Takács Edit
edit.takacs@gmail.com
GitHub:
https://github.com/gtakacse/CS_class_scripts

Emberi Erőforrások Minisztériuma

Az Emberi Erőforrások Minisztériuma Új Nemzeti Kiválóság Programjának támogatásával készült

Programozás nyelvészeknek (9.-10. óra)

Ismétlés
re modul
xml, html formátumok
nltk modul

Emlékeztetőül 1.

- adattípusok (int, float, bool, none, string, list, dict)
- kontroll struktúrák
  - if, if, if
  - if, elif, else
  - beágyazott struktórák pl.:

1. if
    1. if
        1. if
        2. else
    2. else
2. else

Emlékeztetőül 2.

- ciklusok
  1. for ciklus

```
1.  for elem in sorozat:
2.          csinálj az elemmel valamit
```

  2. while ciklus

```
1.  while kondíció igaz:
2.          csinálj valamit
```

Emlékeztetőül 3.

- saját funkciók írása

```
1.  def add_a_b(a,b):
2.          c = a + b
3.          return c
4.
5.  #  használat
6.  osszeg = add_a_b(4,3) # osszeg értéke 7
```

Emlékeztetőül 4.

- fájlok beolvasása

```
1. path = 'some_file.txt' #a fájl helye
2. f = open(path, 'r', encoding = 'utf-8'
3. data = f.read() # egész dokumentum
4. rows = f.readlines() # összes sor
5. row = f.readline() # egy sor
6. f.close() # fájl bezérása
```

- fájlok írása

```
1. path = 'some_file.txt' # fájl neve és helye
2. f  = open(path, 'w', encoding='utf-8')
3. f.write('line\n') # egy sor írása
4. f.close() # fájl bezárása
```

Regex - intelligens keresés Pythonban

- import re
- match = re.search(pattern, text)
- match.group()
- match.span()
- all_matches = re.findall(pattern, text) #returns a list of all matches
- new_string = re.sub(pattern, replacement, text)

Regex patterns

- b,U,3 - egyszerű karakterek
- ^ $ * + ? { [ ] \ | ( ) - speciális krakterek
- . - bármilyen karakter kivéve \n
- ^ - string eleje
- $ - string vége
- \ - ha a kötvekező karakter spciális, felfüggeszti a speciális jelentését (pl. \. a pont)
- * - előtte lévő karakter 0 vagy több ismétlése
- + - előtte lévő karakter 1 vagy több ismétlése

Regex - intelligens keresés Pythonban

- import re
- documentáció: https://docs.python.org/3/library/re.html
- cheat sheet: https://www.debuggex.com/cheatsheet/regex/python
- online editor: https://regex101.com
- tutorial: https://developers.google.com/edu/python/regular-expressions

Html dokumentumok letöltése és feldolgozása

- Példa

```
<html>
  <head>
    <title>Ez a dokumentum címe</title>
    <meta charset = 'utf-8'>
  </head>
  <body>
  Ez a dokumentum törzse
  </body>
</html>
```

Html letöltése

- from urllib import request
- url = 'http://www.origo.hu'
- html = request.urlopen(url).read() .decode('iso-8859-2')
- A kódolást ellenőrizni kell a html fájlban. (<meta charset = …>)

Html feldolgozása

- a html egyszerű szöveg >> ugyanúgy kereshető, mint bármelyik string (re modul, str method-ok)
- ha a html struktúráját is fel akarjuk használni:
- from bs4 import BeautifulSoup
- soup = BeautifulSoup(html, 'html.parser')
- dokumentáció: https://www.crummy.com/software/BeautifulSoup/bs4/doc/

