

AUTOMATIZACE OBJEDNÁVÁNÍ OBĚDŮ S PYTHONEM

AUTOMATIZACE

AUTOMATIZACE

"Samočinná technologie provádějící opakovaně proces s minimální lidskou asistencí."

Proces:

Proces:
manuální objednávání obědů

Proces:

manuální objednávání obědů

- výběr restaurace

Proces:

manuální objednávání obědů

- výběr restaurace
- výběr jídla

Proces:

manuální objednávání obědů

- výběr restaurace
- výběr jídla
- odeslání rezervačního e-mailu

Proces:

manuální objednávání obědů

- výběr restaurace
- výběr jídla
- odeslání rezervačního e-mailu
- 5-10 minut zabitých každý den

Proces:

manuální objednávání obědů

- výběr restaurace
- výběr jídla
- odeslání rezervačního e-mailu
- 5-10 minut zabitých každý den

Řešení:

Proces:

manuální objednávání obědů

- výběr restaurace
- výběr jídla
- odeslání rezervačního e-mailu
- 5-10 minut zabíjých každý den

Řešení:

Python!

Plán:

Plán:

1. Získat denní menu restaurací ve strojově čitelné formě

Plán:

1. Získat denní menu restaurací ve strojově čitelné formě
2. Nabídnout volbu jídla ve webové aplikaci

Plán:

1. Získat denní menu restaurací ve strojově čitelné formě
2. Nabídnout volbu jídla ve webové aplikaci
3. Automaticky vygenerovat a odeslat e-mail do restaurace

JACK'S BURGER BAR V PORUBĚ



Zdroj: <http://www.jacksburgerbar.cz/poruba/galerie/>

Jack

**ZDROJ DAT:
ZOMATO API 😊**

Jack

**ZDROJ DAT:
ZOMATO API 😊**

- databáze restaurací a denních nabídek jídel

Jack

**ZDROJ DAT:
ZOMATO API 😊**

- databáze restaurací a denních nabídek jídel
- zdarma 1000 API volání denně

Jack

**ZDROJ DAT:
ZOMATO API 😊**

- databáze restaurací a denních nabídek jídel
- zdarma 1000 API volání denně

POUŽITÉ KNIHOVNY:

Jack

**ZDROJ DAT:
ZOMATO API 😊**

- databáze restaurací a denních nabídek jídel
- zdarma 1000 API volání denně

POUŽITÉ KNIHOVNY:

- requests (komunikace s REST API)

Jack

ZDROJ DAT: ZOMATO API 😊

- databáze restaurací a denních nabídek jídel
- zdarma 1000 API volání denně

POUŽITÉ KNIHOVNY:

- requests (komunikace s REST API)
- re (zpracování textu pomocí regulárních výrazů)

Jack

Jak to vypadá v Pythonu?



Jack

Jak to vypadá v Pythonu?

```
import requests
```


Jack

Jak to vypadá v Pythonu?

```
import requests  
url = "https://developers.zomato.com/api/v2.1/dailymenu"
```

Jack

Jak to vypadá v Pythonu?

```
import requests  
url = "https://developers.zomato.com/api/v2.1/dailymenu"  
data = requests.get(url,
```

Jack

Jak to vypadá v Pythonu?

```
import requests
url = "https://developers.zomato.com/api/v2.1/dailymenu"
data = requests.get(url,
                    params={"res_id": restaurant},
```


Jack

Jak to vypadá v Pythonu?

```
import requests
url = "https://developers.zomato.com/api/v2.1/dailymenu"
data = requests.get(url,
                    params={"res_id": restaurant},
                    headers={"user_key": ZOMATO_API_KEY}
).json()
```

Jack

Jak to vypadá v Pythonu?

```
import requests
url = "https://developers.zomato.com/api/v2.1/dailymenu"
data = requests.get(url,
                    params={"res_id": restaurant},
                    headers={"user_key": ZOMATO_API_KEY}
).json()
```



```
[{'name': 'Čočka na kyselo s pečenou cibulkou, 2 ks vejce, chléb', 'price': '99 Kč'},
{'name': 'Kuřecí plátky v domácím bramboráku, salát Coleslaw', 'price': '99 Kč'},
{'name': 'Losos s bazalkovou omáčkou, penne', 'price': '119 Kč'}]
```

