



JOHANNES GUTENBERG  
UNIVERSITÄT MAINZ

# INTRODUCTION TO PYTHON FOR BIOLOGISTS

## IF-FOR-WHILE Revisited

Katerina Taškova<sup>1</sup>    Jean-Fred Fontaine<sup>1,2</sup>

<sup>1</sup>Faculty of Biology, Johannes Gutenberg-Universität Mainz, Mainz, Germany

<sup>2</sup>Genomics and Computational Biology, Kernel Press, Mainz, Germany

<https://cbdm.uni-mainz.de/mb17>

March 24, 2017

# Table of Contents

IF revisited

FOR revisited

WHILE revisited

**\*\*Exercise\*\***

References

IF revisited

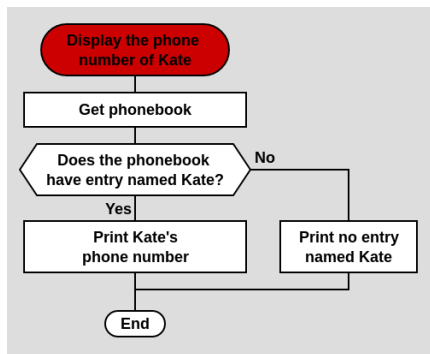
FOR revisited

WHILE revisited

**\*\*Exercise\*\***

References

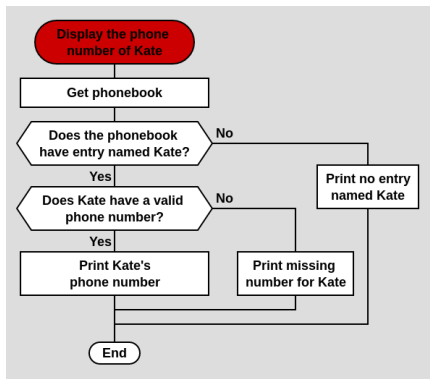
## IF I



```

1 phonebook={ 'Mara': 491761132347,
2   'Daniel': 491761262348,
3   'Stefan': None,
4   'Kate': 491734263358,
5   'Ana': None,
6   'Joerg': 4917756708022,
7   'Marc': 4917822451089}
8
9 name = 'Kate'
10 if name in phonebook.keys():
11     print(name, phonebook[name], sep = ":")
12 else:
13     print("No entry named", name)
  
```

## IF II



```

1 phonebook={'Mara': 491761132347,
2   'Daniel': 491761262348,
3   'Stefan': None,
4   'Kate': 491734263358,
5   'Ana': None,
6   'Joerg': 4917756708022,
7   'Marc': 4917822451089}
8
9 name = 'Kate'
10 if name in phonebook.keys():
11     if phonebook[name] is not None:
12         print(name, phonebook[name], sep = ":")
13     else:
14         print("Missing number for entry", name)
15
16 else:
17     print("No entry named", name)
  
```

IF revisited

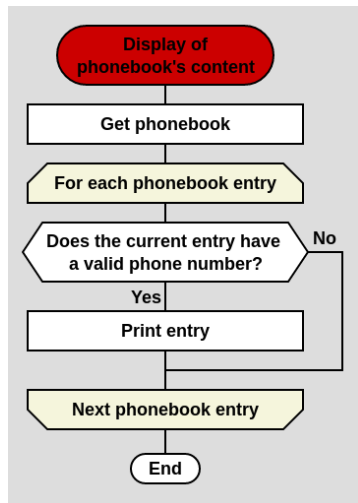
**FOR revisited**

WHILE revisited

**\*\*Exercise\*\***

References

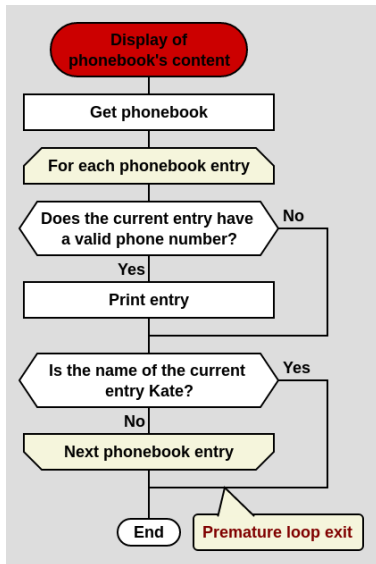
## FOR I



```

1  phonebook={'Mara': 491761132347,
2  'Daniel': 491761262348,
3  'Stefan': None,
4  'Kate': 491734263358,
5  'Ana': None,
6  'Joerg': 4917756708022,
7  'Marc': 4917822451089}
8
9  for name in sorted(phonebook): # fixed order
10
11     if phonebook[name] is not None:
12         print(name, phonebook[name], sep=": ")
13     else:
14         print(name, "missing number", sep=": ")
15
16     #####
17
18  Iteration 1 -> Ana: missing number
19  Iteration 2 -> Daniel: 491761262348
20  Iteration 3 -> Joerg: 4917756708022
21  Iteration 4 -> Kate: 491734263358
22  Iteration 5 -> Mara: 491761132347
23  Iteration 6 -> Marc: 4917822451089
24  Iteration 7 -> Stefan: missing number
  
