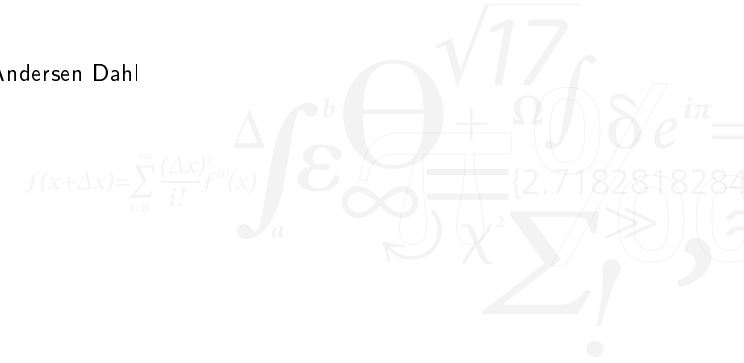
Lecture 6: Dictionaries

Morten Rieger Hannemose, Vedrana Andersen Dahl

Fall 2023



Today's lecture

1. Dictionaries (ca. 30 min)
2. Coding example (ca. 30 min)

Python Installation Support

Do you have problems getting Python to work?

Check https://pythonsupport.dtu.dk for changes on opening hours

Office hours	Monday	Tuesday	Wednesday	Thursday	Friday
10-12:30	302.A92	302.A92	302.A92	302.A92	302.A92
12:30-16		302.A92	302.A92	358.002	302.A92
18-22	Online	Online	Online	Online	Online

Online Help

pythonsupport@dtu.dk

https://pythonsupport.dtu.dk

Dictionaries: creation

```
my_dictionary = {'name': 'Sasha', 'age': 14, 'class': '8b'}
another_dictionary = {1: 'one', 303: 'three o three'}
third_dictionary = {'aqua': 101, 'audit': 27}
yal = {'seg': [1, 2, 3], 'dct': my_dictionary}
empty_dictionary = {}
```

- Look back at lists: a sequence of values indexed by integers starting with 0, i.e. index-value pairs.
- Dictionary – a sequence of key-value pairs.
- Word `dict` is a built-in type, don't use it as a variable name.

Dictionaries: accessing values

```
my_dict = {'a': 'apple',
           'b': 'banana',
           'c': 'cucumber',
           'd': 'durian'}
print('apple' in my_dict)
print('a' in my_dict)
for k in my_dict:
    print(k)
keys = my_dict.keys()
values = my_dict.values()
items = my_dict.items()
```

- Keyword `in` checks for dictionary keys
- When using `in` with `for` loop, the dictionary keys are traversed

Dictionaries: adding and changing

```
my_dict = {'a': 'apple', 'b': 'banana',
           'd': 'durian', 'g': 'grapes',
           'j':'jackfruit', 'p':'pear'}

keys = my_dict.keys()
values = my_dict.values()

my_dict['d'] = 'date'
my_dict['m'] = 'mango'

print(keys)
print(values)
```

- Dictionary keys and values are mutable like lists

Dictionaries: use

```
person = {
  "first_name": "John",
  "last_name": "Smith",
  "age": 27,
  "address": {
    "street_address": "21 2nd Street",
    "city": "New York",
  },
  "phone_numbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
  ],
  "children": [
    "Catherine",
    "Thomas",
    "Trevor"
  ],
  "spouse": None
}
```

- Setting for larger projects
- Collection of unstructured information

Dictionaries use

```
names = ["John", "Bob", "Alice"]
ages = [23, 45, 67]

print(ages[names.index("Bob")])

# This is a bad idea.
# It's better to use a dictionary.

name_to_age = {"John": 23, "Bob": 45, "Alice": 67}
print(name_to_age["Bob"])

```

- Can accomplish the same as lists... but with much faster and cleaner syntax

Example 1

Write a function that takes a string. The function should return a dictionary where the keys are letters. The value for each key is a list of all the words from the string starting with that letter.

Solve the problem in two different ways:

- By creating a dictionary that contains 26 keys, one for each letter of the (English) alphabet.
- By creating a dictionary that contains only keys needed for storing the words that occur in the string.

Coding example

`alpha2phone.py`, (slightly modified) exam from May 2016.

Alpha to phone

On a phone keypad, each letter of the alphabet is assigned to one of the digits 2-9. This makes it possible to write alpha-numeric phone numbers using a mix of letters and digits (by replacing digits in the phone number by the corresponding letters).

Problem formulation

Create a function `alpha2phone` that takes as input an alpha-numeric (letters and digits) phone number as a string, and returns the corresponding numeric (only digits) phone number as a string. You may assume that all letters in the input are given as upper case.

Consider the alpha-numeric phone number 4525DTU1. Converted to a numeric phone number, it should be 45253881.

Code used for coding examples

Example 1

```
text = "hej med dig der xylophon"

# Solution with dict having all letters as keys
alphabet = 'qwertyuiopasdfghjklzxcvbnm'
word_dict1 = {}
for letter in alphabet:
    word_dict1[letter] = []
for word in text.split():
    word_dict1[word[0]].append(word)
print(word_dict1)

# Solution with dict having only keys for letters that are in the text
word_dict2 = {}
for word in text.split():
    if word[0] not in word_dict2:
        word_dict2[word[0]] = []
    word_dict2[word[0]].append(word)
print(word_dict2)
```

alpha2phone

```
phone_dict = {'A': 2, 'B': 2, 'C': 2,
        'D': 3, 'E': 3, 'F': 3,
        'G': 4, 'H': 4, 'I': 4,
        'J': 5, 'K': 5, 'L': 5,
        'M': 6, 'N': 6, 'O': 6,
        'P': 7, 'Q': 7, 'R': 7, 'S': 7,
        'T': 8, 'U': 8, 'V': 8,
        'W': 9, 'X': 9, 'Y': 9, 'Z': 9}

def alpha2phone(phone_input):
    phone_output = ""
    for character in phone_input:
        if character in phone_dict:
            phone_output += str(phone_dict[character])
        else:
            phone_output += character
    return phone_output

print(alpha2phone("4525DTU1"))
```