MENZA Č. 5 (VŠB-TUO)



Zdroj: <https://menza.vsb.cz/>

Menza

**ZDROJ DAT:
HTML TABULKA NA WEBU MENZY 😞**

Menza

ZDROJ DAT: HTML TABULKA NA WEBU MENZY ☹️

```
<tr class="Jidlo">
  <td id="Id">5</td>
  <td id="Fotka"></td>
  <td id="Nazev">1/2 pečené kuře, chléb, okurek, zeleninová obloha</td>
  <td id="PocetKusu">8</td>
  <td id="Cena">81,00</td>
</tr>
```

Menza

ZDROJ DAT: HTML TABULKA NA WEBU MENZY ☹️

```
<tr class="Jidlo">
  <td id="Id">5</td>
  <td id="Fotka"></td>
  <td id="Nazev">1/2 pečené kuře, chléb, okurek, zeleninová obloha</td>
  <td id="PocetKusu">8</td>
  <td id="Cena">81,00</td>
</tr>
```

POUŽITÉ KNIHOVNY:

Menza

ZDROJ DAT: HTML TABULKA NA WEBU MENZY ☹️

```
<tr class="Jidlo">
  <td id="Id">5</td>
  <td id="Fotka"></td>
  <td id="Nazev">1/2 pečené kuře, chléb, okurek, zeleninová obloha</td>
  <td id="PocetKusu">8</td>
  <td id="Cena">81,00</td>
</tr>
```

POUŽITÉ KNIHOVNY:

- mechanicalsoup (automatizace browseru)

Menza

ZDROJ DAT: HTML TABULKA NA WEBU MENZY ☹️

```
<tr class="Jidlo">
  <td id="Id">5</td>
  <td id="Fotka"></td>
  <td id="Nazev">1/2 pečené kuře, chléb, okurek, zeleninová obloha</td>
  <td id="PocetKusu">8</td>
  <td id="Cena">81,00</td>
</tr>
```

POUŽITÉ KNIHOVNY:

- mechanicalsoup (automatizace browseru)
- re (zpracování textu pomocí regulárních výrazů)

Menza

Jak to vypadá v Pythonu?



Menza

Jak to vypadá v Pythonu?

```
import mechanicalsoup
```

Menza

Jak to vypadá v Pythonu?

```
import mechanicalsoup  
browser = mechanicalsoup.StatefulBrowser()
```

Menza

Jak to vypadá v Pythonu?

```
import mechanicalsoup
browser = mechanicalsoup.StatefulBrowser()
browser.open("{} / {}".format(MENZA_URL, "ZalozkaObjednavani.aspx"))
```


Menza

Jak to vypadá v Pythonu?

```
import mechanicalsoup
browser = mechanicalsoup.StatefulBrowser()
browser.open("{} / {}".format(MENZA_URL, "ZalozkaObjednavani.aspx"))
page = browser.get_current_page()
```

Menza

Jak to vypadá v Pythonu?

```
import mechanicalsoup
browser = mechanicalsoup.StatefulBrowser()
browser.open("{} / {}".format(MENZA_URL, "ZalozkaObjednavani.aspx"))
page = browser.get_current_page()
table = page.find_all("table")
```

Menza

Jak to vypadá v Pythonu?

```
import mechanicalsoup
browser = mechanicalsoup.StatefulBrowser()
browser.open("{} / {}".format(MENZA_URL, "ZalozkaObjednavani.aspx"))
page = browser.get_current_page()
table = page.find_all("table")
rows = table.find_all("tr")
```

Menza

Jak to vypadá v Pythonu?

```
import mechanicalsoup
browser = mechanicalsoup.StatefulBrowser()
browser.open("{} / {}".format(MENZA_URL, "ZalozkaObjednavani.aspx"))
page = browser.get_current_page()
table = page.find_all("table")
rows = table.find_all("tr")
```