NLTK - Python könyvtár nyelvészeti elemzésekhez

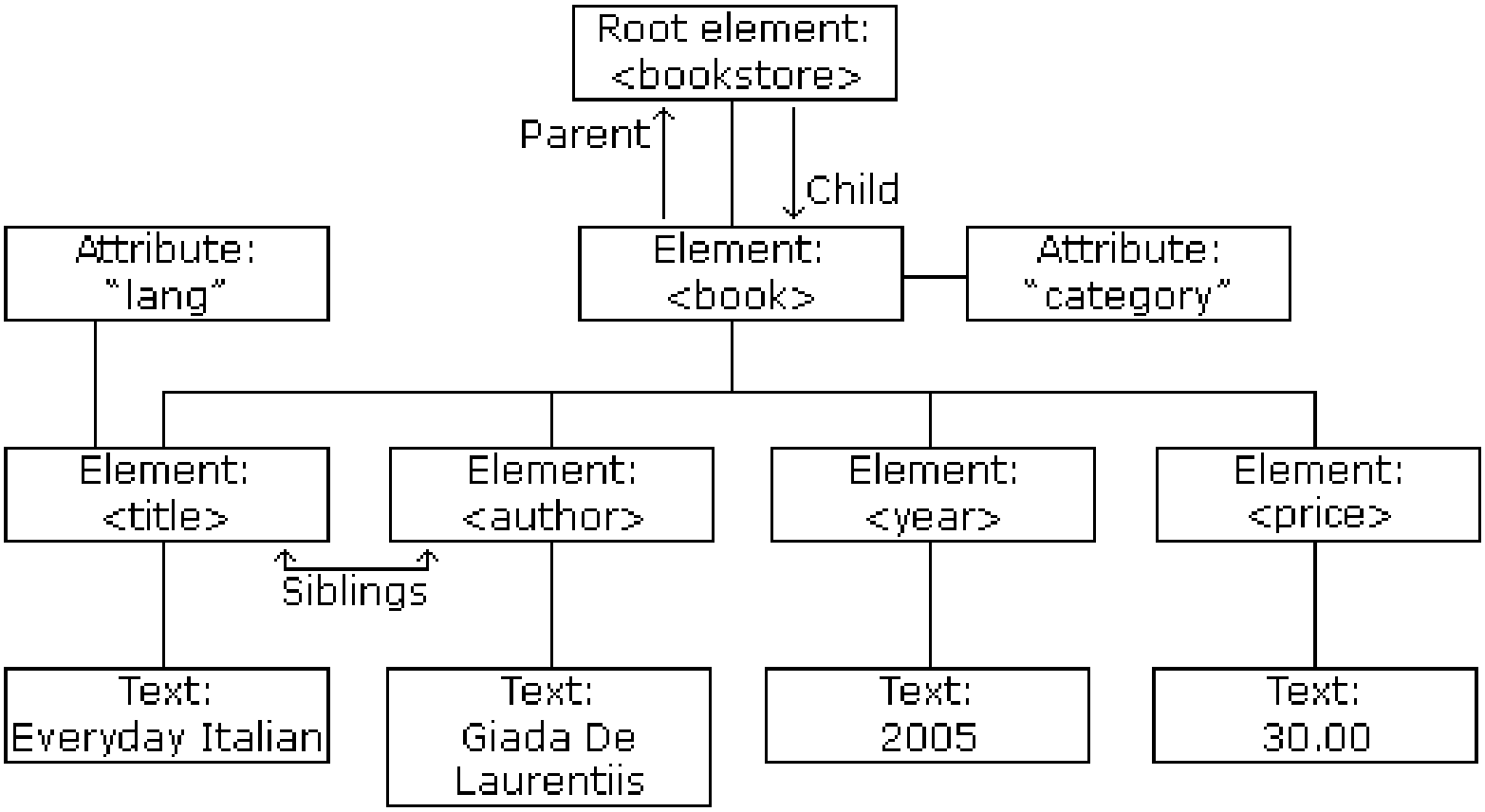
- dokumentáció: http://www.nltk.org/index.html
- könyv: http://www.nltk.org/book/
- import nltk

XML fájlok kezelése

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>

  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
</bookstore>
```

XML struktúra



XML - különleges karakterek

&lt;	<	less than
&gt;	>	greater than
&amp;	&	ampersand
&apos;	'	apostrophe
&quot;	"	quotation mark