CHVorl1

November 12, 2016

1 Vorlesung 1: Individuum und Art

1.1 Wissenschaftssprache der Forschungsdomäne

1.2 Hybride Forschungspublikation von wissenschaftlichen Narrativen

Notebooks in Jupyter
http://jupyter.org/
Lokale Installation des gesamten Pakets
https://www.continuum.io/downloads

1.3 Python: Sprache für computational humanities

Vorlesung: https://computational-humanities.github.io/vorlesung/
Tutorial: - https://py-tutorial-de.readthedocs.io/de/python-3.3/index.html Lectures: - https://github.com/rajathkumarmp/Python-Lectures - https://www.hdm-stuttgart.de/~maucher/Python/html/

1.4 Grundelemente der Sprache

1.4.1 Module und Bibliotheken deklarieren

https://www.hdm-stuttgart.de/~maucher/Python/html/Module.html

```
In [1]: # importieren von Modulen/Bibliotheken
```

1.4.2 Objekte

Typen von Daten https://www.hdm-stuttgart.de/~maucher/Python/html/Datentypen.html

```
In [2]: # strings
        "Dieses ist eine Zeichenfolge"
        'Auch dieses ist eine Zeichenfolge'
        # number
        2.12
        # Liste
        ["Zeichen1","Z2","Z3"]
        # dictionaries
        {"eins":1,"zwei":2}

Out[2]: {'eins': 1, 'zwei': 2}
```

1.4.3 Variablen als Namen

"=" als Zuordnung

```
In [3]: # strings
        a="Dieses ist eine Zeichenfolge"
        # number
        b=2.12
        # Liste
        c=["Zeichen1","Z2","Z3"]
        # dictionaries
        d={"eins":1,"zwei":2}
```

```
In [4]: print(d)
```

```
{'zwei': 2, 'eins': 1}
```

```
In [5]: a.count("e")
```

```
Out[5]: 7
```

```
In [6]: "ie" in a
```

```
Out[6]: True
```

1.4.4 Funktionen

```
In [7]: len(c)
```

```
Out[7]: 3
```

1.4.5 Methoden der Objekte

```
In [8]: # nutzen Sie TAB, um nach dem Punkt die zulässigen Methoden anzuzeigen
        d["eins"]
```

```
Out[8]: 1
```

```
In [9]: c.sort()
        c
```

```
Out[9]: ['Z2', 'Z3', 'Zeichen1']
```

1.4.6 Kontrollstrukturen der Verfahrensabläufe

https://www.hdm-stuttgart.de/~maucher/Python/html/Kontrollstrukturen.html #### Iteratoren