Vyhledání obrázku v řádku tabulky

```
def find_image(row):
    col = row.find(class_="Fotka") # hledání elementů s class="Fotka"
    if col and col.contents:
        link = col.find("img") # hledání obrázku
        if link and "onclick" in link.attrs:
            match = re.search("ImgID=(\d+)", link["onclick"])
            if match:
                return match.group(1)
```

KOVORK



Zdroj: <https://cpi.vsb.cz/cs/sluzby/pronajmy-a-inkubator/kovork/>

Kovork

ZDROJ DAT:

Kovork

**ZDROJ DAT:
RUČNĚ FORMÁTOVANÝ TEXT**

Kovork

**ZDROJ DAT:
RUČNĚ FORMÁTOVANÝ TEXT VE WORDU 2003**

Kovork

**ZDROJ DAT:
RUČNĚ FORMÁTOVANÝ TEXT VE WORDU 2003
ODESÍLANÝ JAKO PŘÍLOHA E-MAILU**

Kovork

ZDROJ DAT:

RUČNĚ FORMÁTOVANÝ TEXT VE WORDU 2003
ODESÍLANÝ JAKO PŘÍLOHA E-MAILU (ノ益益)ノ三

Kovork

ZDROJ DAT: RUČNĚ FORMÁTOVANÝ TEXT VE WORDU 2003 ODESÍLANÝ JAKO PŘÍLOHA E-MAILU (ノ益々)ノ三

Recepce Podnikatelský inkubátor <recepce.pi@vsvb.cz>

komu: pi.obedy@vsvb.cz ▼

Dobrý den,

v příloze Vám zasílám menu kavárny Kovork. MENU SI MŮŽETE OBJEDNAT **POMOCÍ E-MAILU** kavarna.kovork@seznam.cz do 10:45.

Přeji pěkný den.

Kovork

ZDROJ DAT: RUČNĚ FORMÁTOVANÝ TEXT VE WORDU 2003 ODESÍLANÝ JAKO PŘÍLOHA E-MAILU (ノ益々)ノ三

Recepce Podnikatelský inkubátor <recepce.pi@vsvb.cz>

komu: pi.obedy@vsvb.cz ▼

Dobrý den,

v příloze Vám zasílám menu kavárny Kovork. MENU SI MŮŽETE OBJEDNAT **POMOCÍ E-MAILU** kavarna.kovork@seznam.cz do 10:45.

Přeji pěkný den.



Kovork

ZDROJ DAT: RUČNĚ FORMÁTOVANÝ TEXT VE WORDU 2003 ODESÍLANÝ JAKO PŘÍLOHA E-MAILU (ノ益ノ三)

Recepce Podnikatelský inkubátor <recepce.pi@vsb.cz>

komu: pi.obedy@vsb.cz ▾

Dobrý den,

v příloze Vám zasílám menu kavárny Kovork. MENU SI MŮŽETE OBJEDNAT **POMOCÍ E-MAILU** kavarna.kovork@seznam.cz do 10:45.

Přeji pěkný den.



ČTVRTEK 11.10.

OBĚDY SE VYDÁVAJÍ 11:00-13:30

• BRAMBOROVÝ KRÉM SE SMETANOU ⁽⁷⁾

• GULÁŠOVÁ POLÉVKA ⁽¹⁾

K obědovému menu zdarma. Samostatně 29 Kč

1) LASAGNE BOLOGNESE ZAPEČENÉ SÝREM
PODÁVANÉ SE SALÁTKEM

109,- (1,3,7,9)

2) SMAŽENÉ TVARŮŽKY (3KS) DOPLNĚNÉ O VAŘENÝ
BRAMBŮREK A ĎÁBELSKOU OMÁČKU

99,- (1,3,7)

Kovork

POUŽITÉ KNIHOVNY:

Kovork

POUŽITÉ KNIHOVNY:

- google-api-python-client (čtení a odesílání e-mailů)

Kovork

POUŽITÉ KNIHOVNY:

