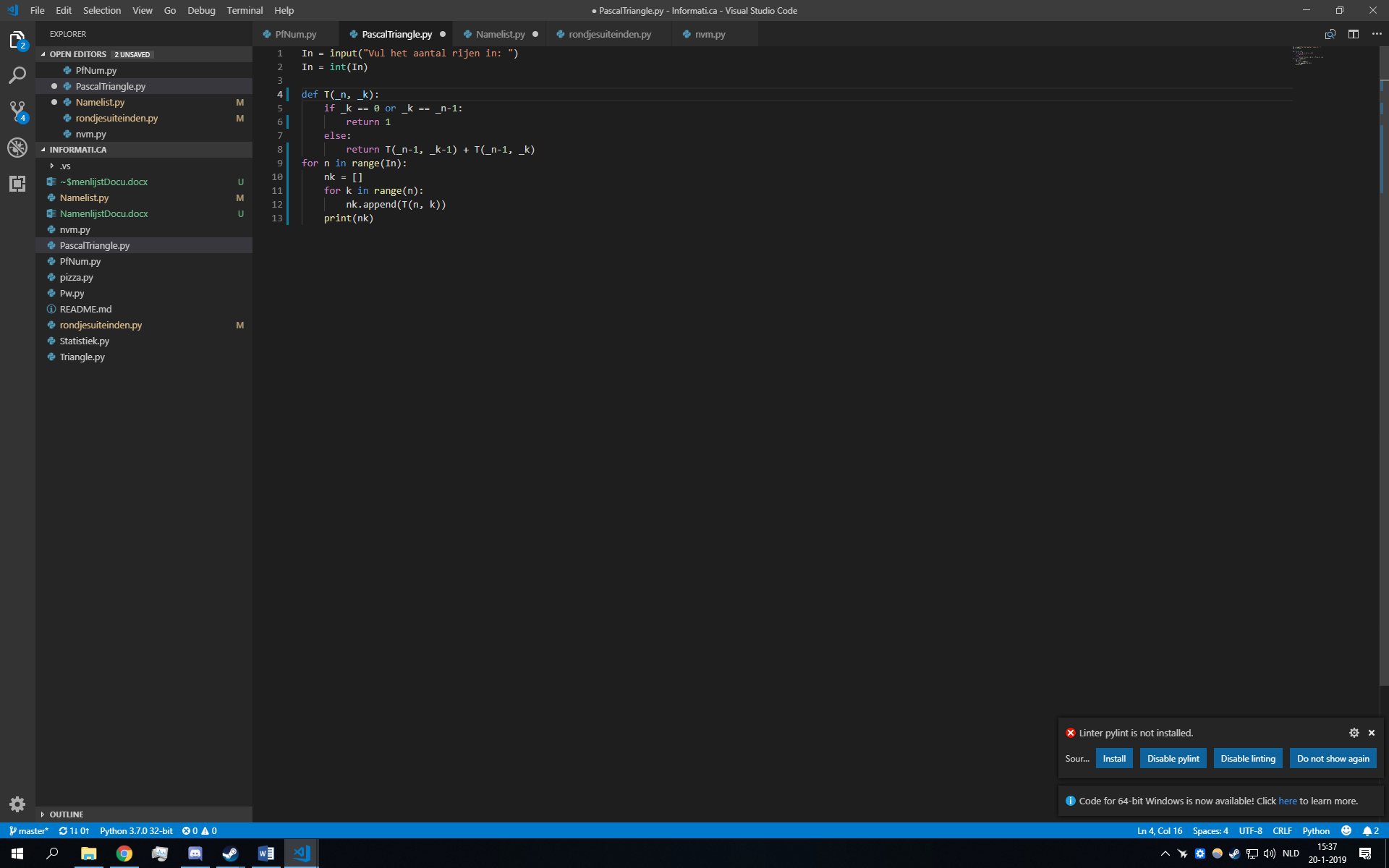
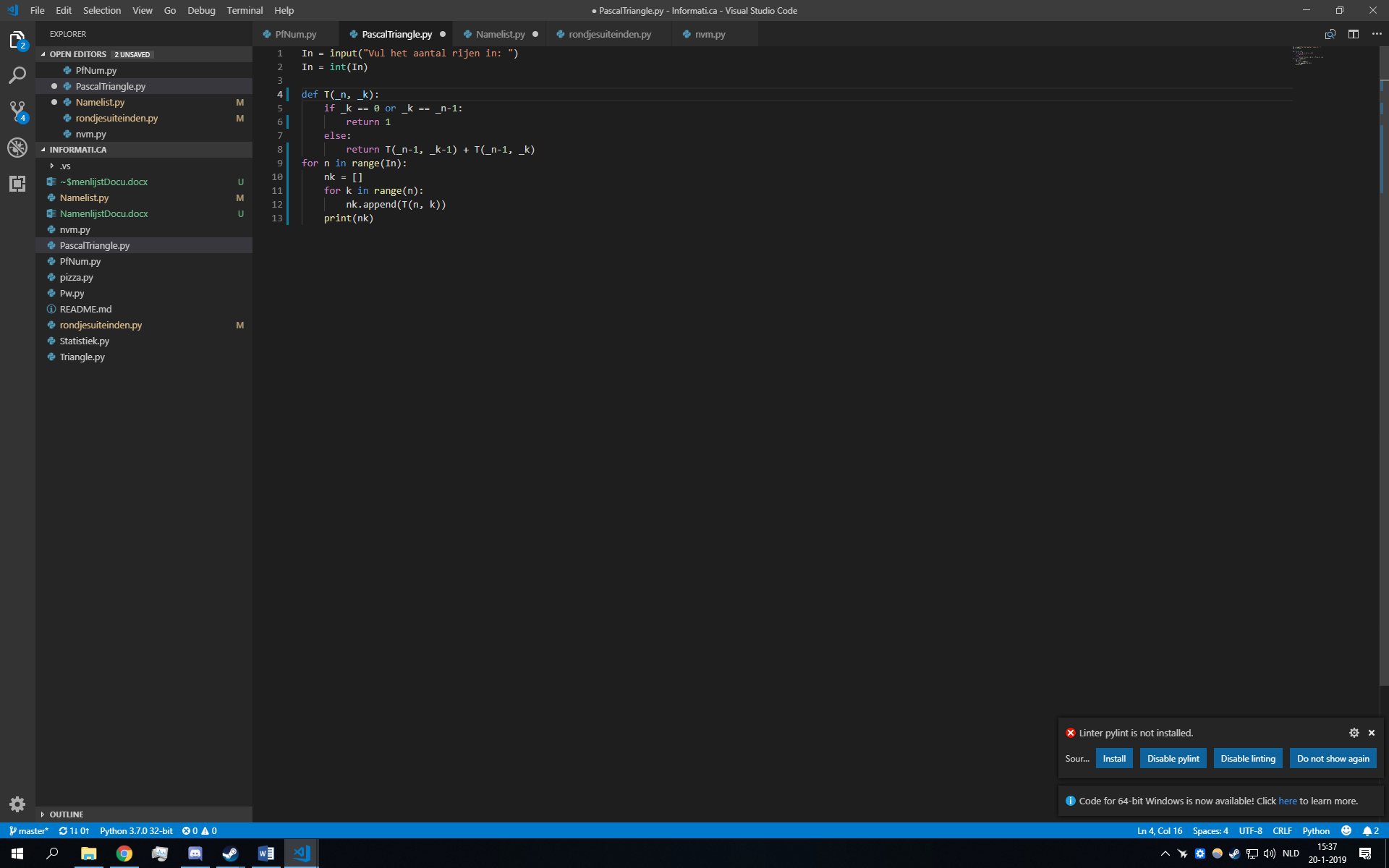
\_namelist wordt hier gesorteerd en als key gebruikt in seen. Hierbij print het programma de unieke namen en de frequentie van de namen. Deze techniek heb ik gebruikt omdat het ingewikkeld is om een dictionary normaal te sorteren. (r. 12-15)



