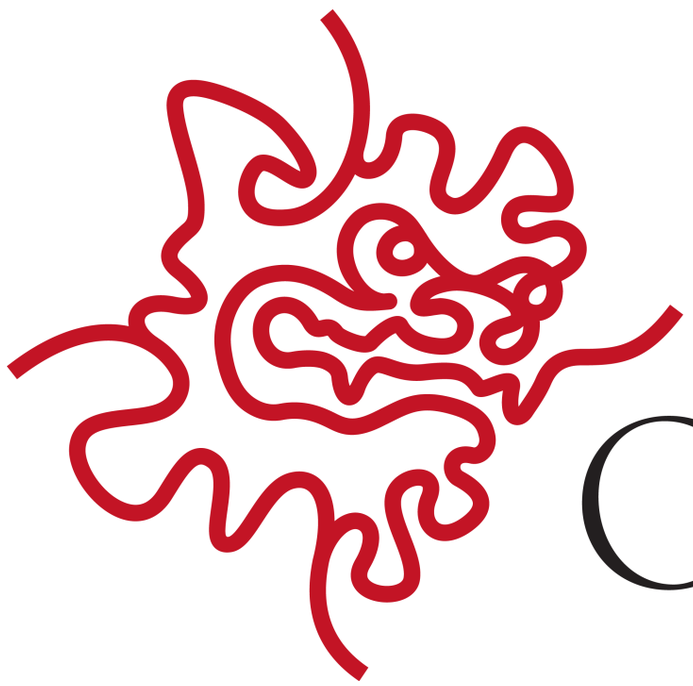
SKILLPILLS

Skill Pill: Intro to Programming

Spyder, variables and lists

Jeremie Gillet



OIST

Outline

SKILLPILLS

- Python's strengths and weaknesses
- Spyder IDE, scripts and console
- Variables
- Lists

Python's pros and cons

SKILLPILLS

- Easy to read, write and learn
- Very flexible
- Many libraries (scipy, numpy, matplotlib, data science, machine learning, games, web…)
- Well documented, all mistakes have been made
- Slow

Note on Python 2 vs 3

SKILLPILLS

- There are small visible differences between Python 2 and 3
- Currently, both versions co-exist
- Python 2 will not be maintained past 2020
- From now on, Python = Python 3

Anaconda and Spyder

SKILLPILLS

- Anaconda is an open source distribution of Python and R for data science
- Anaconda manages thousands of packages
- Spyder is an open-source integrated development environment (IDE) for scientific programming in Python
- Spyder helps you write Python code

Run scripts

Spyder

SKILLPILLS

Users/jie/OneDrive - OIST/OIST/Events/Bootcamp 2018/Exercices

Editor - /Users/jie/OneDrive - OIST/OIST/Events/Bootcamp 2018/Exercices/Day 1/Day 1 - Session 3a.py

untitled0.py | Day 1 - Session 3b.py | Day 1 - Session 3a.py

```
#Ex: give the value 3 to a, 5 to b and 0 to c
a, b, c = 3, 5, 0


#Ex: divide a by b and b by a
a/b
b/a

#Ex: same, with integer division
a//b
b//a

#Ex: divide a by c, with float and integer division
#a/c
#a//c

#Ex: calculate b modulo a and b modulo c
b%a
#b%c

#Ex: calculate b to the power a
b**a

#Ex: set a,b,c,d to 1,2,4,12 and find 1 or 2 ways to combine all 4 variables with +-*// to make
#example: a*(c-b)*d
a,b,c,d = 1,2,4,12
a*(c-b)*d
(d-c)*(a+b)
(a+b)*c+d
a*c*d//b

#Ex: set a to 7
a = 7

#Ex: check the type of a
type(a)

#Ex: without touching the "7" key on your keyboard, set b to the float value of 7
b = 1.0 * a

#Ex: check that b is a float
type(b)

#Ex: guess what the type of a+b should be
type(a+b)

#Ex: what is the largest int you can define?
#Ans: something big enough to use all your computer's memory
```

Scripts

Variable explorer

Name	Type	Size	Value
a	str	1	I like Skill Pills! I like Skill Pills! I like Skill Pills! I like Ski ...
b	float	1	7.0
c	int	1	4
d	int	1	12
fasta_ex6	str	1	>sp|Q4VBE4|EGFLA_MOUSE Pikachurin OS=Mus musculus OX=10090 GN=Egflam P ...
s_ex5	str	1	#this is a comment line #this is another comment line
x	str	1	hi

Variable explorer, file explorer, help

File explorer | Variable explorer | Help

IPython console

Console 1/A

```
In [80]: ex5()
---------- EXERCICE 5 ----------
Traceback (most recent call last):

  File "<ipython-input-80-fe859a23b026>", line 1, in <module>
    ex5()

  File "/Users/jie/OneDrive - OIST/OIST/Events/Bootcamp 2018/Exercices/Day 1/Day 1 - Session 3b.py", line 84, in ex5
    splits = [line.split("\t") for line in s_ex5.split("\n") if line[0]!="#"]

  File "/Users/jie/OneDrive - OIST/OIST/Events/Bootcamp 2018/Exercices/Day 1/Day 1 - Session 3b.py", line 84, in <listcomp>
    splits = [line.split("\t") for line in s_ex5.split("\n") if line[0]!="#"]

IndexError: string index out of range

In [81]:
```