- google-api-python-client (čtení a odesílání e-mailů)
- python-docx (získání textu z DOCu)

Kovork

POUŽITÉ KNIHOVNY:

- google-api-python-client (čtení a odesílání e-mailů)
- python-docx (získání textu z DOCu)
- re (zpracování textu pomocí regulárních výrazů)

Kovork

Načtení wordu z Gmailu



Kovork

Načtení wordu z Gmailu

```
inbox = gmail.list(userId='me',
```

Kovork

Načtení wordu z Gmailu

```
inbox = gmail.list(userId='me',  
    q='from:kavarna.kovork@seznam.cz after:1539248619')
```

Kovork

Načtení wordu z Gmailu

```
inbox = gmail.list(userId='me',  
    q='from:kavarna.kovork@seznam.cz after:1539248619')  
for msg in inbox['messages']:
```

Kovork

Načtení wordu z Gmailu

```
inbox = gmail.list(userId='me',  
    q='from:kavarna.kovork@seznam.cz after:1539248619')  
for msg in inbox['messages']:  
    email = gmail.get(userId='me', id=msg['id'])
```

Kovork

Načtení wordu z Gmailu

```
inbox = gmail.list(userId='me',  
    q='from:kavarna.kovork@seznam.cz after:1539248619')  
for msg in inbox['messages']:  
    email = gmail.get(userId='me', id=msg['id'])  
    subject = email['headers']['subject']
```

Kovork

Načtení wordu z Gmailu

```
inbox = gmail.list(userId='me',  
    q='from:kavarna.kovork@seznam.cz after:1539248619')  
for msg in inbox['messages']:  
    email = gmail.get(userId='me', id=msg['id'])  
    subject = email['headers']['subject']  
    if 'menu' in subject.lower():
```


Kovork

Načtení wordu z Gmailu

```
inbox = gmail.list(userId='me',  
    q='from:kavarna.kovork@seznam.cz after:1539248619')  
for msg in inbox['messages']:  
    email = gmail.get(userId='me', id=msg['id'])  
    subject = email['headers']['subject']  
    if 'menu' in subject.lower():  
        word = email['attachments'][0]
```

Kovork

Získání textu z DOCu



Kovork

Získání textu z DOCu

```
from docx import Document
```

Kovork

Získání textu z DOCu

```
from docx import Document  
document = Document(word)
```

Kovork

Získání textu z DOCu

```
from docx import Document  
document = Document(word)  
menu = document.paragraphs[0].text
```

Kovork

Získání textu z DOCu

```
from docx import Document  
document = Document(word)  
menu = document.paragraphs[0].text
```

python-docx nepodporuje Word 2003 ☹

Kovork

Získání textu z DOCu

```
from docx import Document  
document = Document(word)  
menu = document.paragraphs[0].text
```

python-docx nepodporuje Word 2003 ☹

Řešení:

Kovork

Získání textu z DOCu

```
from docx import Document
document = Document(word)
menu = document.paragraphs[0].text
```

python-docx nepodporuje Word 2003 ☹

Řešení:

- www.convertapi.com

Kovork

Získání textu z DOCu

```
from docx import Document
document = Document(word)
menu = document.paragraphs[0].text
```

python-docx nepodporuje Word 2003 ☹

Řešení:

- www.convertapi.com
- requests (konverze DOC na PDF)

Kovork

Získání textu z DOCu

```
from docx import Document
document = Document(word)
menu = document.paragraphs[0].text
```

python-docx nepodporuje Word 2003 ☹

Řešení:

- www.convertapi.com
- requests (konverze DOC na PDF)
- pdfminer (konverze PDF na text)

Kovork

Konverze DOC na PDF



Extrakce textu z PDF



Kovork

Konverze DOC na PDF

```
data = requests.post('https://v2.convertapi.com/convert/doc/to/pdf',
```

Extrakce textu z PDF

Kovork

Konverze DOC na PDF

```
data = requests.post('https://v2.convertapi.com/convert/doc/to/pdf',  
    params={'SECRET': CONVERT_API_SECRET},
```