Code voor de driehoek van pascal







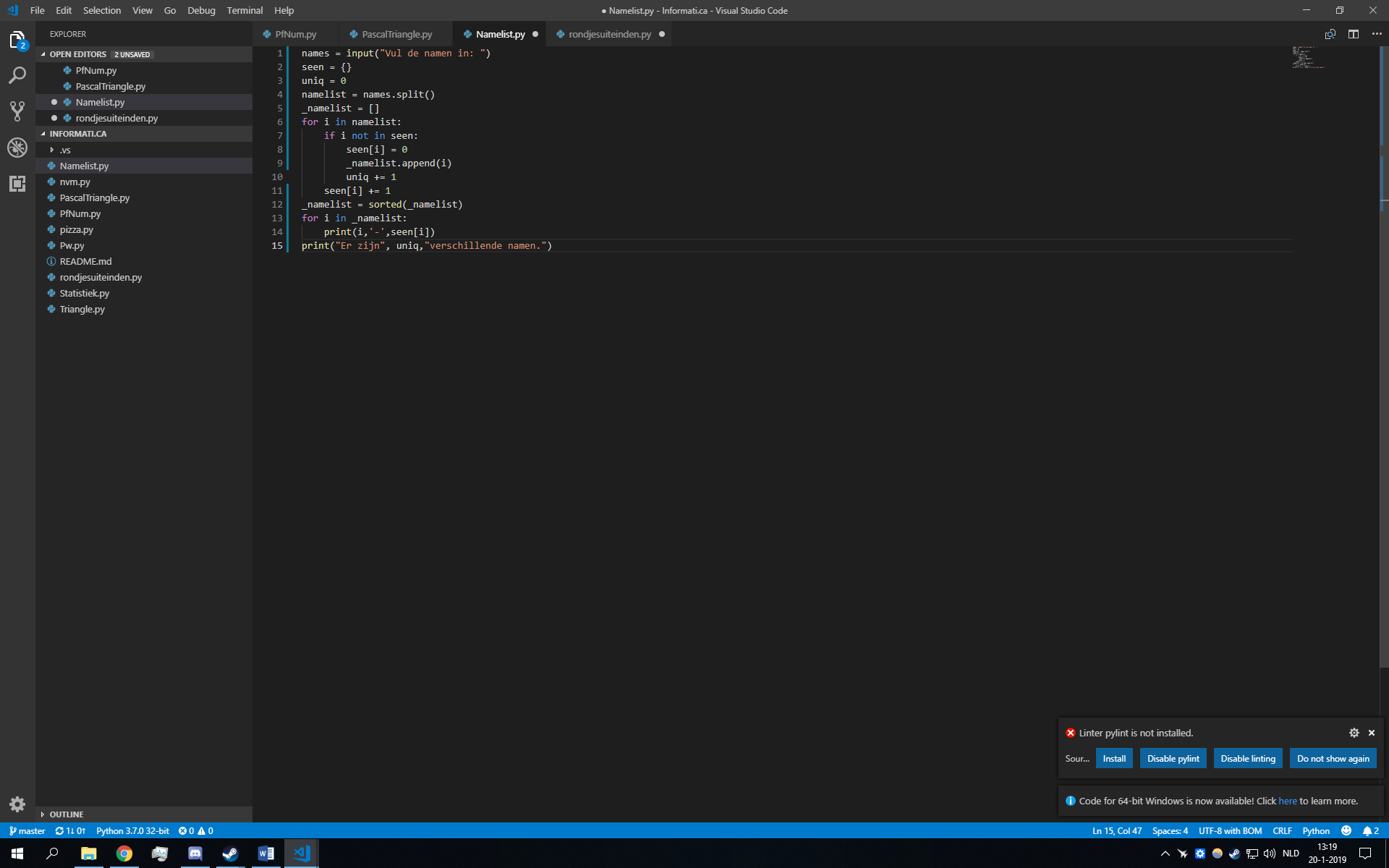
Deze regels maken variabelen aan voor de code

(r. 1-5)

