

Python: 4. Condities

Dr. Cornelis Stal

April 27, 2022

1 Condities

1.1 Algemeen

Een `if`-statement bestaat uit een test en één of meerdere acties. De test is een zogeheten booleaanse expressie of “boolean expression”. De acties worden alleen uitgevoerd als de test evalueert als zijnde waar of `True`. Een app op een smartphone kan bijvoorbeeld een waarschuwing geven als de batterij minder dan 5% vol is. Dat betekent dat de app test of een zekere variabele (bijvoorbeeld “`batterij_energie`”) kleiner is dan 0.05. De booleaanse expressie is dan:

```
batterij_energie < 0.05
```

Als voor deze expressie een waarde wordt gevonden die kleiner is dan 0.05, is de expressie waar of `True`. Indien echter een de variabele een waarde heeft die gelijk is of groter is dan 0.05, evalueert de test de expressie als zijnde onwaar of `False`.

Zoals eveneens besproken in cursussen en documentatie over JavaScript, zijn de meest gebruikte booleaanse expressies eigenlijk vergelijkingen. Een vergelijking bestaat uit twee waardes met een vergelijkingsoperator ertussen. Deze Vergelijkingsoperatoren zijn:

- “`<`”: kleiner dan
- “`<=`”: kleiner dan of gelijk aan
- “`==`”: gelijk aan
- “`>=`”: groter dan of gelijk aan
- “`>`”: groter dan
- “`!=`”: niet gelijk aan

Opmerking: een veelgemaakte fout is om twee waardes te vergelijken met een enkele `=`.

Dit is echter een zogenaamde toekenningsoperator die gebruikt wordt om een bepaalde waarde aan een variabele toe te kennen.

1.2 Enkelvoudige expressies

De syntax van een `if`-statement is als volgt:

```
if <expressie>:      <actie>
```

Let op de dubbele punt (`:`) die achter de booleaanse expressie staat, en het feit dat `<actie>` inspringt. Deze dubbele punt komen we straks ook nog tegen bij de behandeling van lussen en functies. Zoals gezegd is Python zeer streng met inspringen. Foutief gebruik van inspringen zal resulteren in een `SyntaxError`, meer bepaald een “`IndentationError`”.

De volgende code zal resulteren in een boodschap wanneer de variabele `x` gelijk is aan 5.

```
[ ]: x = 5
if x == 5:
    print("x is 5")
```

Met een `if`-statement kunnen we bepaalde delen van de code wel of niet uit laten voeren door Python:

```
[ ]: x = 7
if x < 10:
    print("Deze regel wordt alleen uitgevoerd als x < 10")
    print("En dat geldt ook voor deze regel")
print("Deze regel wordt echter altijd uitgevoerd")
```

Een beslissing kan dikwijls twee kanten uitgaan. Dat wil zeggen, onder bepaalde omstandigheden willen we een bepaalde actie uit laten voeren en als die omstandigheden niet optreden willen we een andere actie uitvoeren. Python staat dit toe door aan een `if`-statement een `else`-tak toe te voegen. De syntax hiervan is als volgt:

```
if <expressie>:      <actie> else:      <actie>
```

Bijvoorbeeld:

```
[ ]: x = 4
if x > 2:
    print("x is groter dan 2")
else:
    print("x is kleiner dan of gelijk aan 2")
```

Het komt voor dat we een situatie hebben waarbij we slechts een blok uit een collectie van meerdere blokken code willen uitvoeren. Dit soort “meer-weg beslissingen” kunnen we implementeren met een extra toevoeging aan een `if`-statement, namelijk in de vorm van een of meer `elif`-statements. `elif` staat hier voor `else if` en is dus een combinatie van het `if`- en het `else`-statement.

```
[ ]: nauwkeurigheid = 3
if nauwkeurigheid < 2:
    print("Meting voldoet")
elif nauwkeurigheid < 4:
    print("Meting voldoet net")
else:
    print("Meting voldoet niet")
```

1.3 Meervoudige en geneste expressies

Bovenstaande condities zijn opgebouwd uit enkele expressies, waarbij het resultaat van een vergelijking afhankelijk is van een simpele vergelijking van twee zaken. Indien er meerdere zaken vergeleken moeten worden, maken we gebruik van meervoudige expressies. De combinatie van twee conditionele expressies maken we met behulp van `and`. In dat geval moet aan beide voorwaarden voldaan worden. Indien voor twee vergelijkingen beide of slechts één `True` teruggegeven moet worden, zal `or` gebruikt worden.