Extrakce textu z PDF

Kovork

Konverze DOC na PDF

```
data = requests.post('https://v2.convertapi.com/convert/doc/to/pdf',  
    params={'SECRET': CONVERT_API_SECRET},  
    json=word)
```

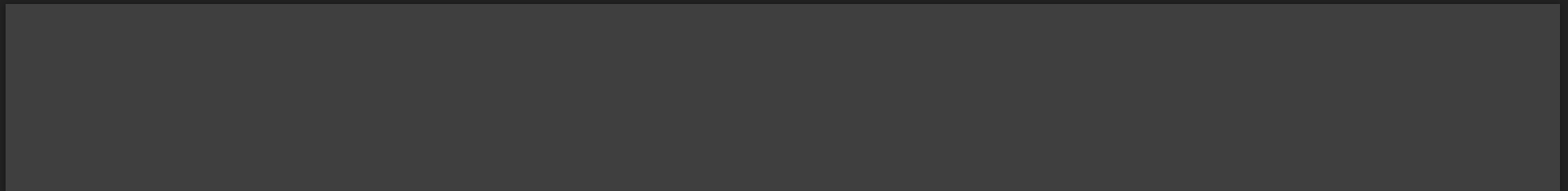
Extrakce textu z PDF

Kovork

Konverze DOC na PDF

```
data = requests.post('https://v2.convertapi.com/convert/doc/to/pdf',  
                    params={'SECRET': CONVERT_API_SECRET},  
                    json=word)  
pdf = base64.b64decode(data)
```

Extrakce textu z PDF



Kovork

Konverze DOC na PDF

```
data = requests.post('https://v2.convertapi.com/convert/doc/to/pdf',  
                    params={'SECRET': CONVERT_API_SECRET},  
                    json=word)  
pdf = base64.b64decode(data)
```

Extrakce textu z PDF

```
from pdfminer import TextConverter, PDFPageInterpreter, PDFPage
```


Kovork

Konverze DOC na PDF

```
data = requests.post('https://v2.convertapi.com/convert/doc/to/pdf',  
                    params={'SECRET': CONVERT_API_SECRET},  
                    json=word)  
pdf = base64.b64decode(data)
```

Extrakce textu z PDF

```
from pdfminer import TextConverter, PDFPageInterpreter, PDFPage  
interpreter = PDFPageInterpreter(TextConverter('utf-8'))
```

Kovork

Konverze DOC na PDF

```
data = requests.post('https://v2.convertapi.com/convert/doc/to/pdf',  
                      params={'SECRET': CONVERT_API_SECRET},  
                      json=word)  
pdf = base64.b64decode(data)
```

Extrakce textu z PDF

```
from pdfminer import TextConverter, PDFPageInterpreter, PDFPage  
interpreter = PDFPageInterpreter(TextConverter('utf-8'))  
page = PDFPage.get_pages(pdf)[0]
```

Kovork

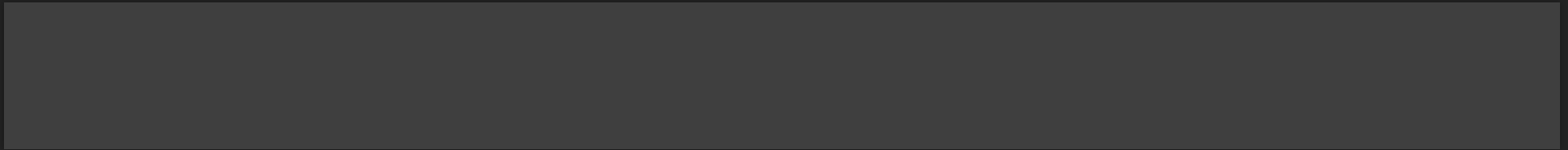
Konverze DOC na PDF

```
data = requests.post('https://v2.convertapi.com/convert/doc/to/pdf',  
                      params={'SECRET': CONVERT_API_SECRET},  
                      json=word)  
pdf = base64.b64decode(data)
```