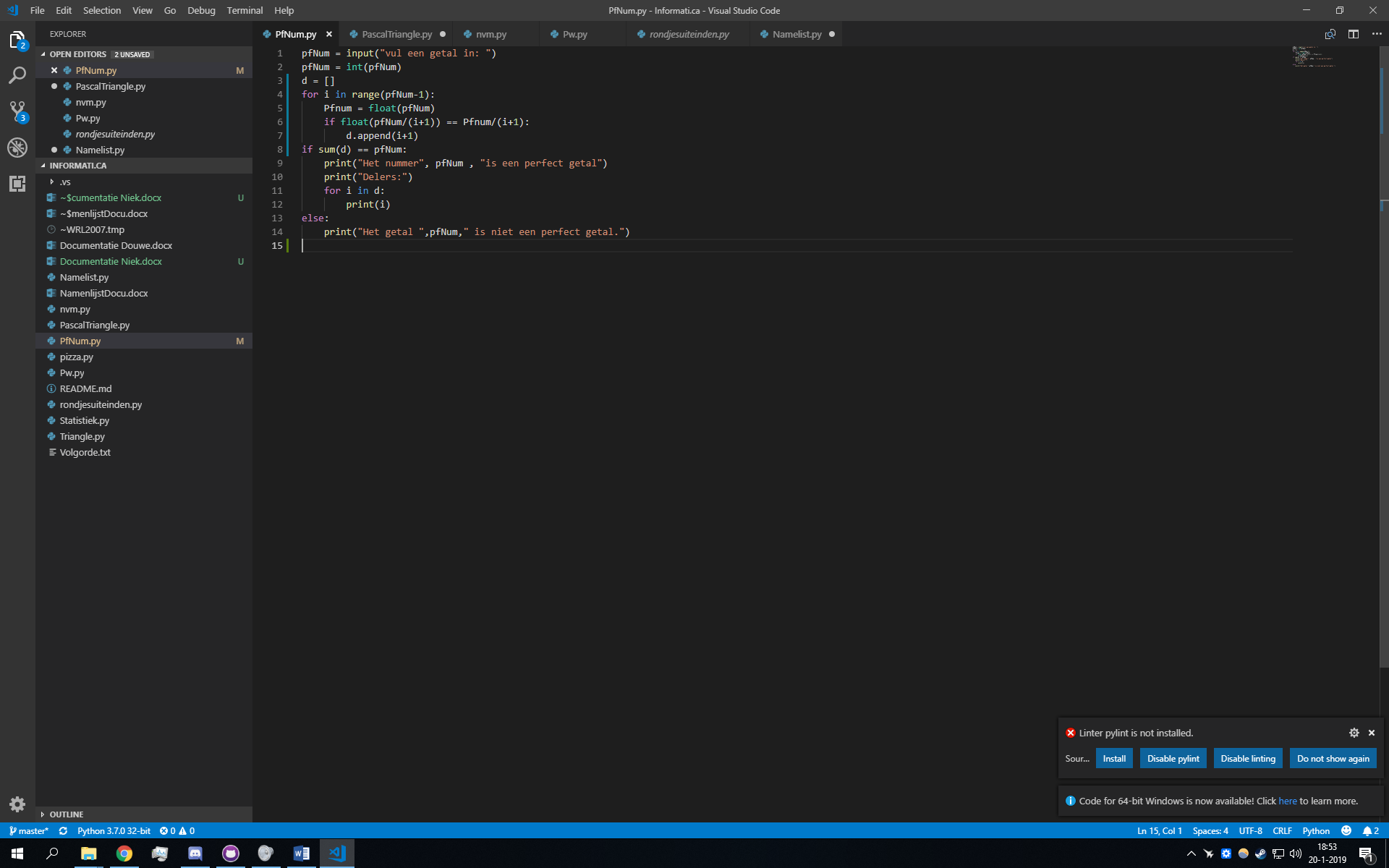
Dit is de recursie method. Deze wordt gebruikt om de positie van het cijfer in de driehoek te bepalen.

