



Práce s XLSX tabulkami

Workshop  Sobota 16. listopadu 2019  Brno

Motivace

- Proč XLSX?
 - v současnosti nejpoužívanější formát pro uživatelskou práci s tabulkami
 - snadné nastavení formátování, interaktivní zobrazení (šíře tabulky, filtry, ..)
 - možnost využití dat pro další zpracování (přenos dat, kontingenční tabulky, ..)
- Proč automatizované zpracování?
 - pravidelné reporty mají konstantní podobu
 - zpracování a využití dat současně
 - data z databáze, ze souboru, ze vstupu
 - vypočtená data
 - hromadné zpracování velkého množství souborů



Použité knihovny

- openpyxl
 - + možnosti formátování
 - rychlost zpracování
- PyExcellent
 - + rychlost zpracování
 - možnosti formátování



Začínáme s openpyxl

1. naklonuj si: `git clone https://github.com/luipenox/pytable`



Co zvládneme s openpyxl

- vkládání dat
- nastavení výšky řádků, šířky sloupců
- nastavení zarovnání, odsazení
- nastavení písma (font, barva, tučné, kurzíva, ...)
- nastavení formátu dat
- nastavení barvy pozadí
- vytvářet grafy
- vytvářet vzorce
- nastavovat podmíněné formátování



Co nezvládneme s openpyxl

- vytvářet „tabulky“ (paradoxně)
- nastavovat automatickou šířku sloupce



No!

Základy práce s openpyxl

- hlavní objekt jako ekvivalent souboru

```
from openpyxl import Workbook  
workbook = Workbook()
```

- práce s daty na samostatných listech

```
worksheet = workbook.active
```

nebo

```
worksheet = workbook.create_sheet('Název listu')
```



Uložení souboru

- do souboru zapisujeme, je potřeba ho tedy vždy uzavřít
- z tohoto důvodu je nutné ošetřit zpracování kódu
- použijeme blok `try` a `finally`

```
try:
```

```
...
```

```
finally:
```

```
    workbook.save(filename='nazev_souboru.xlsx')
```



Základy práce s listem

- nastavení názvu listu
`worksheet.title = 'Nový název listu'`
- identifikace podle názvu listu
`worksheet = workbook['Název listu']`
- listy v sešitu
`workbook.sheetnames`
- procházení *for* cyklem
`for worksheet in workbook:
 print(worksheet.title)`



Základy práce s buňkou

- nastavení hodnoty buňky
`worksheet['A1'] = 'Budiž světlo'`
`worksheet['A1'].value = 'Budiž světlo'`
- přečtení hodnoty z buňky
`worksheet['A1'].value`
- adresování pomocí **cell**
`worksheet.cell(row=2, column=3, value=6)`



Práce s více buňkami

- výběr rozsahu více buněk
`cell_range = worksheet['A1:F5']`
- výběr celých sloupců
`column = worksheet['D']`
- výběr celých řádků
`row = worksheet[8]`
- výběr více sloupců
`columns = worksheet['B:D']`
- výběr více řádků
`rows = worksheet[2:6]`



Procházení buněk

- procházení *for* cyklem

```
for row in worksheet['A1:F5']:  
    for cell in row:  
        print(cell.value)
```

```
for row in worksheet.iter_rows(min_row=1, max_col=2, max_row=3):  
    for cell in row:  
        print(cell.value)
```

```
for column in worksheet.iter_cols(min_row=1, max_col=2, max_row=3):  
    for cell in column:  
        print(cell.value)
```



Rozsah buněk v listu

- automatické rozpoznání dle poslední buňky

```
for row in worksheet.rows:  
    for cell in row:  
        print(cell.value)
```

```
for column in worksheet.column:  
    for cell in column:  
        print(cell.value)
```



Read-only

- v procházení přímo values nebo v iteraci s values_only

```
for row in worksheet.values:  
    for value in row:  
        print(value)
```

```
for row in worksheet.iter_rows(min_row=1, max_col=2, max_row=3, values_only=True):  
    for value in row:  
        print(value)
```

