



Univerzita Palackého  
v Olomouci



---

Univerzita Palackého  
v Olomouci

# **Práce se soubory Excel a výsledek v HTML**



## Cíl programu

- Načíst data z Excelu
- Rozdělit zákazníky podle splatnosti jejich faktur
- Vytvořit přehledné tabulky a zobrazit je jako HTML stránku
- Využití programu
  - Vytvoření úkolů v Excelu a jejich následné rozdělení dle data
  - Lze modifikovat tento program a nerozdělovat data podle aktuálního datumu



## Použité knihovny

- **openpyxl**
  - Načtení Excel souborů
- **pandas**
  - Práce s tabulkovými daty
- **Datetime**
  - Práce s daty a časy (např. dnešní datum, porovnání dat, ...)
- **jinja2**
  - Tvorba HTML šablon
- **IPython.display**
  - Zobrazení HTML v Jupyter Notebooku
- **webbrowser**
  - V případě spouštění kódu v integrovaném vývojovém prostředí (např. PyCharm) místo IPython.display využijeme knihovnu webbrowser
  - Otevře výsledek v prohlížeči



## Načítání dat

- Kód

```
excel_file = "zakaznici.xlsx"  
df = pd.read_excel(excel_file, engine="openpyxl")
```

- Načítáme data z Excel souboru *zakaznici.xlsx*
- Pomocí knihovny *pandas* vytvoříme datový rámec (*DataFrame*)
  - tabulka podobná Excelu



## Práce s daty

- Převede sloupce *Datum splatnosti* na datum → umožnění porovnání data splatnosti s aktuálním datem
- Užijeme knihovnu *datetime* na zjištění aktuálního data a času
- Následně data rozdělíme:
  - *Před splatností* – faktury s budoucím datem
  - *Po splatnosti* – faktury s minulým datem splatnosti
  - *Den splatnosti* – faktury s datem splatnosti právě dnes

```
# Datum splatnosti převedeno na datový typ datetime
df["Datum splatnosti"] = pd.to_datetime(df["Datum splatnosti"])

# Dnešní datum
dnes = datetime.now()

# Rozdělení na tři tabulky
pred = df[df["Datum splatnosti"] > dnes]
po = df[df["Datum splatnosti"] < dnes]
den = df[df["Datum splatnosti"] == dnes]
```



# Tvorba a zobrazení HTML

- Strukturu HTML stránky vytvoříme pomocí *jinja2*
- *Dynamiccké vkládání data z tabulek do stránky*

```
html_template = """
<!DOCTYPE html>
<html>
<head>
    <title>Tabulky</title>
</head>
<body>
    <h1>Po splatností:</h1>
    {{ po.to_html(classes="table table-striped", index=False) | safe }}
    <h1>Den splatnosti:</h1>
    {{ den.to_html(classes="table table-striped", index=False) | safe }}
    <h1>Před splatností:</h1>
    {{ pred.to_html(classes="table table-striped", index=False) | safe }}
</body>
</html>
"""
```



# Šablona HTML

- Definuje strukturu výstupní webové stránky
- Šablonovací jazyk *jinja2* – umožní vkládat dynamická data
- Dokument je ve formátu HTML 5 `<!DOCTYPE html>`
- Kořenové elementy
  - `<html>`, `<head>...</head>`, `<body>...</body>`, `</html>`
  - Html – začátek dokumentu `<html>`
  - Head – obsahuje metadata (např. název stránky)
    - Definuje název zobrazený na kartě prohlížeče (u nás **Tabulky**)
- Body – hlavní obsah stránky

```
<head>  
  <title>Tabulky</title>  
</head>
```





# Šablona HTML – obsah stránky

- Nadpisy jednotlivých sekcí

```
<h1>Po splatností:</h1> <h1>Den splatnosti:</h1> <h1>Před splatností:</h1>
```

- Dynamické vkládání tabulek

```
{{ po.to_html(classes="table table-striped", index=False) | safe }}
```

- `{{...}}` – syntax pro vkládání dat
- `po.to_html()` – převede datový rámec (*DateFrame*) na HTML tabulku
- `classes="table table-striped",` - přidává CSS třídy
  - *Pomocí CSS tříd upravujeme vzhled tabulky (v našem případě bez úpravy)*
- `index=False` – skryje číslování
- `| safe` – říká šablonovacímu systému, že HTML kód je bezpečný

```
<body>
  <h1>Po splatností:</h1>
  {{ po.to_html(classes="table table-striped", index=False) | safe }}
  <h1>Den splatnosti:</h1>
  {{ den.to_html(classes="table table-striped", index=False) | safe }}
  <h1>Před splatností:</h1>
  {{ pred.to_html(classes="table table-striped", index=False) | safe }}
</body>
```



## Nahrazení dat v šabloně

- Vytvoříme šablonu z textu HTML
- Nahradíme data v šabloně
  - V proměnných *po*, *den* a *pred* jsou rozdělena data zákazníků podle splatnosti
  - `template.render()` – nahrazení dynamických částí šablony skutečnými daty

```
# Vytvořit šablonu
template = Template(html_template)

# Nahrazení dat v šabloně
html_output = template.render(po=po, den=den, pred=pred)
```



## Zobrazení HTML

- Jupyter Notebook
  - Zobrazí nám vytvořenou HTML stránku přímo v Jupyter Notebooku

```
# Zobrazit výstup přímo v Jupyter Notebooku  
display(HTML(html_output))
```

- Při spuštění např. v PyCharmu
  - Otevřeme přímo v prohlížeči

```
webbrowser.open("splatnost.html")  
print("Výstup byl uložen do splatnost.html")
```



Univerzita Palackého  
v Olomouci

## Uložení

- Uložíme do HTML souboru jako *splatnost.html*

```
with open("splatnost.html", "w") as f:  
    f.write(html_output)
```



## Jaké dovednosti nám tento kód přinesl?

- Na zpracování dat z Excelu
- Rozdělení podle porovnání s data s aktuálním datem
- Různé možnosti otevření stejného souboru
- Jak můžeme tento kód vylepšit?
  - Vizualizace – pomocí grafů
  - Automatické aktualizování dat