Universiteit van Stellenbosch
Departement Wiskundige Wetenskappe
Afdeling Rekenaarwetenskap

University of Stellenbosch
Department of Mathematical Sciences
Computer Science Division

Wetenskaplike Berekening 272 / Scientific Computing 272

Tutoriaal 5: Keuses in Python / Tutorial 5: Choices in Python

2020-07-23/24 Opgestel deur Willem Bester Gemodereer deur Brink van der Merwe

Agtergrond

Dié tutoriaal is 'n praktiese oorsig van keuses in Python. Ter voorbereiding, bestudeer die vyfde reeks Python-skyfies. Ek stel voor dat u die gids `~/wb272/tut05` skep vir u werk aan dié tutoriaal.

Background

This tutorial is a practical overview of choices in Python. To prepare, study the fifth set of Python slides. I suggest you create the directory `~/wb272/tut05` for your work on this tutorial.

Uitkomste

Wanneer u die tutoriaal voltooi het, behoort u in staat te wees om die volgende te doen: (1) Boolese uitdrukkings per hand te evalueer, (2) stilisties korrekte Boolese uitdrukkings in Python te skryf, en (3) die `if`-stelling (met `elif`- en `else`-klousules) te gebruik om keuses te maak.

Outcomes

When you have complete the tutorial, you should be able to do the following: (1) evaluate Boolean expressions by hand, (2) write stylistically correct Boolean expressions in Python, and (3) use the `if` statement (with `elif` and `else` clauses) to make choices,

1. Sonder om Python te gebruik, skryf die waarde van elkeen van die volgende uitdrukkings neer. Toets u antwoorde hierná met behulp van die Python-intepreteerder.
   (a) `True and not False`
   (b) `True or True and False`
   (c) `not True or not False`
   (d) `True and not 0`
   (e) `52 < 52.3`
   (f) `1 + 52 < 52.3`
   (g) `4 != 4.0`

   Without using Python, write down which value each of the following expressions give. Then verify your answers by using the Python interpreter.

2. Gegee die veranderlikes `x` en `y`, skryf *stilisties korrekte* uitdrukkings wat tot `True` evalueer as en slegs as
   (a) albei veranderlikes die waarde `True` het;
   (b) `x` die waarde `False` het;
   (c) ten minste een van die veranderlikes `True` is.

   Let wel: "Stilisties korrek" verwys, onder andere, daarna dat u nie onnodige uitdrukkings in u kode moet gebruik nie. Byvoorbeeld, die stellings "`if x`" en "`if x == True`" is ekwivalent, want die waguitdrukking van die eerste `if` evalueer tot waar as en slegs as `x` waar is; daarom is die eerste `if`-stelling stilisties beter as die tweede. Soortgelyk is dit stilisties beter om "`if not x`" in plaas van "`if x == False`" te skryf.

   Given the variables `x` and `y`, write *stylistically correct* expressions that evaluates to `True` if and only if
   both variables are `True`;
   `x` is False;
   at least one of the variables is `True`.

   Note: "Stylistically correct" refers, inter alia, to your not using unnecessary expressions in your code. For example, die statements "`if x`" and "`if x == True`" are equivalent, because the guard expression of the first `if` evaluates to true if and only if `x` is true; therefore, the first `if` statement is stylistically better than the second. Similarly, it is stylistically better to write "`if not x`" instead of "`if x == False`".

3. U wil hê 'n outomatiese kamera moet aanskakel wanneer die ligvlak kleiner as 0.01 is of as die temperatuur bo vriespunt is, maar nie as albei voorwaardes waar is nie. U eerste probeerslag om dié te skryf, lyk so:

   You want an automatic camera to switch on if the light level is less than 0.01 or if the temperature is above freezing, but not if both conditions are true. Your first attempt to write this is as follows:

   ```
   if light < 0.01 or temperature > 0.0:
       if light < 0.01 and temperature > 0.0:
           pass
       else
           camera.on()
   ```

   'n Vriend—en neem vir die oomblik aan u het ten minste een—sê dat dié 'n ekslusiewe OF is en dat u dit meer eenvoudig soos volg kan skryf:

   A friend—assuming, for the moment, you have at least one—says that this is an exclusive OR, and that you can write it more simple as follows:

   ```
   if (light < 0.01) != (temperature > 0.0):
       camera.on()
   ```

   Is u vriend korrek? Indien wel, verduidelik waarom; indien nie, gee waardes vir `light` en `temperature` wat lei tot verskillende resultate vir die twee kodefragmente.