```

## FOR II



```

1  phonebook={'Mara': 491761132347,
2  'Daniel': 491761262348,
3  'Stefan': None,
4  'Kate': 491734263358,
5  'Ana': None,
6  'Joerg': 4917756708022,
7  'Marc': 4917822451089}
8
9  stop_name = 'Kate'
10 for name in sorted(phonebook): # fixed order
11
12     if phonebook[name] is not None:
13         print(name, phonebook[name], sep=": ")
14     else:
15         print( name, "missing number", sep=": ")
16
17     if name == stop_name:
18         print("Found", stop_name, "- force exit")
19         break
20
21 else: # belongs to the for-loop
22     print("Normal loop exit - no break")
23
24 #####
25 iteration 1 -> Ana: missing number
26 iteration 2 -> Daniel: 491761262348
27 iteration 3 -> Joerg: 4917756708022
28 iteration 4 -> Kate: 491734263358
29                 Found Kate - force exit
  
```



## FOR III

Proteins expressed in tissues

	P1	P2	P3	P4
liver	x	x		
kidney	x			
blood	x	x		x
bone			x	
muscle			x	
brain				x

Similarity matrix, N=4

 $(N^2 - N)/2 = 6$  values

	P1	P2	P3	P4
P1		?	?	?
P2			?	?
P3				?
P4				

## FOR example III

Proteins expressed in tissues

	P1	P2	P3	P4
liver	x	x		
kidney	x			
blood	x	x		x
bone			x	
muscle			x	
brain				x

Similarity matrix, N=4  
 $(N^2 - N)/2 = 6$  values

	P1	P2	P3	P4
P1		?	?	?
P2			?	?
P3				?
P4				

```

1 # Example with nested for-loops
2 proteins = {'P1': {'liver', 'kidney', 'blood'},
3            'P2': {'liver', 'blood'},
4            'P3': {'bone', 'muscle'},
5            'P4': {'brain', 'blood'}}
6
7 names = list(proteins.keys())
8 print(names) # ['P1', 'P4', 'P3', 'P2']
9
10 for id1 in range(0, len(names)):
11     for id2 in range(id1+1, len(names)):
12
13         set1 = proteins[names[id1]]
14         set2 = proteins[names[id2]]
15
16         # Calculate similarity[%] as 100*overlap/union
17         overlap = len(set1 & set2)
18         union = len(set1 | set2)
19         sim = 100*overlap/union
20
21     print(names[id1], names[id2], sim)

```

IF revisited

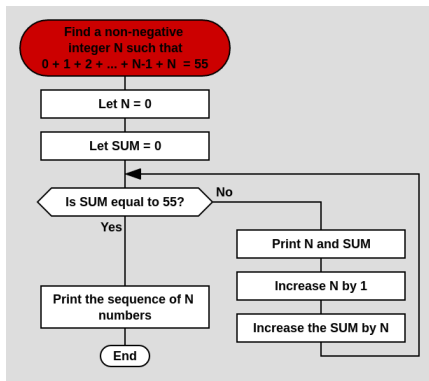
FOR revisited

**WHILE revisited**

**\*\*Exercise\*\***

References

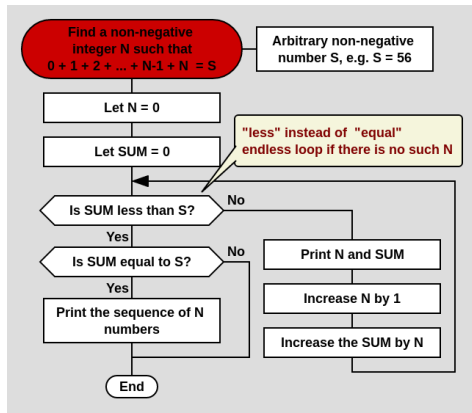
# WHILE I



```

1 # N will hold the integer we are looking for
2 # SUM will hold the sum of the first
3 # N non-negative integers
4
5 N = 0 # zero initialization
6 SUM = 0 # zero initialization
7
8 while SUM != 55:
9     N += 1
10    SUM += N
11
12 # Final print after loop exit
13 print("The sequence found is ")
14 print(list(range(0, N+1))) # Why N+1 ?
15 print("The sum of the sequence elements is")
16 print(SUM)
  
```

## WHILE II



```

1 # Carefully set the loop termination
2 # condition to avoid infinite loop
3
4 N = 0 # zero initialization
5 SUM = 0 # zero initialization
6 S = 56 # the threshold for the sum
7
8 while SUM < S:
9     N += 1
10    SUM += N
11
12 if SUM == S:
13     print("We found a sequence with sum of
14         its elements equal to", S)
15 else:
16     print("There is no sequence with sum
17         of its elements equal to", S)
  
```

IF revisited

FOR revisited

WHILE revisited

**\*\*Exercise\*\***

References

## Exercise

Exercise in Jupiter Notebook format

- IfForWhileRevisited.ipynb
- Download from <https://cbdm.uni-mainz.de/mb17/>

IF revisited

FOR revisited

WHILE revisited

**\*\*Exercise\*\***

**References**



## References

- Python documentation
  - [docs.python.org](https://docs.python.org)
- Modules of particular interest
  - numpy and scipy (scientific)
  - BioPython (Biology)
- Online tutorials (Python 2 or 3)
  - Google's Python Class
  - [ProgrammingForBiologists.org](http://ProgrammingForBiologists.org)
- Flowcharts
  - [drakon-editor.com](http://drakon-editor.com)