

Programação

Diferentes tipos de ficheiros

Fátima Leal

DCT DEPARTAMENTO CIÊNCIA
E TECNOLOGIA

Até agora ...

- Módulos em Python
 - Funções recursivas
 - Funções de ordem superior
 - Geradores e compreensões de listas
 - Escrita e leitura de ficheiros
-
- Nesta aula:
 - Diferentes tipos de ficheiros

Tipos de ficheiros

- Tipos de ficheiros mais populares
 - .txt – ficheiros de texto sem nenhuma configuração em concreto
 - .csv – ficheiro com dados separados por vírgulas
 - html – páginas web
 - .zip – ficheiros comprimidos
- Módulos
 - csv
 - urllib
 - json

Leitura de ficheiros csv

```
import csv

with open("dados.csv") as fin:
    csvFile = csv.reader(fin)
    for linha in csvFile:
        print(linha)
```

- Para ficheiros csv o python disponibiliza o módulo csv
- O `csv.reader` retorna os dados de cada linha numa lista
- **Exercício:** o ficheiro `dados.csv` contém as classificações de 13 alunos. Calcula a média de cada aluno sendo que os 5 momentos de avaliação têm os seguintes pesos: 25 %, 15 %, 10 %, 30 % e 20 %.

Leitura de ficheiros csv

```
import csv

pesos = [0.25, 0.15, 0.10, 0.30, 0.20]
notas=[]
with open("dados.csv") as fin:
    csvFile = csv.reader(fin)
    for linha in csvFile:
        linha = [float(x) for x in linha]
        soma=0
        for x in range(len(pesos)):
            soma+=linha[x]*pesos[x]
        notas.append(soma)

print(notas)
```

Leitura de ficheiros csv

Possível solução com compreensão de listas

```
import csv

pesos = [0.25, 0.15, 0.10, 0.30, 0.20]
notas=[]
with open("dados.csv") as fin:
    csvFile = csv.reader(fin)
    for linha in csvFile:
        linha = [float(x) for x in linha]
        product = [x * y for x, y in zip(linha, pesos)]
        notas.append(sum(product))
print(notas)
```

Leitura de páginas web

```
import urllib.request as url

site = url.urlopen("http://www.upt.pt")
mb = site.read()
mc = mb.decode("utf8")
print(mc)
site.close()
```

- Para descarga dados provenientes da web o Python disponibiliza o pacote urllib
- Este módulo permite a manipulação de páginas web
- O código html retornado é tratado como strings

Leitura ficheiros JSON e XML

- JSON (JavaScript Object Notation)
- XML (Extensible Markup Language)

Ficheiros JSON e XML

JSON

```

1 {
2   "sessionStart": "16-03-18-12-33-09",
3   "sessionEnd": "16-03-18-12-33-12",
4   "mapName": "TestMap",
5   "logSections": [{
6     "sector": {
7       "x": 2.0,
8       "y": -1.0,
9       "z": 0.0
10    },
11    "logLines": [{
12      "time": 37.84491729736328,
13      "state": 0,
14      "action": 1,
15      "playerPosition": {
16        "x": 24.560218811035158,
17        "y": -8.940696716308594e-8,
18        "z": 3.3498525619506838
19      },
20      "cameraRotation": {
21        "x": 0.24549755454063416,
22        "y": 0.017123013734817506,
23        "z": 0.031348951160907748,
24        "w": -0.9687389135360718
25      }
26    },
27    ...

```

XML

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <root>
3   <sessionStart>16-03-18-12-33-09</sessionStart>
4   <sessionEnd>16-03-18-12-33-12</sessionEnd>
5   <mapName>TestMap</mapName>
6   <logSections>
7     <sector>
8       <x>2</x>
9       <y>-1</y>
10      <z>0</z>
11    </sector>
12    <logLines>
13      <time>37.84491729736328</time>
14      <state>0</state>
15      <action>1</action>
16      <playerPosition>
17        <x>24.560218811035156</x>
18        <y>-8.940696716308594e-8</y>
19        <z>3.3498525619506836</z>
20      </playerPosition>
21      <cameraRotation>
22        <x>0.24549755454063416</x>
23        <y>0.017123013734817505</y>
24        <z>0.031348951160907745</z>
25        <w>-0.9687389135360718</w>
26      </cameraRotation>
27    ...

```

- ficheiros de texto key-value hierárquicos
- formato “standard” para troca de dados semi-estruturados entre aplicações

Leitura ficheiros JSON

```
import json
with open('ipma.json', 'r') as f:
    dict = json.load(f)
    print(dict)
```

- JSON = dicionários e listas aninhados uns nos outros, cujas folhas são strings, números ou booleanos
- O ficheiro `ipma.json` contém a previsão meteorológica de 5 dias para o Porto publicada pelo IPMA
- **Exercício:** Utilizando este ficheiro JSON calcula a média da probabilidade da precipitação

Leitura ficheiros JSON

```
import json
import statistics as sts

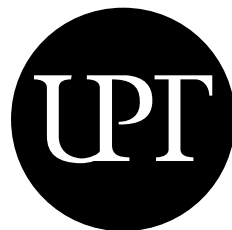
with open('ipma.json','r') as f:
    dict = json.load(f)

dados = dict['data']
print(dados)

pb = [d['precipitaProb'] for d in dados]
pb = [float(x) for x in pb]
print("Media: ", sts.mean(pb))
```

Exercícios

- Encontra os dias cuja probabilidade de precipitação é maior que 50 %.
- Verifica a diferença entre a temperatura máxima e mínima
- Qual o dia em que essa diferença é mais acentuada?
- Tendo em conta a probabilidade da precipitação, faz um gráfico que mostra a sua evolução.
 - Utiliza a biblioteca `matplotlib.pyplot`
 - Para o desenho do gráfico utiliza o método `plot` da biblioteca que importaste
 - Explora os métodos do `matplotlib`
 - Experimenta traçar um histograma e um scatter plot



UNIVERSIDADE
PORTUCALENSE

Do conhecimento à prática.