Extrakce textu z PDF

```
from pdfminer import TextConverter, PDFPageInterpreter, PDFPage  
interpreter = PDFPageInterpreter(TextConverter('utf-8'))  
page = PDFPage.get_pages(pdf)[0]  
text = interpreter.process_page(page).getvalue()
```

Extrakce dat z formátovaného textu



Extrakce dat z formátovaného textu

```
import re
```

Extrakce dat z formátovaného textu

```
import re  
regex = re.compile(
```

Extrakce dat z formátovaného textu

```
import re
regex = re.compile("^(\d+)\s*(.*?)\s*(\d+), -",
```

Extrakce dat z formátovaného textu

```
import re
regex = re.compile("^(\\d)\\)\\s*(.*?)\\s*(\\d+),-",
    re.DOTALL | re.MULTILINE)
```


Extrakce dat z formátovaného textu

```
import re
regex = re.compile("^(\\d)\\)\s*(.*?)\s*(\\d+), -",
    re.DOTALL | re.MULTILINE)
```

```
"1)HOVĚZÍ HAMBURGER (150g) SE ZELENINOU  
A BBQ OMÁČKOU PODÁVANÝ S HRANOLKAMI  
129, - (1,3,7)"
```

Extrakce dat z formátovaného textu

```
import re
regex = re.compile("^(\\d)\\)\s*(.*?)\s*(\\d+), -",
    re.DOTALL | re.MULTILINE)
```

```
"1)HOVĚZÍ HAMBURGER (150g) SE ZELENINOU  
A BBQ OMÁČKOU PODÁVANÝ S HRANOLKAMI  
129, - (1,3,7)"
```



```
{
  "id": 1,
  "food": "HOVĚZÍ HAMBURGER (150g) SE ZELENINOU A BBQ  
OMÁČKOU PODÁVANÝ S HRANOLKAMI",
  "price": 129
}
```


1) Vepřo knedlo zelo 115, -

[^](\d)\)\s*(.*?)\s*(\d+), -

[^] začni testovat výraz na začátku řádku

1) Vepřo knedlo zelo 115, -

`^(\d+)\s*(.*?)\s*(\d+), -`

<code>^</code>	začni testovat výraz na začátku řádku
<code>(\d)</code>	zaznamenej číslici (ID jídla)

1) Vepřo knedlo zelo 115,-

`^(\d)\)\s*(.*?)\s*(\d+), -`

<code>^</code>	začni testovat výraz na začátku řádku
<code>(\d)</code>	zaznamenej číslíci (ID jídla)
<code>\)</code>	následovanou znakem konce závorky <code>)</code>

1)_Vepřo knedlo zelo 115,-

`^(\d)\)\s*(.*?)\s*(\d+), -`

<code>^</code>	začni testovat výraz na začátku řádku
<code>(\d)</code>	zaznamenej číslíci (ID jídla)
<code>\)</code>	následovanou znakem konce závorky <code>)</code>
<code>\s*</code>	ignoruj mezery

1) Vepřo knedlo zelo 115, -

`^(\d)\)\s*(.*?)\s*(\d+), -`

<code>^</code>	začni testovat výraz na začátku řádku
<code>(\d)</code>	zaznamenej číslíci (ID jídla)
<code>\)</code>	následovanou znakem konce závorky <code>)</code>
<code>\s*</code>	ignoruj mezery
<code>(.*?)</code>	zaznamenej cokoliv, co najdeš (název jídla)

1) Vepřo knedlo zelo_115, -

`^(\d)\)\s*(.*?)\s*(\d+), -`

<code>^</code>	začni testovat výraz na začátku řádku
<code>(\d)</code>	zaznamenej číslíci (ID jídla)
<code>\)</code>	následovanou znakem konce závorky <code>)</code>
<code>\s*</code>	ignoruj mezery
<code>(.*?)</code>	zaznamenej cokoliv, co najdeš (název jídla)
<code>\s*</code>	ignoruj mezery