- read-only nefunguje při iteraci přes sloupce (kvůli výkonu)



Zápis pomocí rozsahu

- protože je ve *for* cyklu definována buňka, je třeba přistupovat přes *value*

```
for row in worksheet.rows:  
    for cell in row:  
        cell.value = 'Něco vážně důležitého'
```



Čtení dat ze souboru

- importujeme potřebnou funkci a ještě funkci na zjištění písmena sloupce

```
from openpyxl import load_workbook
from openpyxl.utils import get_column_letter
```
- načteme soubor podle jména

```
workbook = load_workbook(filename='demo_openpyxl.xlsx')
```
- vybereme list dle jména

```
worksheet = workbook['DEMO (openpyxl)']
```
- vypíšeme označení buňky a hodnoty ze zvoleného rozsahu

```
for row in worksheet['C1':'D5']:
    for value in row:
        print(f"{get_column_letter(cell.column)}{cell.row}:", {cell.value})
```



Cvičení dělá mistra

- nyní jsme připraveni na první cvičení
 - prošmejdi adresář *"lesson_01"*
 - najdi soubor s názvem *"Seznam pracovníků - aktualizováno k 14.11.2019.xlsx"*
 - ulož do slovníku pracovníky podle osobních čísel pomocí funkce `load_workbook`
 - najdi soubor s názvem *"seznam_oddeleni.csv"*
 - ulož do slovníku čísla oddělení a jejich názvy pomocí knihovny `csv`
 - najdi soubor s názvem *"log_pristup_trezor.csv"*
 - pomocí knihovny `csv` načítej jednotlivé řádky
 - pro každý řádek kontroluj čas přístupu
 - pokud je čas přístupu před 6:00 nebo po 22:00, řádek zapiš do souboru*
 - formát výpisu:
 - osobní číslo, příjmení, jméno, název oddělení, datum přístupu, čas přístupu

*nezapisuj řádky, u kterých přistupoval ředitel (soubor je určen pro ředitele)






Čas na oběd

Formátování

- co standardně v x/sx formátovat:
 - výška řádku/šířka sloupce
 - formát dat (obecný, text, číslo, datum)
 - formát písma (font, barva, tučně, kurzíva, podtržení)
 - zarovnání v buňce (horizontální, vertikální, odsazení)
 - ohraničení buňky
 - podbarvení buňky
- sloučení/rozdělení buněk



```
C:\>format
```

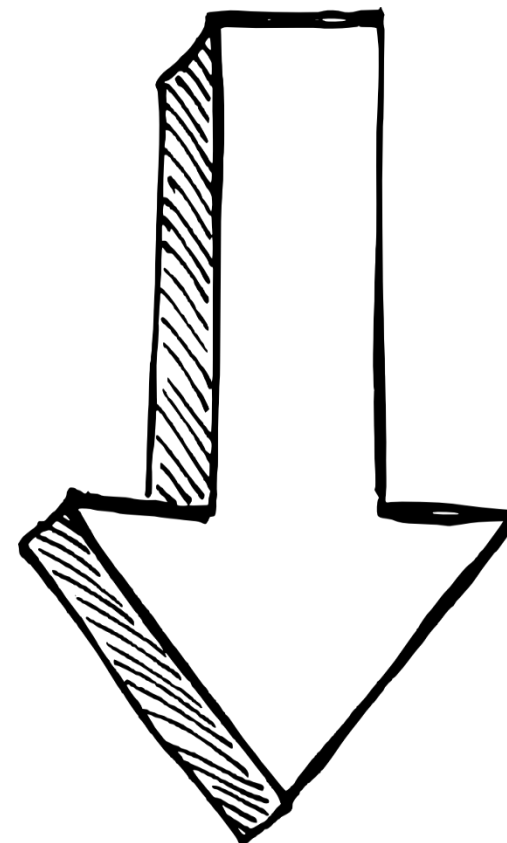
Výška řádku / šířka sloupce

- nastavení výšky řádku:
 - `worksheet.row_dimensions[no_row].height = height`
- nastavení šířky sloupce:
 - `worksheet.row_dimensions[letter_column].width = width`
- pro identifikaci sloupce můžeme použít i čísla:
 - `letter = get_column_letter[no_column]`



