

Занятие 11

Работа с файлами

time, datetime, calendar

Что напечатает?

```
import functools
functools.reduce(lambda x, y: x * y, [1,2,3,4,5])
functools.reduce(lambda x, y: x * y, [1,2,3,4,5], 0)
functools.reduce(lambda x, y: x * y, [1,2,3,4,5], 100)
```

```
with open('test.txt', 'w') as fo:
```

```
    fo.write('')
```

```
with open('test.txt') as f:
```

```
    print(111, len(f.read()))
```

```
print(222, len(f.read()))
```

Задача 10-1

Есть Excel файл, в первом листе которого находится информация по выработке каждого программиста.

В первой строке ФИО, во второй ячейке результат его работы за один день.

Дана информация за несколько дней по разным людям.

Необходимо во втором листе сформировать суммарные итоги по каждому сотруднику и ИТОГО по списку по всем сотрудникам.

Например:

Иванов 100

Петров 400

Иванов 200

Во втором листе должно быть:

Иванов 300

Петров 400

ИТОГО 700

Задача 10-2

Дан Excel файл. В первом листе фамилии и выработка по дням за один период времени.

Во втором тоже самое за другой период времени.

Фамилии могут быть в другом порядке.

Фамилии могут повторяться в каждом листе несколько раз, а могут быть разными.

Необходимо создать третий лист, который суммирует выработку из первых двух.

Список должен быть отсортирован по фамилиям.

Задача 10-3

Дан эксельный файл со списком людей и результатами их работы (люди не повторяются).

Необходимо сформировать еще один лист со следующей информацией:

- Минимальное значение
- Максимальное значение
- Среднее арифметическое
- Медиана (серединное значение). Для списка с нечетным количеством членов, это значение посередине упорядоченного списка (1, 3, 5, 7, 9) -> 5.
Для четного количества – полусумма средних чисел (1, 3, 5, 7) -> $(3 + 5) / 2 = 4$

with ... as ... - менеджер контекста

```
with open('file.txt', 'r', encoding = 'utf-8') as fi:
```

```
    print(fi.readlines())
```

файл закрывается автоматически

```
with open('test1.txt', 'w') as fo, open('test.txt') as fi: # два файла
```

```
    for line in fi:
```

```
        print(line.strip(), file = fo)
```

CSV (Comma Separated Values)

CSV (comma-separated value) - это формат представления табличных данных (например, это могут быть данные из таблицы или данные из БД).

В этом формате каждая строка файла - это строка таблицы.

Несмотря на название формата, разделителем может быть не только запятая.

И хотя у форматов с другим разделителем может быть и собственное название, например, TSV (tab separated values), тем не менее, под форматом CSV понимают, как правило, любые разделители.

Пример:

```
employee,name,salary  
1,Ivanov,200  
2,Petrov,300  
3,Sidorov,400
```

```
import csv - writer
```

Запись:

```
import csv
```

```
columns = ['first_name', 'second_name', 'rating']
```

```
data = [['Иван', 'Иванов', 123],  
        ['Петр', 'Петров', 234],  
        ['Степан', 'Степанов', 345]]
```

```
with open('test1.csv', 'w', encoding='utf-8', newline='') as file:
```

```
    writer = csv.writer(file)
```

```
    writer.writerow(columns)
```

```
    for row in data:
```

```
        writer.writerow(row)
```



```
import csv - reader()
```

Чтение:

```
import csv
```

```
with open('products.csv', encoding='utf-8') as file:
```

```
    rows = csv.reader(file)                # создаем reader объект
```

```
    for row in rows:
```

```
        print(row)
```

Объект reader дает доступ к построчному итератору, полностью аналогичному работе с файлом или списком.

```
import csv - DictReader()
```

Чтение каждой строки как словаря, с ключами из первой строки:

```
import csv
```

```
with open('products.csv', encoding='utf-8') as file:
```

```
    rows = csv.DictReader(file)    # создается словарь
```

```
    for row in rows:
```

```
        print(row)
```

Задание

Создайте CSV файл, содержащий:

Name,Salary

Иванов,100

Петров,200

Сидоров,300

Напишите программу, которая читает эту информацию, вычисляет и печатает на экране ИТОГО: ???

Модуль openpyxl

- [Модуль openpyxl](#) - это библиотека Python для чтения/записи форматов Office Open XML (файлов Excel 2010) с расширениями xlsx/xlsm/xltx/xltm.

Не входит в стандартную библиотеку, необходимо его установить

- IDLE: `pip install openpyxl`
- PyCharm: File/Settings/Project.../Python Interpreter/ + / набрать openpyxl / Install Package

Попробуйте в консоли `import openpyxl`

Создание книги Excel (workbook)

```
import openpyxl  
from openpyxl import Workbook  
wb = Workbook() # создаем экземпляр класса Workbook  
wb.save('test.xlsx') # сразу его записываем пустой
```

Рабочие листы

```
import openpyxl
wb = openpyxl.load_workbook("test.xlsx")
print(wb.sheetnames)           # список листов
ws = wb.active                 # Кто активный рабочий лист?
print(ws.title)               # Посмотрим
ws.title = "NewPage"          # Изменим его имя
print(ws.title)               # Проверим
ws3 = wb["NewPage"]           # Другой лист
print(ws3)
print(wb.sheetnames)          # список листов
wb.active = ???               # назначим активный лист
wb.remove(ws)                  # удаление рабочего листа
print(wb.sheetnames)
wb.save('test.xlsx')
```