   Is your friend correct? If so, explain why; if not, give values for `light` and `temperature` that produces different results for the two code fragments.

4. Gegee twee veranderlikes `population` en `land_area`, wat albei wisselpuntwaardes bevat, skryf 'n `if`-stelling wat

   Given two variables `population` and `land_area`, both containing floating-point values, write an `if` statement that

   (a) die bevolking uitdruk as dit kleiner as 10 000 000 is;

   prints out the population if it is less than 10 000 000;

   (b) die bevolking uitdruk is dit tussen 10 000 000 en 35 000 000 (albei ingeluit) is;

   prints out the population if it is between 10 000 000 and 35 000 000 (both inclusive);

   (c) "Dig bevolk" uitdruk indien die bevolkingsdigtheid—die aantal mense per eenheid oppervlakte—groter as 100 is;

   prints out "Densely populated" if the population density—the number of people per unit of area—is greater than 100;

   (d) "Dig bevolk" uitdruk indien die bevolkingsdigtheid groter as 100 is; indien dit nie is nie, druk "Yl bevolk" uit.

   prints out "Densely populated" if the population density is greater than 100; if it is not, print out "Sparsely populated".

5. Skryf 'n funksie `rock_paper_scissors` wat twee parameters `p1` en `p2` neem, wat aandui wat deur twee spelers in 'n beurt van die spel Steen–Papier–Skêr gespeel is. Die parameters is stringe: "R" vir steen, "P" vir papier en "S" vir skêr. U funksie moet uitdruk watter speler die spel wen.

   Write a function `rock_paper_scissors` that takes two parameters `p1` and `p2`, which indicate what two players played in a game of Rock–Paper–Scissors. The parameters are strings: "R" for rock, "P" for paper, and "S" for scissors. Your function must print out which player wins the game.

6. Beskou die volgende funksie:

   Consider the following function:

   ```
   def remove_negs(num_list):
       '''Remove negative numbers from the list num_list.'''
       for item in num_list:
           if item < 0:
               num_list.remove(item)
   ```

   Wanneer die lys $[1, 2, 3, -3, 6, -1, -3, 2]$ aan die funksie gestuur word, is die resultaat $[1, 2, 3, 6, -3, 2]$. Daar is kennelik 'n probleem met die implementasie. Herskryf die funksie om die probleem reg te stel. WENKE: Vra uself af watter items oorweeg word, en stel ook vas wat die `del` sleutelwoord doen.

   When the list $[1, 2, 3, -3, 6, -1, -3, 2]$ is passed to the function, the result is $[1, 2, 3, 6, -3, 2]$. Obviously, there is a flaw in the implementation. Rewrite the function to fix the problem. HINTS: Ask yourself which items are considered, and also determine what the `del` keyword does.

7. Skryf 'n Python-funksie `find_max` wat twee parameters, 'n lys `L` en 'n indeks `i`, neem en dan die *indeks* van die maksimum element in die sublys `L[i:]` terugstuur. Behalwe `len` en `range`, mag u geen (ander) funksies of metodes vanuit `find_max` roep nie: Moenie die `index`-metode of die `max`-funksie gebruik nie. Neem aan dat alle elemente in die lys wedersyds vergelykbaar is, dat daar 'n natuurlike orde op die elemente bestaan en dat die waarde van `i` 'n geldige, nie-negatiewe indeks in `L` is. WENK: Begin u funksie met die aanname dat die maksimumwaarde by die indeks lê wat aanvanklik deur die parameter `i` gegee word.

8. NET VIR DIE PRET: Kyk na die videosnit by `https://youtu.be/iapcKVn7DdY` en implementeer 'n funksie wat uitwerk watter speler die spel wat in die video beskryf word, wen.

Write a Python function `find_max` that takes two parameters, a list `L` and an index `i`, and returns the *index* of the maximum element in the sublist `L[i:]`. Except for `len` and `range`, you may not call any (other) functions or methods from inside `find_max`: Do not use the `index` method or the `max` function. Assume that all elements in the list are mutually comparable, that a natural order exists over the elements, and that the value of `i` is a valid, non-negative index into `L`. HINT: Start your function with the assumption that the largest value is at the index given initially by the parameter `i`.

JUST FOR FUN: Watch the video clip at `https://youtu.be/iapcKVn7DdY` and implements a function to work out which player wins the game described in the video.