Formát dat

- nastavení formátu:
 - obecný formát:
 - `cell.number_format = "General"`
 - číselné formáty:
 - `cell.number_format = "0.00"`
 - `cell.number_format = "0"`
 - textový formát:
 - `cell.number_format = "@"`
 - formát data:
 - `cell.number_format = "dd.mm.yyyy"`



Formát písma

- nastavení formátu písma:
 - importujeme:
 - `from openpyxl.styles import Font`
 - nastavení písma a velikosti:
 - `cell.font = Font(name='Calibri', size=11)`
 - další nastavení:
 - `cell.font = Font(bold=True, italic=False, underline='none')`
 - nastavení barvy:
 - `cell.font = Font(color='FF00FF')`

Aa Bb Cc Dd Ee

Ff Gg Hh Ii Jj Kk

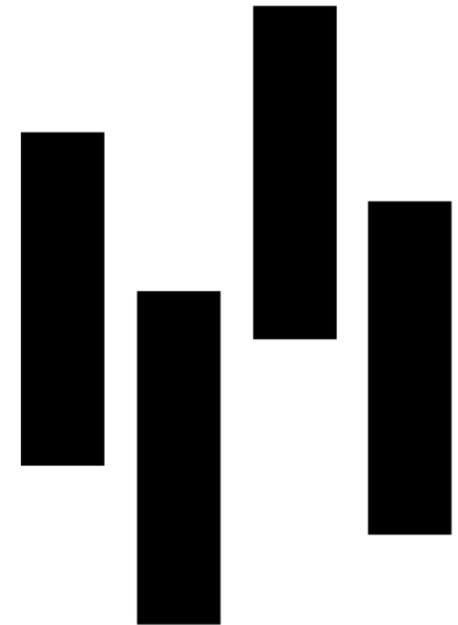
Ll Mm Nn Oo Pp

Qq Rr Ss Tt Uu Vv

Ww Xx Yy Zz

Zarovnání v buňce

- nastavení zarovnání:
 - importujeme:
 - `from openpyxl.styles import Alignment`
 - horizontální zarovnání:
 - `cell.alignment = Alignment(horizontal='left')`
 - vertikální zarovnání:
 - `cell.alignment = Alignment(vertical='top')`
 - odsazení:
 - `cell.alignment = Alignment(indent=1)`
 - rotace textu:
 - `cell.alignment = Alignment(text_rotation=90)`



Ohraničení buňky

- nastavení ohraničení buňky:
 - importujeme:
 - `from openpyxl.styles import Border, Side`
 - nastavení ohraničení dle stran:
 - `cell.border = Border(left=Side(border_style="thin", color='FF00FF'), right=...)`
- pro každou stranu je možné volit jiné ohraničení
- příklady `border_style`:
 - `None`
 - `'thin'`
 - `'dashed'`
 - `'double'`
 - `'medium'`
 - `'thick'`



Podbarvení buňky

- nastavení podbarvení buňky:
 - importujeme:
 - `from openpyxl.styles import PatternFill`
 - nastavení ohraničení dle stran:
 - `cell.fill = PatternFill(bgColor="FF00FF", fill_type = "solid")`



Sloučení / rozdělení buněk

- sloučení buněk:
 - `ws.merge_cells('B3:D5')`
- rozdělení buněk:
 - `ws.unmerge_cells('B3:D5')`



At' to nějak vypadá

- nyní jsme připraveni na druhé cvičení
 - prošmejdi adresář *"lesson_02"*
 - najdi soubor s názvem *„seznam_firem.csv“*
 - pomocí knihovny **csv** načítej jednotlivé řádky
 - pro každý řádek vytvoř jednotný soubor s informacemi o daňovém subjektu
 - obsah souboru naformátuj s využitím již zmíněného
 - název souboru a název listu bude stejný jako číslo daňového subjektu
 - soubory ulož do samostatné složky

