



Practicum vrijdag

Collapse context

Om goed te leren programmeren moet je veel oefenen. Deze tool zal je begeleiden terwijl je oefent. Het idee is eenvoudig, per probleem zal je code kunnen schrijven in de daarvoor voorziene vakken. Heb je een probleem opgelost dan kan je onderaan op "submit" klikken, vervolgens wordt je volledige inzending nagekeken. Je krijgt per probleem feedback of deze oplossing correct was of welke problemen er voor kwamen tijdens het uitvoeren en nakijken van je code.

Belangrijk om rekening mee te houden tijdens het gebruik van deze automatische tool is dat gevraagde output exact dezelfde vorm moet hebben om goedgekeurd te worden. Let dus altijd heel goed op wat exact gevraagd wordt. Indien je output niet correct was zal je steeds een indicatie krijgen van wat de verwachte output was. Op deze manier kan je je code verbeteren.

Deze eerste reeks van oefeningen is bedoeld om de basiselementen die we reeds gezien hebben in te oefenen, en om te leren werken met deze tool. Lees aandachtig alle stappen en los de gegeven problemen op!

Your answer passed the tests! Your score is 100.0%. [Submission #66f671dd8aa19d951e4d47b2 (2024-09-27 08:50:37)]

Question 1: We like Python! - deel 1

✓ Perfect

Schrijf een programma dat `We like Python!` 5 keer, onder elkaar, afprint zonder gebruik te maken van een loop.

```
1 print("We like Python!")
2 print("We like Python!")
3 print("We like Python!")
4 print("We like Python!")
5 print("We like Python!")
```

Submit

Question 2: We like Python! - deel 2

✓ Perfect

Schrijf een programma dat `We like Python!` 5 keer, onder elkaar, afprint. Maak gebruik van een `while` loop.

```
1 i = 0
2 while i < 5:
3     print("We like Python!")
4     i=i+1
```

Submit

Question 3: Absolute waarde

✓ Perfect



Schrijf een programma dat van een gegeven waarde (in variabele `a`) de absolute waarde afdruckt, zonder gebruik te maken van de ingebouwde functie `abs`. Er is al een variabele `a` die verschillende waarden kan aannemen.

```
1 # Jouw code komt hier
2 if a < 0 : a = -a
3 print(a)
```

Submit

Question 4: Zet om naar een While-loop

✓ Perfect



Zet onderstaande code om naar een code die gebruik maakt van een `while` loop.

```
print(1)
print(2)
print(3)
print(4)
print(5)
print("Goodbye!")
```

```
1 i = 1
2 while i < 6:
3     print(i)
4     i+=1
5 print("Goodbye!")
```

Submit

Question 5: Zet om naar een While-loop

✓ Perfect



Zet onderstaande code om naar een code die gebruik maakt van een `while` loop.

```
print(5)
print(4)
print(3)
print(2)
print(1)
print("Goodbye!")
```

```
1 i = 5
2 while i > 0:
3     print(i)
4     i-=1
```

```
5 print("Goodbye!")
```

Submit

Question 6: Schuine zijde

✓ Perfect

Op je rekentoestel en in Geogebra gebruik je het hoedje (^) om machten uit te drukken, bv $2^3 = 8$. In Python gebeurt dat met de `**` operator. Je noteert het dus als volgt: $2^{**3} = 8$.

Schrijf een programma dat, gegeven de lengte van twee zijdes van een rechthoekige driehoek, de lengte van de schuine zijde teruggeeft. (Hint: gebruik de `**` operator om de vierkantswortel te berekenen.)

Er zijn 2 variabelen a en b die verschillende waarden zullen aannemen.

```
1 sz = a**2 + b**2
2 print(sz**(1/2))
```

Submit

Question 7: Puntentabel

✓ Perfect

Schrijf een programma dat een punt aan de gebruiker opvraagt, en een string afprint dat de graad van dit punt aangeeft. Gebruik onderstaande verdeling.