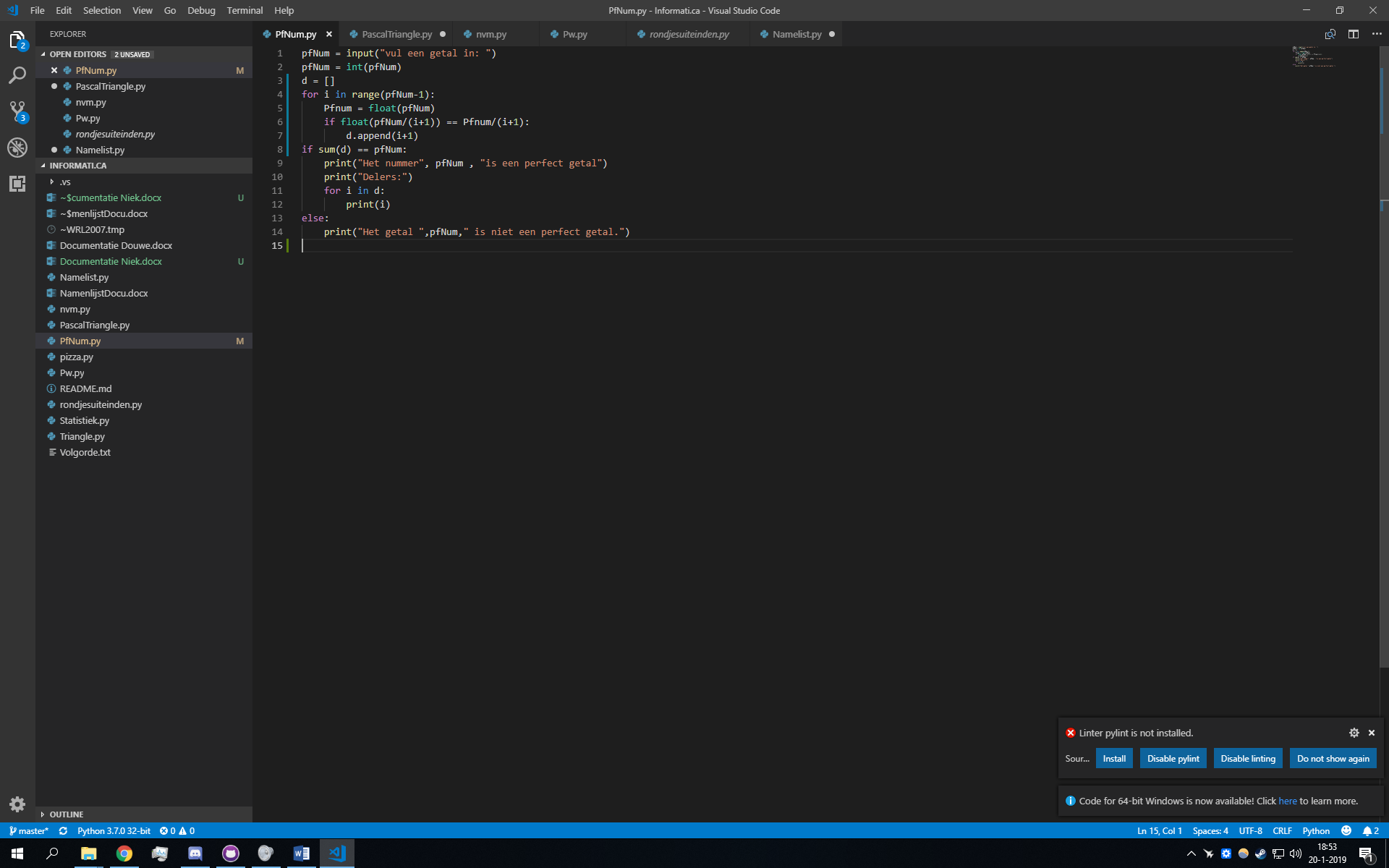
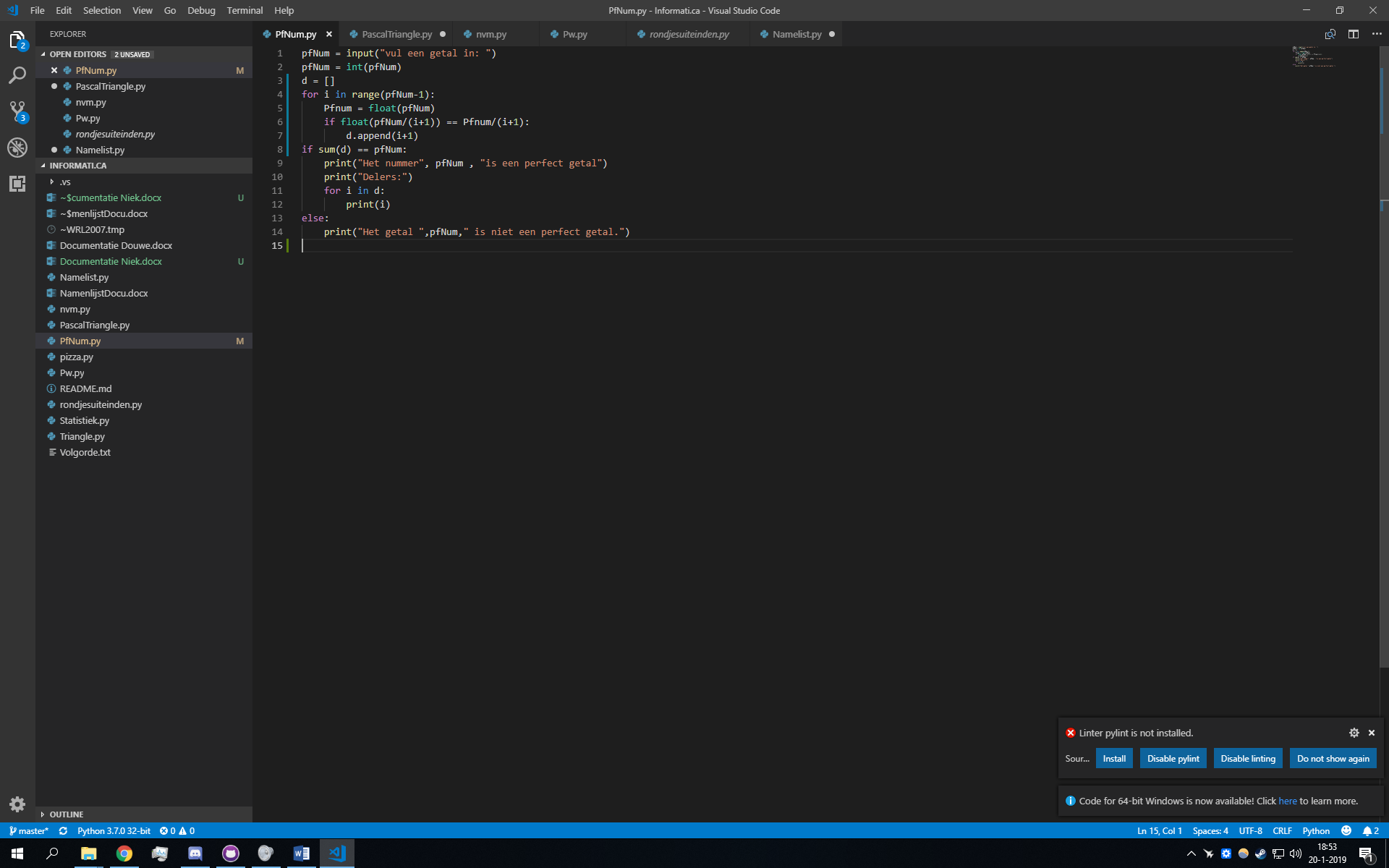
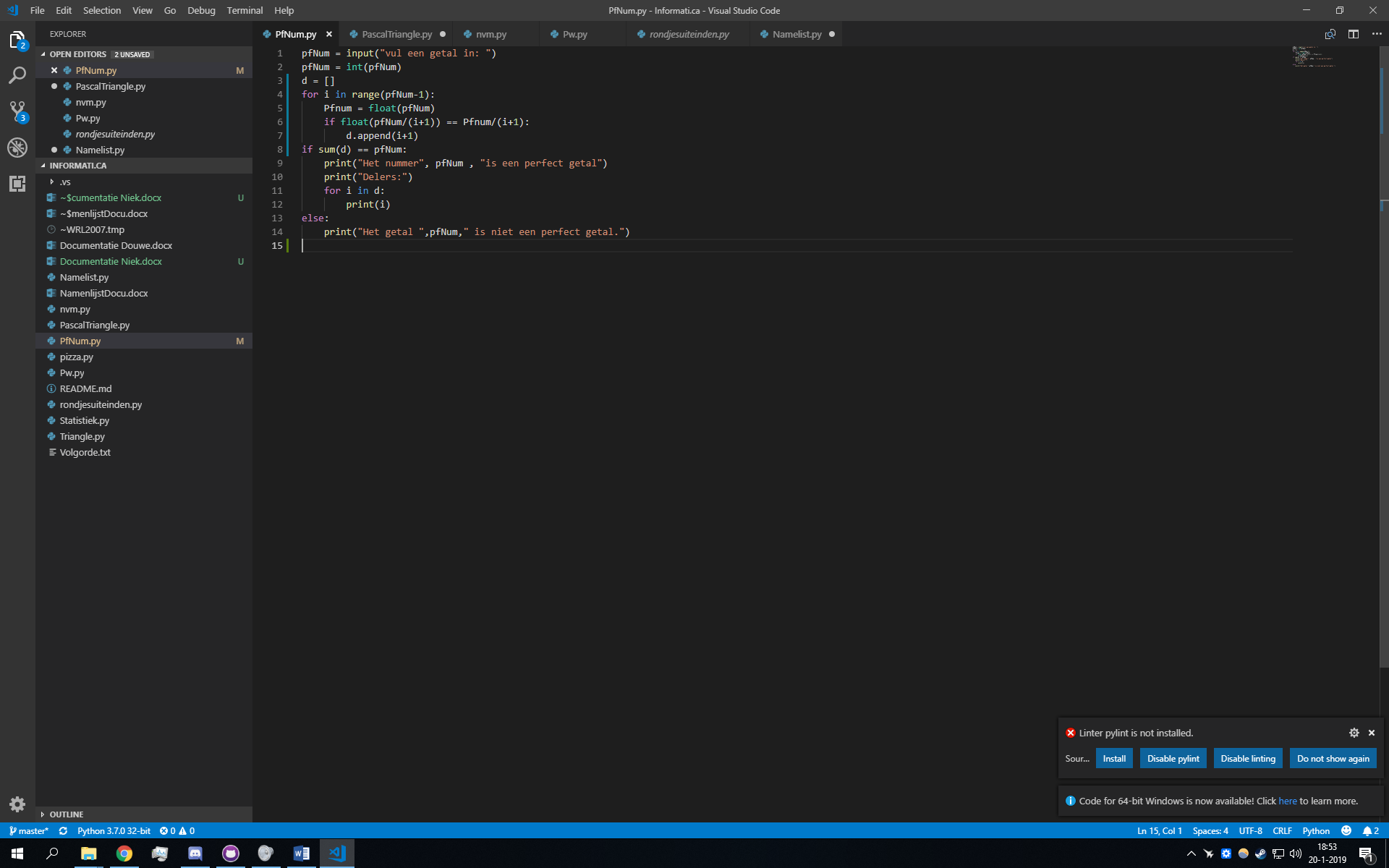
Als het cijfer zich aan de rand van de drie bevindt returnt de functie 1.