1) Vepřo knedlo zelo 115,-

`^(\d)\)\s*(.*?)\s*(\d+),-`

<code>^</code>	začni testovat výraz na začátku řádku
<code>(\d)</code>	zaznamenej číslici (ID jídla)
<code>\)</code>	následovanou znakem konce závorky <code>)</code>
<code>\s*</code>	ignoruj mezery
<code>(.*?)</code>	zaznamenej cokoliv, co najdeš (název jídla)
<code>\s*</code>	ignoruj mezery
<code>(\d+)</code>	zaznamenej sekvenci číslic (cena)

1) Vepřo knedlo zelo 115,-

`^(\d)\)\s*(.*?)\s*(\d+),-`

<code>^</code>	začni testovat výraz na začátku řádku
<code>(\d)</code>	zaznamenej číslíci (ID jídla)
<code>\)</code>	následovanou znakem konce závorky <code>)</code>
<code>\s*</code>	ignoruj mezery
<code>(.*?)</code>	zaznamenej cokoliv, co najdeš (název jídla)
<code>\s*</code>	ignoruj mezery
<code>(\d+)</code>	zaznamenej sekvenci číslic (cena)
<code>, -</code>	následovanou znaky <code>,</code> <code>-</code>

Web pro objednávání

Web pro objednávání

Knihovny:

Web pro objednávání

Knihovny:

- flask (web, REST API)

Web pro objednávání

Knihovny:

- flask (web, REST API)
- jinja2 (šablony pro web)

Web pro objednávání

Knihovny:

- flask (web, REST API)
- jinja2 (šablony pro web)
- flask_sqlalchemy (databáze)

Hello world stránka ve Flasku



Hello world stránka ve Flasku

```
def hello():  
    return "Hello World!"
```

Hello world stránka ve Flasku

```
from flask import Flask
```

```
def hello():  
    return "Hello World!"
```

Hello world stránka ve Flasku

```
from flask import Flask
app = Flask("Obedy")

def hello():
    return "Hello World!"
```

Hello world stránka ve Flasku

```
from flask import Flask
app = Flask("Obedy")

@app.route("/")
def hello():
    return "Hello World!"
```

Render jinja šablony ve Flasku

```
@app.route("/")
def hello():
    return render_template("index.html",
                           todos=["Vytvořit prezentaci",
                                  "Naučit se Python"])
```

Render jinja šablony ve Flasku

```
@app.route("/")
def hello():
    return render_template("index.html",
                           todos=["Vytvořit prezentaci",
                                  "Naučit se Python"])
```

index.html

```
<ul>
  {% for todo in todos %}
    <li>{{ todo }}</li>
  {% endfor %}
</ul>
```

Šablona e-mailu odesílaného do Kovorku

Dobrý den,

```
{{ "chtěl" if user.sex == "m" else "chtěla" }}
```

bych si objednat tyto obědy:

```
{% for lunch in lunches %}  
  Oběd {{ lunch['name'] }} - {{ lunch['count'] }}x  
{% endfor %}
```

S pozdravem,
{{ user.name }}

Uložení menu do databáze (SQLAlchemy)

```
class Menu(db.Model):
```

Uložení menu do databáze (SQLAlchemy)

```
class Menu(db.Model):  
    id = db.Column(db.Integer, primary_key=True)
```

Uložení menu do databáze (SQLAlchemy)

```
class Menu(db.Model):  
    id = db.Column(db.Integer, primary_key=True)  
    restaurant = db.Column(db.String(80))
```

Uložení menu do databáze (SQLAlchemy)

```
class Menu(db.Model):  
    id = db.Column(db.Integer, primary_key=True)  
    restaurant = db.Column(db.String(80))  
    items = db.Column(db.Text)
```

Uložení menu do databáze (SQLAlchemy)

```
class Menu(db.Model):  
    id = db.Column(db.Integer, primary_key=True)  
    restaurant = db.Column(db.String(80))  
    items = db.Column(db.Text)  
    date = db.Column(db.DATE)
```

Uložení menu do databáze (SQLAlchemy)

```
class Menu(db.Model):  
    id = db.Column(db.Integer, primary_key=True)  
    restaurant = db.Column(db.String(80))  
    items = db.Column(db.Text)  
    date = db.Column(db.DATE)  
  
menu = Menu(restaurant="kovork", items="...", date=datetime.today())
```

