



University of
Zurich^{UZH}

Institute of Computational Linguistics

Programmiertechniken in der Computerlinguistik I

Herbstsemester 2015

4. Sitzung, 15. Oktober 2015

Martin Volk




University of
Zurich^{UZH}

Institute of Computational Linguistics

Overview: for loop + lists

	Examples
for loop	<pre>for my_item in my_list for my_num in range(0, 30) for my_num in range(0, 30, 3) for my_num in range(30)</pre>
lists	<pre>[2, 4, 6, 8, 10] ['python', 'is', 'great'] ['27.', 'Oktober', 2011]</pre>
list assignment and concatenation	<pre>my_list1 = [3, 6, 9, 12] my_list2 = [2, 4, 8, 16] my_long_list = my_list1 + my_list2</pre>
accessing and changing items in a list	<pre>my_item = my_list[2] my_part = my_long_list[:5] my_long_list[3] = 64</pre>

Page 2




**University of
Zurich**^{uzh}

Institute of Computational Linguistics

Overview: list operations

	Examples
sorting a list - reverse order - according to length of items	<pre>my_list = sorted(my_list) my_list = sorted(my_list, reverse=True) my_list = sorted(my_list, key=len)</pre>
inserting an item at the end	<pre>my_list.append(42)</pre>
inserting an item at the beginning - or at any other place	<pre>my_list.insert(0, 'hello') my_list.insert(3, 'world')</pre>
deleting a list deleting an item from a list	<pre>del my_list del my_list[3]</pre>
converting a string to a list - by splitting on the blank symbol	<pre>my_text = 'this is magic' my_list = my_text.split(' ')</pre>

Page 3



**University of
Zurich**^{uzh}

Institute of Computational Linguistics

Data structure: List

value

key

my_list

'this'	'is'	'magic'	'.'	'20.'	2011
0	1	2	3	4	...

my_list[2]

Page 4



University of
Zurich^{UZH}

Institute of Computational Linguistics

In class task 1

Given: a simple Python program for compound splitting with a *while* loop.

Rewrite the program using a *for* loop and the *range* function.

Page 5



University of
Zurich^{UZH}

Institute of Computational Linguistics

In class task 2

Given a string.

Write a Python program that prints all **character trigrams** (= sequences of 3 characters).

For example

Given the string **'Python is a great language'**

my_trigram_printer.py
returns:

```
Pyt
yth
tho
hon
on
n i
is
...
```

Page 6



University of
Zurich ^{UZH}

Institute of Computational Linguistics

In class task 3

Given a list of words.

Write a Python program that prints only every second word.

For example:

Given the list: **['Python', 'mostly' 'is', 'a', 'great', 'language']**

my_alternation_filter.py returns:

Python is great

Hint: Use a *for* loop and set the iterator as you need it.

Page 7



University of
Zurich ^{UZH}

Institute of Computational Linguistics

In class task 4

Given a list of numbers.

Write a Python program that computes the median value (i.e. the value where “half” of the numbers are lower and “half” are higher).

For example:

Given the list: **[18, 21, 16, 24, 19, 16, 23, 26]**

my_median.py returns:

The median is 21

Note: When the list has an even number of elements, the median shall be the first number of the upper half.

Page 8



Overview: File handling

	Examples
Command line arguments	<code>import sys</code> <code>my_filename = sys.argv[1]</code> <code>my_arguments = sys.argv</code>
Open file for reading Open file for writing - my_infile / my_outfile are file objects	<code>my_infile = open(my_filename, 'r')</code> <code>my_outfile = open(my_filename, 'w')</code>
Close file	<code>my_infile.close()</code>
Loop over each line in a file	<code>for my_line in my_infile</code>

Page 9



Programming Techniques in this Lesson

- Split words and check for compounding or inflection
- Compute max value (e.g. longest word) over a file
- Compute character trigrams
- Open a file for reading and loop over the lines

Page 10