In deze regels worden de cijfers berekend door alle cijfers per rij te berekenen



Code voor het Perfect getal

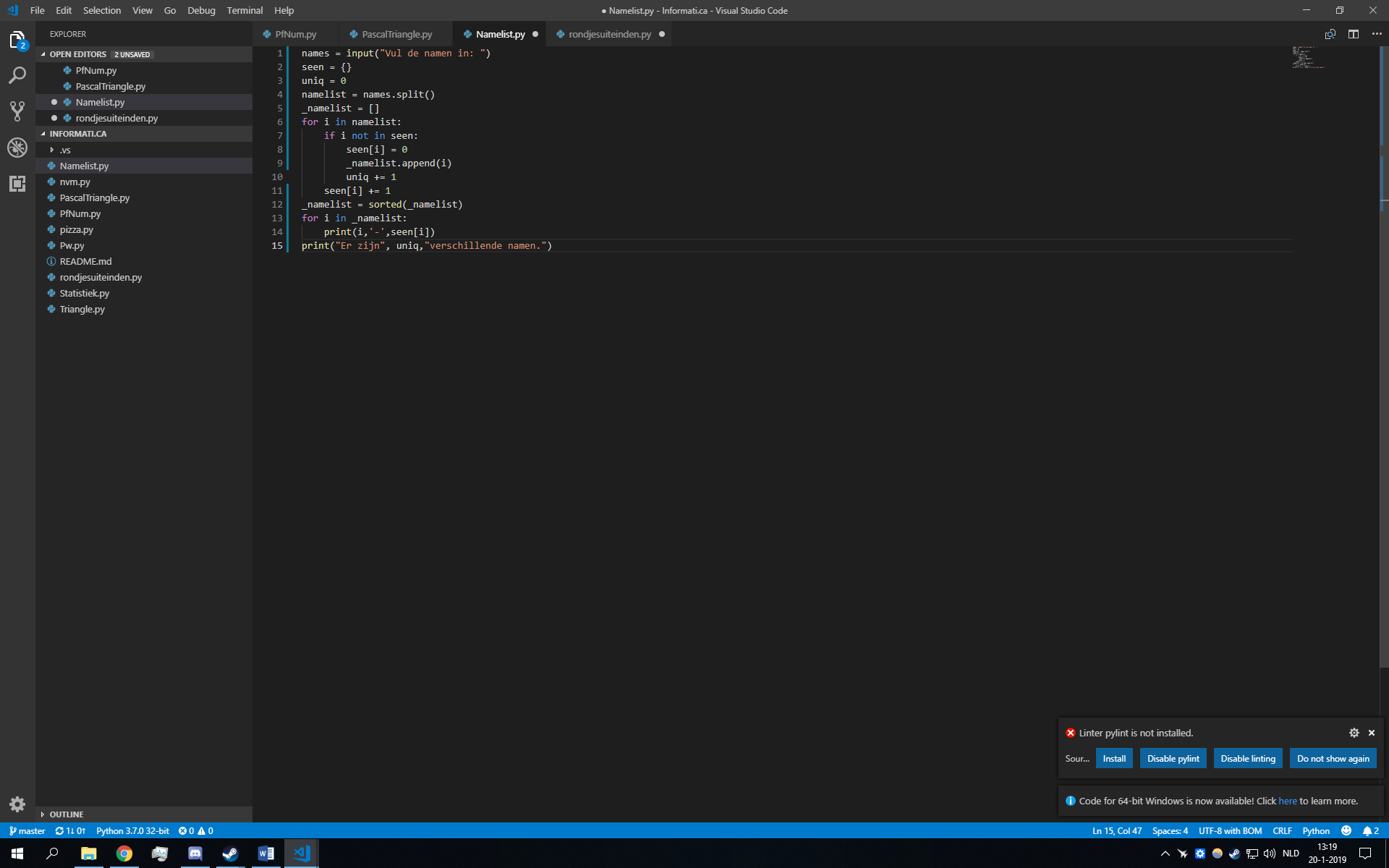




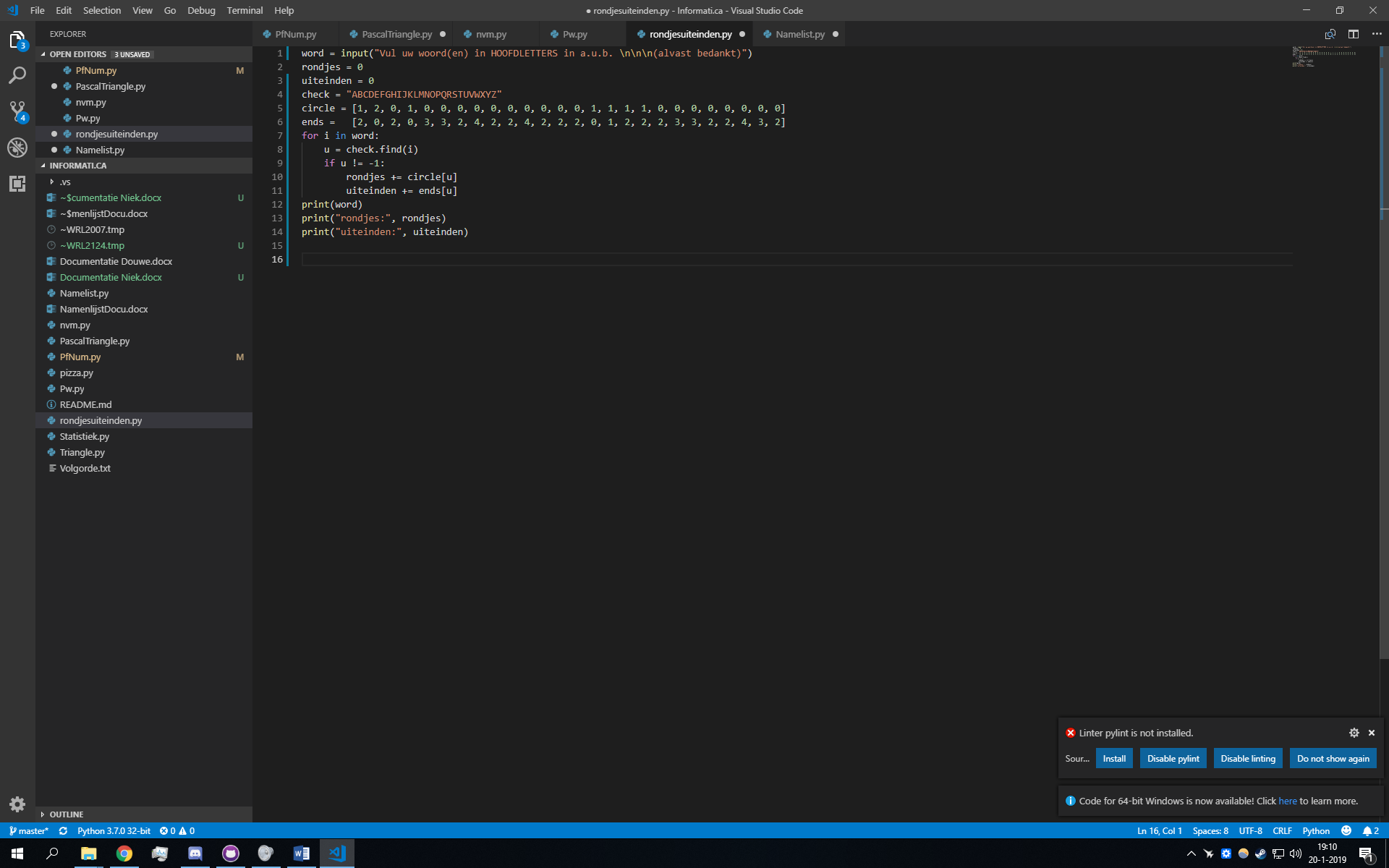
Deze regels zoeken alle hele delers tot aan het nummer. Daarna kijkt hij of het getal gedeeld door de deler