



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

Лабораторна робота №1
з дисципліни “Бази даних 2”
за темою “Вивчення базових операцій обробки XML-документів”

Виконала

студентка III курсу

групи КП-82

Джергалова Рената Олександрівна

(прізвище, ім'я, по батькові)

Київ 2021

Завдання

1. Виконати збір інформації зі сторінок Web-сайту за варіантом.
2. Виконати аналіз сторінок Web-сайту для подальшої обробки текстової та графічної інформації, розміщеної на ньому.
3. Реалізувати функціональні можливості згідно вимог.

Базова сторінка (завдання 1): www.football.ua

Зміст завдання 1: Кількість графічних фрагментів по кожному документу

Адреса інтернет-магазину (завдання 2): www.moyo.ua

Виконання

Налаштування scrapy

<code>scrapy startproject labone</code>

scrapy пайплайн для запису зібраних даних у файли

pipelines.py

<pre>from lxml import etree class LabonePipeline(object): def __init__(self): self.root = None def open_spider(self, spider): if (spider.name == "football"): element = "data" else: element = "internet_shop" self.root = etree.Element(element) def close_spider(self, spider): if (spider.name == "football"): task = 1 else: task = 2 with open(f'task{task}.xml', 'wb') as f: f.write(etree.tostring(self.root, encoding="UTF-8", pretty_print=True, xml_declaration=True)) def process_item(self, item, spider): if spider.name == "football": page = etree.Element("page", url=item["url"]) for payload in item["payload"]:</pre>
--

```

        fragment = etree.Element("fragment", type=payload["type"])
        fragment.text = payload["data"]
        page.append(fragment)
        self.root.append(page)
    else:
        product = etree.Element("product")
        desc = etree.Element("description")
        desc.text = item["description"]
        pr = etree.Element("price")
        pr.text = item["price"]
        img = etree.Element("image")
        img.text = item["img"]
        product.append(desc)
        product.append(pr)
        product.append(img)
        self.root.append(product)
    return item

```

scrapy.spiders для збирання даних зі сторінок

spiders/football.py

```

from scrapy.http.response import Response
import scrapy
class FootballSpider(scrapy.Spider):
    name = 'football'
    allowed_domains = ['football.ua']
    start_urls = ['https://football.ua/galleries/']

    def parse(self, response: Response):
        pages_num = 20

        all_images = response.xpath("//img/@src[starts-with(., 'http')]")
        all_text = response.xpath(
            "//*[not(self::script)][not(self::style)][string-length(normalize-
space(text())) > 30]/text()")
        yield {
            'url': response.url,
            'payload': [{ 'type': 'text', 'data': text.get().strip()} for text in
all_text] +
                        [{ 'type': 'image', 'data': image.get()} for image in
all_images]
        }
        if response.url == self.start_urls[0]:
            all_links = response.xpath(
                "//a/@href[starts-with(., 'https://football.ua/')][substring(.,
string-length() - 4) = '.html']")
            selected_links = [link.get() for link in all_links][:pages_num]
            for link in selected_links:
                yield scrapy.Request(link, self.parse)

```

spiders/moyo.py

```

from scrapy.http.response import Response
import scrapy
class MoyoSpider(scrapy.Spider):
    name = 'moyo'
    allowed_domains = ['www.moyo.ua']
    start_urls = [
        'https://www.moyo.ua/detskij_mir/igrushki/konstruktory/?filters=3967_355245&brands=lego']

```

```

def parse(self, response: Response):
    pages_num = 20

    products = response.xpath("//section[contains(@class, 'product-
tile_product')]")[:pages_num]
    for product in products:
        yield {
            'description': product.xpath("./@data-name").get(),
            'price': product.xpath("./@data-price").get(),
            'img': product.xpath("./@data-img").get()
        }

```

xsl-файл для створення xhtml-сторінки

task2.xhtml

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns="http://www.w3.org/1999/xhtml">
  <xsl:output method="xml" doctype-system="http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd" doctype-
public="-//W3C//DTD XHTML 1.1//EN" indent="yes" />
  <xsl:template match="/">
    <html xml:lang="en">
      <head>
        <title>Task 2</title>
      </head>
      <body>
        <h1>LEGOs in MOYO.ua:</h1>
        <xsl:apply-templates select="/shop" />
        <xsl:if test="count(/shop/product) = 0">
          <div>There are no products on selected path :(</div>
        </xsl:if>
      </body>
    </html>
  </xsl:template>
  <xsl:template match="/shop">
    <table border="3">
      <thead>
        <tr>
          <td>Item</td>
          <td>Description</td>
          <td>Price (UAH)</td>
        </tr>
      </thead>
      <tbody>
        <xsl:apply-templates select="/shop/product" />
      </tbody>
    </table>
  </xsl:template>
  <xsl:template match="/shop/product">
    <tr>
      <td>
        <xsl:apply-templates select="image" />
      </td>
      <td>
        <xsl:apply-templates select="description" />
      </td>
      <td>
        <xsl:apply-templates select="price" />
      </td>
    </tr>
  </xsl:template>
  <xsl:template match="image">
    <img alt="image of product">
      <xsl:attribute name="src">
        <xsl:value-of select="text()" />
      </xsl:attribute>
    </img>
  </xsl:template>

```