Console

IPython console | History log

Permissions: RW End-of-lines: LF Encoding: ASCII Line: 1 Column: 1 Memory: 76 %

Let's try a script

SKILLPILLS

- File > New File…
- Write `print("Hello World")`
- Click Run File (F5)
- Ok to the dialog box
- Look at the console

Let's try the console

SKILLPILLS

- Click on the console
- Write `print("Hello World")`
- Press enter
- Try up arrow, ctrl+a, ctrl+e, escape

Just so you know…

- We could do the same with less fancy tools
- Ex: TextEdit + terminal
- But: no user interface, syntax highlighting, autocomplete, saving variables…

Comments

SKILLPILLS

- Single line comments:
  - `#this is a comment`
- Multi-line comments:
  - """
    these
    are
    comments
    """
- Comment EVERYTHING!!!!!

Variables

- Variables let you store values by giving them a name
  - `x = 2`
  - `my_string = “Hello”`
  - `a, b = 0, 1`
  - `y = x`
- There are rules for naming variables ([a-zA-z0-9_] only, cannot begin by a number, should be easy to understand…)

Types of Variables

SKILLPILLS

Type	Python name	Possible values
Boolean	bool	True, False
Integer	int	…,-2, -1, 0, 1, 2, …
Floating Point	float	…, 1E-8, 0.0, -0.001, …
Text (String)	str	“”, “Hi”, ‘Mom’, …
Complex	complex	3.1 + 7j, …
…		

Check types by using `type( )`

Casting Variables

SKILLPILLS

- Some variables can change types
  - `x = 2  # will be an int`
  - `x = float(2)  # will be a float`
  - `x = 1.0 * 2  # will be a float`
  - `x = int(2.99)  # will be 2`
  - `x = str(2)  # will be a string`
  - `x = int("Hi")  # will be an error`
  - `x = bool("Hi") # will be True`

Manipulating variables

SKILLPILLS

- Boolean logic: `not`, `and`, `or`
- Basic math: `+`, `-`, `*`, `/`, `//`, `**`, `%`
- Augmented assignments: `+=`, `-=`,...
- Comparaisons: `<`, `>`, `<=`, `==`, `!=`,...
- String operations: `+`, `*`

Augmented assignments

SKILLPILLS

- The pattern `count = count + 1` is extremely common and is used to increase the value of a variable by 1
- It is equivalent to `count += 1`

User input

- `x = input("Prompt message")`
- Whichever you type until you press enter will be saved in `x`
- You can cast `x` into something else

Exercise

variables.py

Test your answers in the console

Time: 20 minutes

Lists

- Lists are a type of data structure, in the family of iterators
- Lists can contain any type of values, numerical, strings, other lists…
- `my_list1 = [1, 2, 3]`
- `my_list2 = [“Hi”, 3, [ 3.0, False], my_list1]`

Retrieving list elements

```
   index     0    1    2
my_list = [1,   2,   3]
   index'   -3   -2   -1
```

- `element = my_list[index]`
- The index must be integer, can be negative
- Indexing starts at 0
- For lists in lists: `my_list[i1][i2]`

Slicing lists

```
  index      0   1   2   3   4   5   6
my_list = [1, 2, 3, 4, 5, 6, 7]
```

- Most general: `my_list[from : to : by]`
  - `my_list[1:6:2] # will be [2, 4, 6]`
- `to` is NOT inclusive
- If you omit values, then default to `[0: length of list: 1]`
  - `my_list[3:]  # will be [4, 5, 6, 7]`
  - `my_list[:3]  # will be [1, 2, 3]`
  - `my_list[::4] # will be [1, 5]`
  - `my_list[:]   # will be the same as my_list`

Other list methods

SKILLPILLS

- `x.append(val) # adds val to the end of x`
- `x.insert(i, x) # inserts an element at index i`
- `x.remove(val) # removes the first occurrence of val`
- `del x[i] # deletes the i-th element of x`
- `x.count(val) # counts the number of time val is in x`
- `x.index(val) # returns index of first occurrence of val in x`
- `val in x # checks if x contains val`

More list methods

SKILLPILLS

- `x + y # concatenates x and y`
- `len(x) # Length of x`
- `max(x) # maximum value in x (min works too)`
- `sum(x) # sum of values in x`
- `x.sort() # sorts x in ascending order`
- `sorted(x) # returns a sorted version of x`

Tuples

SKILLPILLS

- A tuple is simpler list (really, an iterator)
- `t = (1, 2, 3)`
- `t[1:3] # returns [2, 3]`
- `len(t) # returns 3`
- …
- Tuples are immutable: cannot use append, del, insert, …

Ranges

- A range is an iterator, often used for loops
- `range(from, to, by) # creates a range`
- `list(range(1,5)) # returns [1, 2, 3, 4]`

Exercise

lists.py

Solve the problem within the script
and test your solution by running it

Time: until the end