een heel getal is. Daarna voegt hij dit getal toe aan de lijst met delers(d). (r. 4-7)

Deze regels maken variabelen aan voor de code (r. 1-3)

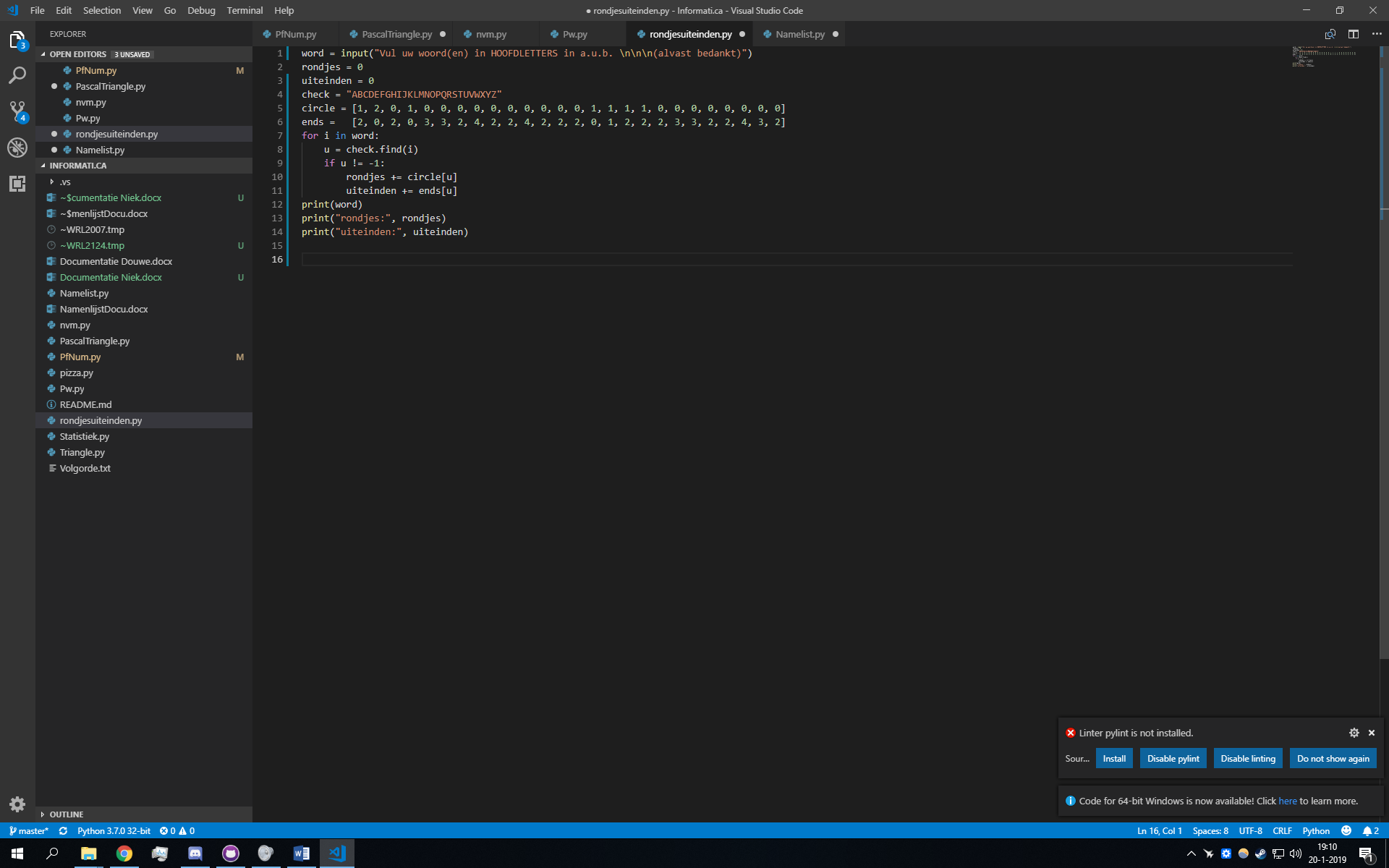
Deze regels kijken of de som van de delers gelijk is aan het getal. Als dat zo is print hij ook alle delers. (r. 8-14)

