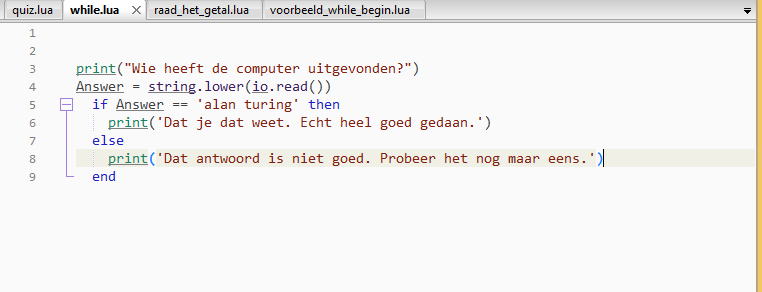
# While lus

Weet je nog de opdracht met de quiz:   
Neem onderstaande code maar eens over,



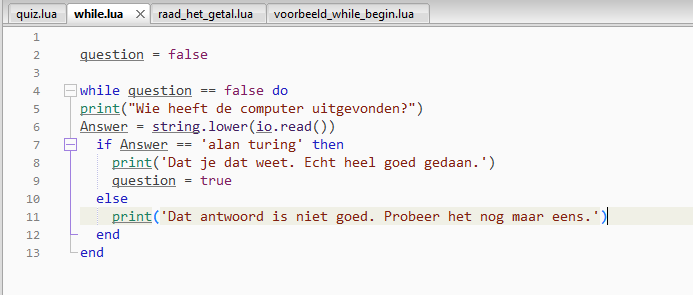
Wat valt je op: inderdaad je kunt de vraag niet nog eens stellen.

Zet er nu nogmaals een Answer = tussen.



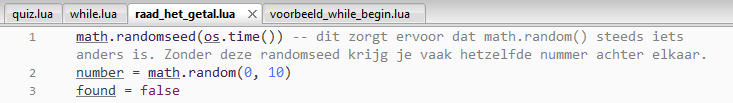
Je ziet dat je nu twee maal het antwoord kunt invullen.  
Maar ik wil dat je net zolang antwoord kunt geven, totdat het juiste is gegeven.

Dat gaan we met een WHILE doen. Neem onderstaande code over:

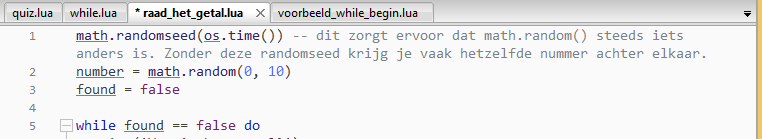


Je ziet dat er een varaibel ‘question’ is toegevoegd die ‘False’ is.  
Bij de WHILE lus wordt er gekeken of ‘question’ nog false is, zolang dit waar is, wordt de lus uitgevoerd. Zodra het goede antwoord wordt gegeven, wordt de varaibel ‘question’ op ‘TRUE’ gezet, en stopt de while.

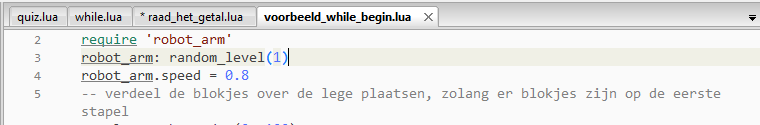
## Opdracht 1 Neem onderstaande code over:



Hier wordt een variable ‘number’ gevuld met een willekeurig getal tussen 0 en 10.  
Maak nu een WHILE lus waarin ik het getal kan raden. Zodra ik dit geraden heb, stopt de vraag.



## Opdracht 2

Neem onderstaande code over:   


Op regel 3 wordt er een random level geladen, met maar 1 kolom maar een willekeurig aantal blokjes.

Zorg nu dat je robot arm alle blokjes verdeeld over de lege plaatsen, maar zodra er geen blokjes meer zijn moet de arm stoppen. (Hint: dan is de kleur die de arm scant ‘nil’)