Uložení menu do databáze (SQLAlchemy)

```
class Menu(db.Model):  
    id = db.Column(db.Integer, primary_key=True)  
    restaurant = db.Column(db.String(80))  
    items = db.Column(db.Text)  
    date = db.Column(db.DATE)  
  
menu = Menu(restaurant="kovork", items="...", date=datetime.today())  
db.session.add(menu)
```

Uložení menu do databáze (SQLAlchemy)

```
class Menu(db.Model):  
    id = db.Column(db.Integer, primary_key=True)  
    restaurant = db.Column(db.String(80))  
    items = db.Column(db.Text)  
    date = db.Column(db.DATE)  
  
menu = Menu(restaurant="kovork", items="...", date=datetime.today())  
db.session.add(menu)  
db.session.commit()
```


Výsledná aplikace

Jméno	Objednávka
Jakub Beránek	kovork - 1) Lasagne bolognese zapečené sýrem podávané se salátkem
Stanislav Štich	-
Michal Štich	kovork - 1) Lasagne bolognese zapečené sýrem podávané se salátkem
Michal Štich	kovork - 1) Lasagne bolognese zapečené sýrem podávané se salátkem
Michal Štich	-
Michal Štich	kovork - 1) Lasagne bolognese zapečené sýrem podávané se salátkem
Michal Štich	-
<div>Obědvám jinde</div> <div>Zrušit výběr</div>	

Kovork

bramborový krém se smetanou		
1.	Lasagne bolognese zapečené sýrem podávané se salátkem	109 Kč
2.	Smažené tvarůžky (3ks) doplněné o vařený brambůrek a ďábelskou omáčku	99 Kč
3.	Srbské rizoto sypané sýrem	79 Kč
4.	Kuřecí kapsa (120g) plněná špenátem podávaná se smaženými bramborovými dolárky	99 Kč

E-mail do kovorku byl odeslán v 10:30:45 uživatelem Jakub Beránek.

Jack's burger bar

0.	Burger mix Beef Chorizo, 120 g domácí hranolky, salát Coleslaw	169 Kč
150 g hovězí mleté maso míchané s chorizo, aromatický sýr Comté, salát Frisée, polníček, bylinkovo – česnekové ailo		
Smetanová polévka s bramborem		
1.	130 g Plněný paprikový lusk, rajská omáčka, houskový knedlík	99 Kč
2.	150 g Kuřecí jatýrka na cibulce a špeku, rýže, okurek	99 Kč
3.	150 g Vepřový kotlet se sázeným vejcem, hranolky, salátek	119 Kč
7.	LC 3. 150 g Vepřový kotlet se sázeným vejcem, zeleninové pyré, salátek	149 Kč
Celotýdenní menu :		
4.	300 g Jablka v županu s tvarohem a ovocnými dipem	92 Kč
5.	400 g Těstovinový salát se zeleninou a kuřecími kousky 150 g s bylinkový dip	149 Kč
6.	150 g Králík na česneku a špeku, červené zelí, domácí bramborový knedlík	159 Kč

ZBÝVAJÍCÍ AUTOMATIZACE

ZBÝVAJÍCÍ AUTOMATIZACE

- cron - každý den v 10:30 se automaticky odešle e-mail

ZBÝVAJÍCÍ AUTOMATIZACE

- cron - každý den v 10:30 se automaticky odešle e-mail
- Slack - objednávání přes Slack příkazy

ZBÝVAJÍCÍ AUTOMATIZACE

- cron - každý den v 10:30 se automaticky odešle e-mail
- Slack - objednávání přes Slack příkazy



A screenshot of a Slack message input field. The input field contains the text `/obedy order kovork 1`. To the left of the text is a plus sign icon, and to the right are an @ icon and a smiley face icon. Below the input field is a formatting toolbar with icons for bold, italic, strikethrough, code, preformatted, and quote.

<https://github.com/kobzol/oled>

<https://github.com/kobzol/obedy>

DÍKY ZA POZORNOST