Expressies kunnen daarnaast ook genest worden binnen een andere expressie:

```
[ ]: nauwkeurigheid = 3
aantal = 100
if nauwkeurigheid < 2 and aantal > 250:
    print("Metting voldoet")
elif nauwkeurigheid < 4 and aantal > 200:
    print("Zwakke meting")
else:
    if nauwkeurigheid < 5 or aantal > 100:
        print("Nazicht vereist")
    else:
        print("Toestel rijp voor de sloop")
```

1.4 Oefeningen

Opdracht: pakje sturen

Voor het versturen van een pakje rekent de post het volgende bedrag aan: Voor een pakje van twee kilogram of minder betaalt men 18 euro. Voor elke kilogram meer wordt er een bijkomend bedrag van 5 euro per kilogram gerekend. Dit geldt ook voor de laatste (eventueel onvolledige) kilogram.

Schrijf code die het te betalen bedrag berekent.

```
[ ]: ## UW CODE HIER ##
```

Opdracht: kinderbijslag berekenen

In een vereenvoudigde versie wordt de oude kinderbijslag als volgt berekend (zonder rekening te houden met de leeftijd):

- kind 1: €90.28;
- kind 2: €167.05;
- kind 3 en volgende: €249.91.

Bereken de kinderbijslag voor een opgegeven aantal kinderen. (Bron: [Gezinsbond, kinderbijslag] (<https://www.gezinsbond.be/Gezinspolitiek/acties/Paginas/Hoe-hog-zijn-de-huidige-bedragen-van-%20de-kinderbijslag.aspx>)

```
[ ]: ## UW CODE HIER ##
```

Opdracht: internationale prijsvergelijkingen

De Big Mac index is een methode waarmee wisselkoersen op een licht verteerbare manier kunnen geanalyseerd worden. Deze index werd in 1986 door Pam Woodall ontwikkeld voor de Amerikaanse krant The Economist. Sindsdien publiceert die krant elk jaar een geactualiseerde versie van de index. Deze index gaf ook aanleiding tot de term burgerconomics. Het principe werkt heel eenvoudig. De Big Mac wisselkoers tussen twee landen kan berekend worden door de prijs van een Big Mac in één land (in de lokale munteenheid van dat land) te delen door de prijs van een Big Mac in een ander land

(opnieuw in de lokale munteenheid van dat land). Deze berekende wisselkoers wordt dan vergeleken met de reële wisselkoers op dat ogenblik. Indien deze lager uitvalt, dan is de eerste munt ondergewaardeerd ten opzichte van de tweede, anders is de eerste munt overgewaardeerd ten opzichte van de tweede. Dit wordt geïllustreerd aan de hand van volgend voorbeeld:

- in Europa betaalt men €4.74 voor een Big Mac;
- in de Verenigde Staten betaalt men \$5.51 voor een Big Mac; – de wisselkoers kan dus berekend worden als $\text{€}4.74 / \$5.51$ per dollar = €0.860;
- de reële wisselkoers bedraagt echter €0.89 per dollar;
- de euro is dus met -3.37% overgewaardeerd ten opzichte van de dollar ($0.86 - 0.89 / 0.89 \times 100 = -3.37$)

Schrijf een programma dat op basis van de Big Mac index een analyse maakt van de wisselkoers van een bepaald land ten opzichte van de dollar. Dit programma heeft twee argumenten die als floating point getallen moeten beschouwd worden:

- De prijs van een Big Mac;
- De reële wisselkoers ten opzichte van de dollar

Bij de berekening van de Big Mac wisselkoers mag de functie uitgaan van een kostprijs van \$5.51 voor een Big Mac in de Verenigde Staten. Op basis van de procentuele verhouding van de berekende dollarkoers ten opzichte van de reële dollarkoers, moet de functie een string teruggeven overeenkomstig onderstaande tabel.

Verhouding (x)	Waardering
v < -25	sterk ondergewaardeerd
-25 < v < -5	ondergewaardeerd
-5 < v < 5	ongeveer gelijk
5 < v < 25	overgewaardeerd
v > 25	sterk overgewaardeerd

[]: *## UW CODE HIER ##*