- ≥ 90 : grootste onderscheiding en de gelukwensen van de examencommissie
- [85-90] : grootste onderscheiding
- [77-85] : grote onderscheiding
- [68-77] : onderscheiding
- [50-68] : op voldoende wijze
- < 50 : onvoldoende

```
1 punt = int(input("Geef een punt op tussen 0 en 100 (incl.)"))
2 if punt >= 90: print("grootste onderscheiding en de gelukwensen van de examencommissie")
3 elif (punt >= 85 and punt < 90): print("grootste onderscheiding")
4 elif (punt >= 77 and punt < 85): print("grote onderscheiding")
5 elif (punt >= 68 and punt < 77): print("onderscheiding")
6 elif (punt >= 50 and punt < 68): print("op voldoende wijze")
7 else: print("onvoldoende")
```

Submit

Question 8: Extra controle voor de vorige vraag

✓ Perfect



Submit

Question 9: Schrikkeljaar

✓ Perfect



In een programmeertaal zoals Python kan je op verschillende manieren een deling uitvoeren:

7/2 geeft 3.5, de deling van reële getallen
7//2 geeft 3, het quotiënt van de gehele deling (de staartdeling weet je nog ,, , lang geleden toen de dieren nog konden praten ...)
7%2 (lees: 7 modulo 2) geeft 1, de rest van de gehele deling. We noemen dit de modulo operator.

Om deelbaarheid te testen, gebruik je dus de modulo operator (want de rest is dan ...)

Schrijf een programma dat nakijkt of een gegeven jaar een schrikkeljaar is. Een schrikkeljaar is deelbaar door 4 maar niet door 100, tenzij het deelbaar is door 400. 2004 en 2000 zijn schrikkeljaren, maar 1800 en 1900 niet. Het gegeven jaar wordt gevraagd aan de gebruiker, het programma print `--schrikkeljaar--` ofwel `--geen schrikkeljaar--` op het scherm af naargelang het resultaat (let op de extra `--`).

```
1 jaar = int(input("Geef een jaar op"))
2 if jaar % 4 == 0 and jaar % 100 != 0 or jaar % 400 == 0:
3     print("--schrikkeljaar--")
4 else:
5     print("--geen schrikkeljaar--")
```

Submit

Question 10: Extra controle voor de vorige vraag

✓ Perfect



Submit

Question 11: Alarm

✓ Perfect



Schrijf een programma dat berekent hoe laat het is over `x` aantal uren. Als input geef je het huidige uur in (als een integer) en vervolgens het aantal uren. Print uit hoe laat het is over `x` uur (in 24u notatie).

```
1 uur = int(input("Hoe laat is het nu (in uren)?"))
2 x = int(input("Hoeveel uur verder wil je?"))
3
4 if x // 24 > 0: x = x % 24
5 if(uur + x >= 24): print(uur + x - 24)
6 else: print(uur + x)
```

Submit

Question 12: Convergentie

✓ Perfect



Schrijf een programma dat nakijkt of $(\frac{n}{n+1})_{n \in \mathbb{N}}$ convergeert naar 1 (ie. de rij $\frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \dots$). Hint: rekenen met floating point getallen is niet altijd exact, daarom is het niet veilig om floating point getallen te testen op gelijkheid. Een goede programmeur zal de volgende constructie gebruiken als zij of hij wilt weten of x gelijk is aan y :

```
if abs(x-y) < epsilon: # if x is approximately equal to y
    # do something
```

Merk op dat abs staat voor absolute waarde.

Definiëer de variabele `epsilon` aan het begin van je programma en stel deze gelijk aan 0.0000001. Van zodra het verschil tussen $(\frac{n}{n+1})$ en 1 kleiner wordt dan epsilon, dan stopt het programma en print het de waarde `n` af op het scherm.

```
1 epsilon = 0.0000001
2 i = 1
3 n = 0
4
5 while abs(n - 1) > epsilon:
6     n = i / (i+1)
7     i+=1
8
9 print(i-1)
```

Submit