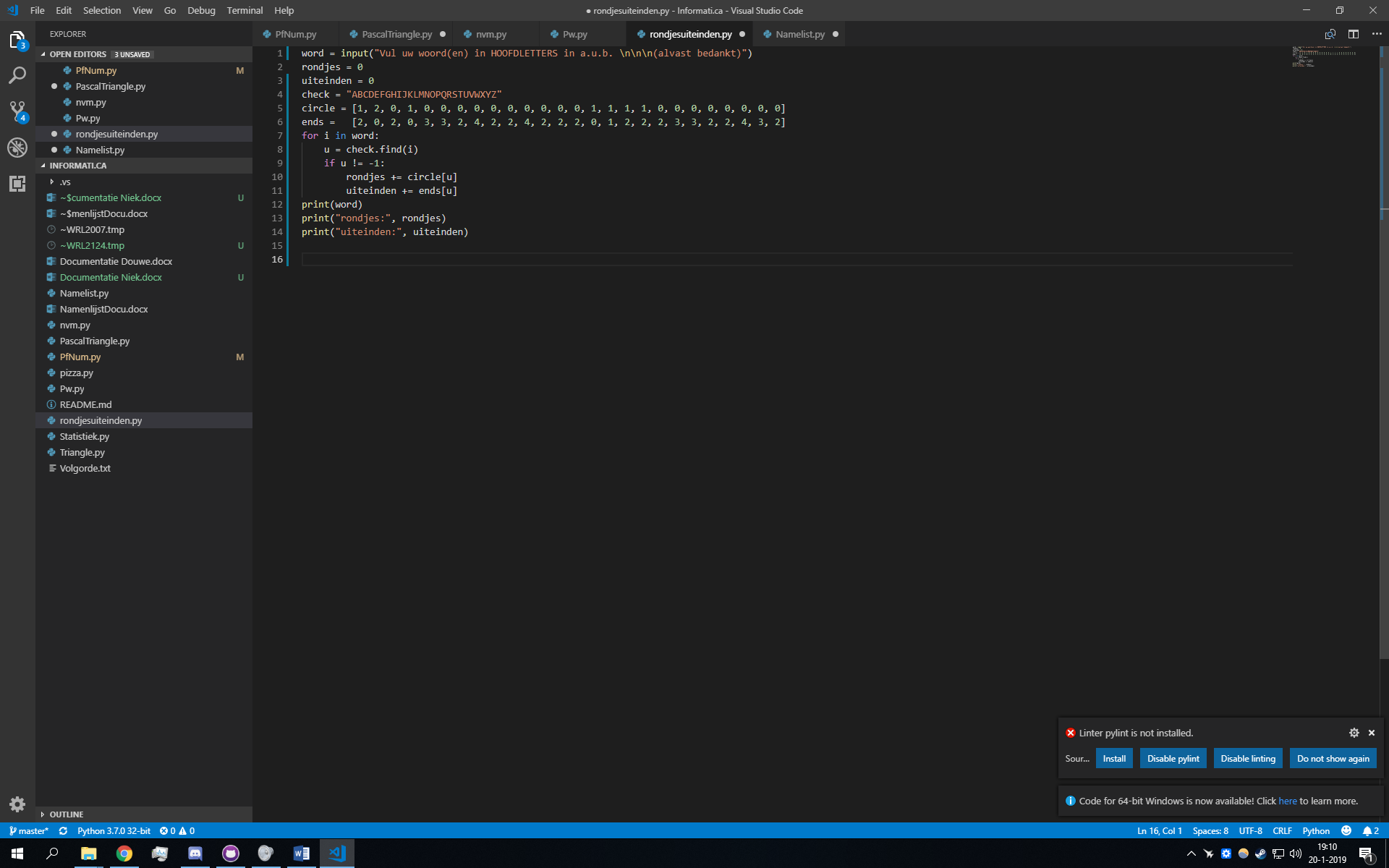


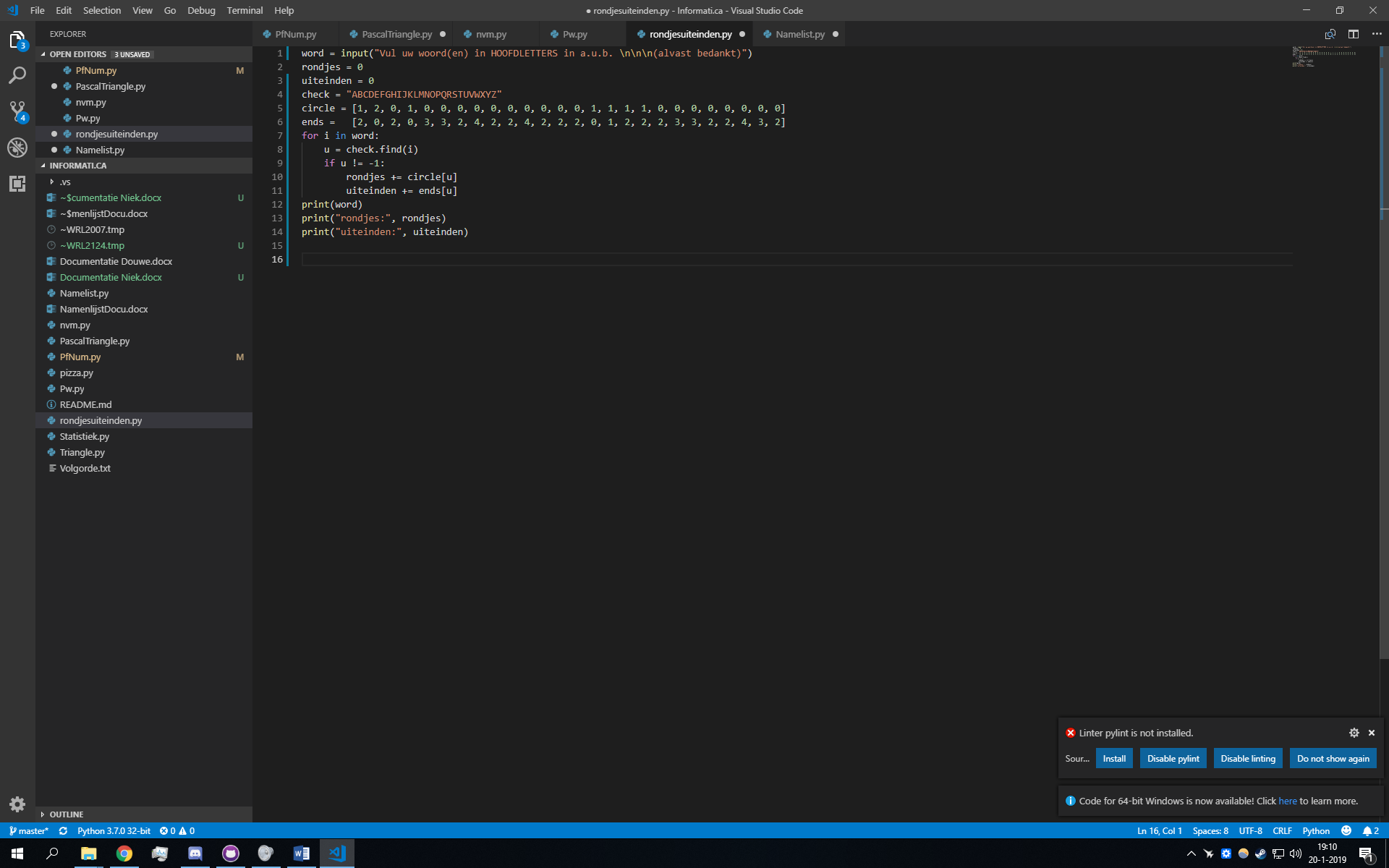
Code voor het Letterbeeld



Deze regels maken variabelen aan voor de code (r. 1-6)



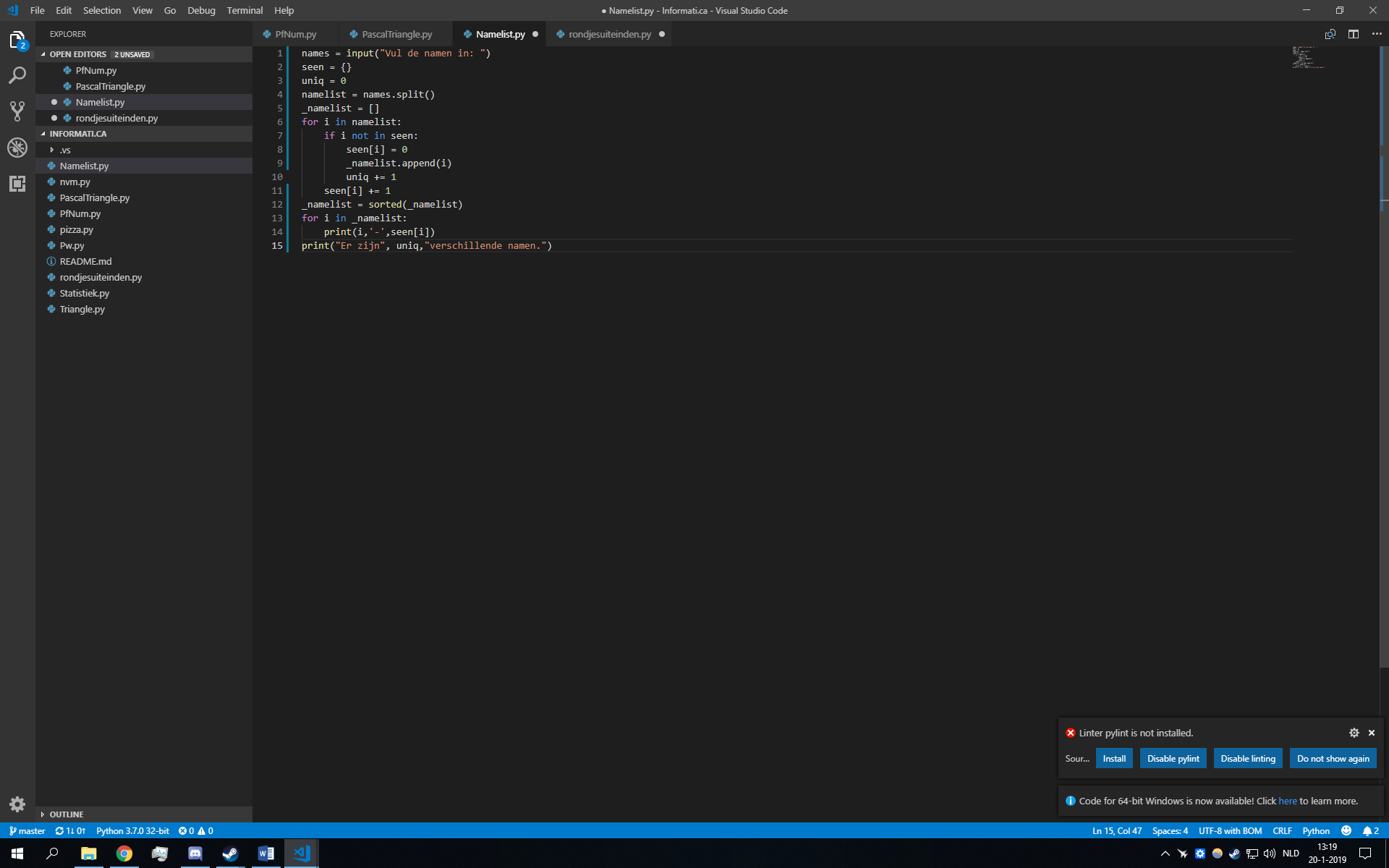




Deze regels zoekt voor elke char in het woord het indexnummer in het alfabet.

Dit indexnummer kunnen we dan gebruiken in de lijsten circle en ends. Wat zich optelt in de variabelen rondjes en uiteinden

Deze regels weergeven het woord, de rondjes en de uiteinden



Code voor het Trapezium