2 Arten von Individuen: Dataframes

```
In [10]: import json
         import pandas as pd
         import re
```

```
In [11]: with open('chapter1.json') as json_data:
             PoleisRawData = json.load(json_data)
         PoleisKeyList = list(PoleisRawData)
```

```
In [12]: PoleisRawData.keys()
```

```
Out[12]: dict_keys(['19. Henna ', '6. Adranon ', '50. (Tyrrhenoi)', '34. Lipara ',
```

```
In [13]: PoleisKeyList[1]
```

```
Out[13]: '6. Adranon '
```

```
In [14]: type(PoleisRawData)
```

```
Out[14]: dict
```

```
In [15]: PoleisRawData[PoleisKeyList[2]]
```

```
Out[15]: 'Map 47.  Unlocated.  Type:  C: .The  Tyr- rhenoi  are  known exclusively
```

```
In [16]: for i in PoleisKeyList:
             if "Megara" in PoleisRawData[i]:
                 print(PoleisRawData[i])
```

```
(Leontinos) Map  47.  Lat. 37.15,long.  15.00. Size of  territory: 4.Type:A: .The t
```

```
                                            12. Alontion   \
0  (Alontinos) Map  47.  Lat. 38.05,long.  14.40...

                                           13. Apollonia   \
0  (Apolloniates) Map  47.  Lat. 38.00,long. 14.3...

                                              14. Engyon   \
0  (Engyinos)  Map  47.  Lat. 37.45,long.  14.35...

                                              15. Euboia   \
0  (Euboeus)  Map  47.  Unlocated.  Type:  C: .Th...

                                             16. Galeria   \
0  (Galarinos) Map  47.Unlocated  (Manni  ( 1981)...

                                                17. Gela   \
0  (Geloios,  Geloaios) Map  47.  Lat. 37.05,long...

                                             18. Heloron   \
0  (Ailoros) Map  47.  Lat. 36.50,long.  15.05.  ...

                                               19. Henna   \
0  (Hennaios) Map  47.  Lat. 37.35,long.  14.15. ...

                         ...                               \
0                        ...

                                          47. Syrakousai   \
0  (Syrakosios) Map  47.  Lat. 37.05,long. 15.15...

                                         48. Tauromenion   \
0  (Tauromenitas) Map  47.  Lat. 37.50, long. 15...

                                            49. Tyndaris   \
0  (Tyndarites) Map  47.  Lat. 38.10,long.  15.05...

                                            5. Abakainon   \
0  (Abakaininos) Map  47.  Lat. 38.05,long. 15.05...

                                          50. (Tyrrhenoi)           51. Zankle
0  Map 47.  Unlocated.  Type:  C: .The  Tyr- rhen...  (Zanklaios)/Messana

                                              6. Adranon   \
0  (Adranites) Map  47.  Lat. 37.40,long.  14.50...

                                              7. Agyrion   \
0  (Agyrinaios) Map  47.  Lat. 37.40,long.  14.30...
```

```
                                          8. Aitna   \
0  (Aitnaios) Map  47.Location  of  Aitna  I  as ...

                                        9. Akragas
0  (Akragantinos) Map  47.  Lat. 37.20,long.  13...

[1 rows x 47 columns]
```

```
In [19]: #Namen der Poleis als Index benutzen
         dfPoleis = dfPoleis.transpose()
         dfPoleis
```

```
Out[19]:                                                                    0
         10. Akrai          (Akraios) Map  47.  Lat. 37.05,long.  14.55.  ...
         11. Alaisa         (Alaisinos) Map  47.  Lat. 38.00,long.  14.15...
         12. Alontion       (Alontinos) Map  47.  Lat. 38.05,long.  14.40...
         13. Apollonia      (Apolloniates) Map  47.  Lat. 38.00,long. 14.3...
         14. Engyon         (Engyinos)  Map  47.  Lat. 37.45,long.  14.35...
         15. Euboia         (Euboeus)  Map  47.  Unlocated.  Type:  C: .Th...
         16. Galeria        (Galarinos) Map  47.Unlocated  (Manni  ( 1981)...
         17. Gela           (Geloios,  Geloaios) Map  47.  Lat. 37.05,long...
         18. Heloron        (Ailoros) Map  47.  Lat. 36.50,long.  15.05.  ...
         19. Henna          (Hennaios) Map  47.  Lat. 37.35,long.  14.15. ...
         20. Herakleia 1    (Herakleotes) Map  47.  Lat. 37.25,long. 13.15...
         21. Herakleia 2   Map 47.Unlocated  site  in  western  Sicily, i...
         22. Herbes(s)os    (Herbessinos) Map  47.  Unlocated,  but presum...
         23. Herbita        (Herbitaios) Map  47.Unlocated  (cf.  C.  Boeh...
         24. Himera         (Himeraios) Map  47.  Lat. 37.55,long.  13.50...
         25. Hippana        (Hipanatas)  Map  47.  Lat. 37.40,long.  13.25...
         26 *Imachara       (Imacharaios) Map  47.  Unlocated.  Barr. tent...
         27. Kallipolis     (Kallipolites)  Map  47.  Unlocated.  Type:  A...
         28. Kamarina       (Kamarinaios) Map  47.  Lat. 36.50,long. 14.25...
         29. Kasmenai       (Kasmenaios) Map  47.  Lat. 37.05,long. 14.50...
         30. Katane         (Katanaios) Map  47.  Lat. 37.30,long.  15.05...
         31. Kentoripa      (Kentoripinos) Map  47.  Lat. 37.35,long. 14.4...
         32. Kephaloidion   (Kephaloiditas) Map  47.  Lat. 38.00, long. 14...
         33. Leontinoi      (Leontinos) Map  47.  Lat. 37.15,long.  15.00...
         34. Lipara         (Liparaios) Map  47.  Lat. 38.30  long. 14.55...
         35. *Longane       (Longenaios) Map  47.  Lat. 38.05,long. 15.10...
         36. Megara         (Megareus) Map  47.  Lat. 37.10,long.  15.10. ...
         37. Morgantina     (Morgantinos) Map  47.  Lat. 37.25,long. 14.30...
         38. Mylai          (Mylaios) Map  47.  Lat. 38.15,  long  15.15. ...
         39. Mytistratos    (Mytiseratinos) Map  47.  Lat. 37.35,long. 14...
         40. Nakone         (Nakonaios)  Map  47.  Unlocated  (Tegon ( 199...
         41. Naxos          (Naxios) Map  47.  Lat. 37.50,long.  15.15.  S...
         42. Petra          (Petrinos) Map  47.  Unlocated  (cf.  Bejor  (...
         43. Piakos         (Piakinos) Map  47.  Unlocated,  but  possibly...
```

```
44. Selinous        (Selinousios) Map  47.  Lat. 37.35,long.  12.5...
45. (Sileraioi)     Map 47.  Unlocated  (cf.  Manni  ( 1981)  225)...
46. (Stielanaioi)   Map 47.  Lat. 37.10,long.  14.55:  the  loca- ...
47. Syrakousai      (Syrakosios) Map  47.  Lat. 37.05,long. 15.15...
48. Tauromenion     (Tauromenitas) Map  47.  Lat. 37.50, long. 15...
49. Tyndaris        (Tyndarites) Map  47.  Lat. 38.10,long.  15.05...
5. Abakainon        (Abakaininos) Map  47.  Lat. 38.05,long. 15.05...
50. (Tyrrhenoi)     Map 47.  Unlocated.  Type:  C: .The  Tyr- rhen...
51. Zankle                                             (Zanklaios)/Messana
6. Adranon          (Adranites) Map  47.  Lat. 37.40,long.  14.50...
7. Agyrion          (Agyrinaios) Map  47.  Lat. 37.40,long.  14.30...
8. Aitna            (Aitnaios) Map  47.Location  of  Aitna  I  as ...
9. Akragas          (Akragantinos) Map  47.  Lat. 37.20,long.  13...
```

```
In [20]: # Umnennen der Spalte von 0 zu 'full_text'
         dfPoleis = dfPoleis.rename(columns={0: 'full_text'})

In [21]: # Extrahiere Name der Polis aus Index
         dfPoleis['city'] = [x[3:] for x in dfPoleis.index]

In [22]: # Extrahiere Nummer des Polis Eintrags
         dfPoleis['city_index'] = [int(re.findall('\d{1,2}', x)[0]) for x in dfPole

In [23]: # Sortiere Dataframe nach Eintragsnummer der Polis
         dfPoleis = dfPoleis.sort_values(by='city_index')

In [24]: # Nutze neuen Index (startet bei 0)
         dfPoleis = dfPoleis.reset_index(drop=True)

In [25]: dfPoleis['full_text'].iloc[0]

Out[25]: '(Abakaininos) Map  47.  Lat. 38.05,long. 15.05.  Size  of  territory:  ?

In [ ]:
```