

## پروژه سیستم عامل – بخش اول

### هدف پروژه

با استفاده از خزنده وب می‌خواهیم قیمت تعدادی گوشی را از سایت های فروشگاه‌های استخراج کنیم سپس به وسیله چند نخ‌سازی سرعت استخراج داده‌ها را در حالت عادی و چند نخ‌مقایسه کنیم.

### کتابخانه‌هایی که در این کد استفاده شده‌اند

Threading

requests

pandas

time

BeautifulSoup

Unidecode

statistics

## توضیحات کلی

ابتدا دو لیست خالی به نام‌های Prices و Names و یک دیکشنری خالی به نام data\_pairs تعریف میکنیم.

```
Names = []  
Prices = []  
data_pairs = {}
```

سپس به وسیله کتابخانه requests یک درخواست به صفحه فروشگاه ارسال میکنیم

```
result = requests.get(  
    "https://emalls.ir/%D9%84%DB%8C%D8%B3%D8%AA-  
%D9%82%DB%8C%D9%85%D8%AA~Category~39"  
)
```

حال یک شرط تعیین میکنم وقتی درخواست ما به صفحه ارسال شد اگر آن صفحه موجود باشد و پاسخ دریافتی ما یا همان کد وضعیت 200 باشد که نشان دهنده معتبر بودن صفحه است وارد بدن شرط شود و اطلاعات را استخراج کند. اگر صفحه معتبر نبود پیامی حاوی عدم دسترسی به صفحه برای ما نمایش داده میشود.

```
if result.status_code == 200:
```

```
    else:  
        print(f"Request was denied by code: {result.status_code}")
```

ابتدا با کتابخانه BeautifulSoup محتوای صفحه ای که به آن درخواست فرستادیم در یک متغیر به نام soup ذخیره میکنیم و به وسیله تجزیه کننده lxml که یک تجزیه کننده است محتوای دریافتی را مرتب میکنم.

```
soup = BeautifulSoup(result.text, "lxml")
```

در مرحله بعد باید تگ مورد نظرم نام گوشی موبایل در آن قرار گرفته است انتخاب کنیم تا بتوانیم به نام گوشی هایی که در آن صفحه هستند دسترسی داشته باشیم این کار را به وسیله متد select و مشخص کردن سلکتورهای CSS انجام میدهیم

```
list_of_names = soup.select("div h2 .maintitle")
```

نام گوشی ها در یک لیست ذخیره شده اند حال باید روی نام ها پیمایش کنیم و کلمات اضافی که کنار این نام ها قرار گرفته اند را حذف کنیم این کار را به وسیله یک حلقه انجام میدهیم و مجدد اسم های تمیز شده را در یک لیست ذخیره میکنیم

```
for item in list_of_names:
    Names.append(
        item.getText().replace(" Mobile Phone", "").replace(" mobile
phone", "")
    )
```

مرحله قبل را مجدد برای دریافت قیمت گوشی ها انجام میدهیم

```
list_of_prices = soup.select(".prd-price span")
for item in list_of_prices:
    Prices.append(int(unicode(item.getText().replace(", ", ""))))
```

نوبت به ذخیره نام و قیمت های میرسید یک دیکشنری تعریف میکنیم و روی دولیستی که قیمت و نام گوشی ها در آن است پیمایش انجام میدهیم و به عنوان کلید، مقدار نام و قیمت را ذخیره میکنیم

```
for i in range(len(Names)):
    data_pairs[Names[i]] = Prices[i]
```

باید از بین قیمت هایی که در دیکشنری ذخیره کردیم، گوشی که بیشترین قیمت و گوشی که کمترین قیمت را دارند پیدا کنیم این کار را به وسیله دو حلقه انجام میدهیم که یکی پایین ترین قیمت و یکی بالاترین قیمت را به ما میدهد

```
max_price = max(data_pairs.values())

for item in data_pairs.items():
    if item[1] == max_price:
        ME_phone = item[0]

min_price = min(data_pairs.values())

for item in data_pairs.items():
    if item[1] == min_price:
        Ch_phone = item[0]
```

در آخر دیکشنری خود را به وسیله کتابخانه pandas تبدیل به دیتافریم میکنیم که شامل جدولی از اطلاعات دیکشنری ماست، دیتافریم ایجاد شده را تبدیل به یک فایل CSV میکنیم با استفاده از متد to.SCV

```
pandas.DataFrame.from_dict(data=data_pairs, orient="index").to_csv(
    "emails.csv", header=False
)
```

حالا قیمت بیشترین و کمترین گوشی و میانگین قیمت گوشی هارا در خروجی چاپ میکنیم

```
print("emails")
print("-----")
print(f"Most expensive phone: {ME_phone} -> {max_price}")
print(f"Cheapest phone: {Ch_phone} -> {min_price}")
print(f"Average Price: {mean(data_pairs.values())}")
print("\n")
```

حال ما سه تابع تعریف کردیم که هر تابع از یک سایت اطلاعات را برای ما واکنشی میکند باید هر تابع را تبدیل به یک نخ کنیم برای اینکار به روش زیر عمل میکنیم

```
thread1 = threading.Thread(target=emails)
thread2 = threading.Thread(target=MoboNews)
thread3 = threading.Thread(target=MobileIr)
```

نخ های ما باید شروع به کار کنند و اطلاعات استخراج شود در اینجا از متد start برای استارت نخ ها استفاده میکنیم

```
thread1.start()
thread2.start()
thread3.start()
```

هرنخ که اجرا میشود باید مطمئن شویم تا انتها انجام میشود و موجب بروز خطا نخواهد شد برای این مورد از متد join برای هر نخ استفاده میکنیم

```
thread1.join()
thread2.join()
thread3.join()
```

در انتها هم زمان انجام کل فرایند ها به وسیله کتابخانه time محاسبه و نمایش داده میشود

```
print(f"Run-time: {ending_time - starting_time}")
```