```
<xsl:template match="price">
  <xsl:value-of select="text()" />
</xsl:template>
<xsl:template match="description">
  <xsl:value-of select="text()" />
</xsl:template>
</xsl:stylesheet>
```

Завдання

main.py

```
def task1():
    print_delimiter()
    print("Task 1 - football.ua")
    root = etree.parse("task1.xml")
    pages = root.xpath("//page")

    print("Number of images for scrapped pages:")
    for page in pages:
        url = page.xpath("@url")[0]
        imgs_count = page.xpath("count(fragment[@type='image'])")
        print(f"{url}: {imgs_count} img(s)")

def task2():
    print_delimiter()
    print("Task 2 - moyo.ua")

    transform = etree.XSLT(etree.parse("task2.xsl"))
    result = transform(etree.parse("task2.xml"))
    result.write("task2.xhtml", pretty_print=True, encoding="UTF-8")

    print("- Opening XHTML page")
    webbrowser.open('file://' + os.path.realpath("task2.xhtml"))

def clean_files():
    try:
        os.remove("task1.xml")
        os.remove("task2.xml")
        os.remove("task2.xhtml")
    except OSError:
        pass

def scrap_data():
    process = CrawlerProcess(get_project_settings())
    process.crawl('football')
    process.crawl('moyo')
    process.start()
```

Приклади роботи програми

```
Lab #1 - Dzherhalova Renata - KP82
=====
- Scrapping data
- Scraping finished
=====
Choose the task
1. Task #1
2. Task #2
0. Quit
:: 1
=====
Task 1 - football.ua
Number of images for scrapped pages:
https://football.ua/galleries/: 12.0 img(s)
https://football.ua/gallery/2780.html: 85.0 img(s)
https://football.ua/fiction.html: 32.0 img(s)
https://football.ua/nationsleague.html: 67.0 img(s)
https://football.ua/uefa.html: 63.0 img(s)
https://football.ua/countriesth.html: 22.0 img(s)
https://football.ua/netherlands.html: 55.0 img(s)
https://football.ua/portugal.html: 37.0 img(s)
https://football.ua/italy.html: 56.0 img(s)
https://football.ua/germany.html: 52.0 img(s)
https://football.ua/france.html: 59.0 img(s)
https://football.ua/spain.html: 59.0 img(s)
https://football.ua/fansector.html: 22.0 img(s)
https://football.ua/bitva-redaktsiy.html: 22.0 img(s)
https://football.ua/ukraine.html: 47.0 img(s)
https://football.ua/life-principles.html: 22.0 img(s)
https://football.ua/footballtest.html: 32.0 img(s)
https://football.ua/bet-ratings.html: 10.0 img(s)
https://football.ua/turkey.html: 36.0 img(s)
https://football.ua/england.html: 56.0 img(s)
https://football.ua/worldcup.html: 67.0 img(s)
=====
```

Рис. 1. Завдання 1

```
=====
Task 2 - moyo.ua
- Opening XHTML page
=====
```

Рис. 2.1. Завдання 2 (консоль)

LEGOs in MOYO.ua:










Item	Description	Price (UAH)
	Конструктор LEGO Harry Potter ТМ Замок Хогвартс (71043)	11462
	Конструктор LEGO Star Wars Имперский звездный разрушитель 75252	19622
	Конструктор LEGO Creator Гараж на углу 10264	5101
	Конструктор LEGO Creator Стадион Олд Траффорд Манчестер Юнайтед	7458
	Конструктор LEGO Ideas Центральная кофейня 21319	2079
	Конструктор LEGO Stranger Things По ту сторону 75810	5731
	Конструктор LEGO Коробка творческого строительства James Bond ТМ Aston Martin DB5 (10262)	4396
	Конструктор LEGO Ideas Пираты из залива Барракуда 21322	5728
	Конструктор LEGO Creator Американские горки 10261	9417

Рис. 2.2. Завдання 2