Доступ к ячейкам

Пусть sheet один из листов

Retrieve the value of a certain cell

sheet['A1'].value # Значение, которое хранится в ячейке

Select element 'B2' of your sheet

c = sheet['B2'] # выбрать ячейку

Retrieve the row number of your element

c.row # номер строки

Retrieve the column letter of your element

c.column # номер колонки

Retrieve the coordinates of the cell

c.coordinate # координаты ячейки

Работа с ячейками

Retrieve cell value

sheet.cell(row=1, column=2).value # значение ячейки по номеру строки и колонки

Print out values in column 2 # печатаем колонку номер 2

for i in range(1, 4):

print(i, sheet.cell(row=i, column=2).value)

Print row per row # печатаем строку за строкой

for cellObj in sheet['A1':'C3']:

for cell in cellObj:

print(cell.coordinate, cell.value)

print('--- END ---')

max_row max_col

Retrieve the maximum amount of rows

sheet.max_row # максимальная строка, где есть информация

Retrieve the maximum amount of columns

sheet.max_column # максимальная колонка, где есть информация

Как напечатать все ячейки листа

```
for i in range(ws.max_row):  
    for j in range(ws.max_column):  
        print(i + 1, j + 1, ws.cell(row = i + 1, column = j + 1).value)
```

Как можно напечатать все ячейки всех листов?

```
for sh in wb.sheetnames:  
    ws = wb[sh]  
    print(ws.title, '-----')  
    for i in range(ws.max_row):  
        for j in range(ws.max_column):  
            print(i + 1, j + 1, ws.cell(i + 1, j + 1).value)
```

Выполните эту программу. Что она напечатает?

Основные функции

##	Команда	Что делает
1	<code>import openpyxl</code>	Импортирует модуль openpyxl
2	<code>from openpyxl import Workbook</code>	Загрузка класса Workbook
3	<code>wb = Workbook()</code>	Создаем рабочую книгу
4	<code>wb.save('test.xlsx')</code>	Сохраняем файл
5	<code>wb = openpyxl.load_workbook("test.xlsx")</code>	Загружаем файл
6	<code>ws = wb.active</code>	Определяем активный рабочий лист
7	<code>wb.active = ws</code>	Переопределяем активный рабочий лист
8	<code>ws.title</code>	Имя листа
9	<code>wb.sheetnames</code>	Список листов книги
10	<code>wb.create_sheet("Newsheet")</code>	Создание нового листа
11	<code>wb.remove(ws)</code>	Удаление листа
12	<code>ws['A1'].value</code>	Значение ячейки (cell)
13	<code>c = ws['B2']</code>	Присвоить ячейку (не значение!!!)
14	<code>c.row</code>	Номер строки
15	<code>c.column</code>	Номер колонки
16	<code>c.coordinate</code>	Координаты ячейки ('A1')
17	<code>ws.cell(row = 1, column = 2).value</code>	Значение ячейки по номеру строки и колонки (нумерация с 1)
18	<code>ws.max_row</code>	Максимальная строка с данными
19	<code>ws.max_column</code>	Максимальная колонка с данными

Еще немного

##	Команда	Что делает
20	<code>source_page = wb.active</code> <code>target_page =</code> <code>wb.copy_worksheet(source_page)</code>	Копирование листа
21	<code>ws.append([111, 'Текст', 333])</code>	Добавление списка в строчку после последней
22	<code>ws.append({1:'123', 3:'345'})</code>	Добавление словаря в строчку после последней. Ключи – номера столбцов
23	<code>col_range = ws['C:D']</code>	Срез. Все доступные ячейки в колонках от C до D
24	<code>row_range = ws[5:10]</code>	Срез. Все доступные ячейки в строках от 5 до 10

Добавьте строку и словарь в эксельный файл и напечатайте их

`itertools.permutations(iterable, r=None)` - Перестановки

```
from itertools import permutations
```

```
for k in permutations([1,2,3]): print(k)
```

```
for k in permutations([1,2,3,4], 2): print(k)
```

```
for k in permutations("abcd", 2): print(k)
```

```
for k in permutations("aba", 2): print(k)
```

```
# Дана строка 'abcde'.
```

```
# Сосчитайте и напечатайте количество перестановок для r = 1, 2, 3, 4, 5
```

Time, Datetime, Calendar

Time – удобен для оценки длительности программы, нахождения самых медленных и неэффективных ее частей

Datetime – огромный набор классов и функций для решения разнообразных задач с датами и временем

Calendar – модуль для работы с календарями

Модуль time

`import time` # Удобен для технической работы со временем.

`t0 = time.time()`

`t1 = time.time()`

`print(t1 - t0, t0, t1)` # Выполните эти команды

`import time` # Например, оценить время работы программы:

`x = 1000000`

`for i in range(0, x, 100000):`

`t0 = time.time()`

`su = 0`

`for j in range(i * 10):`

`su += j`

`t1 = time.time()`

`print(i, t1 - t0)`

Выполните эту программу, увеличивая значение x, чтобы визуальное увидеть замедление

Некоторые функции модуля time

##	Функция	Что делает
1	<code>import time</code>	Импортирует модуль
2	<code>time.time()</code>	Текущее время в секундах
3	<code>time.ctime()</code>	Текущее время
4	<code>time.ctime(t)</code>	Дата и время из количества секунд t
5	<code>time.sleep()</code>	Приостанавливает выполнение программы на ... секунд

Напишите программу, которая запоминает момент начала программы, напечатайте дату и время с помощью функции `ctime`, затем «усыпите» программу на 5 секунд, потом запомните момент завершения программы, напечатайте дату и время. И напечатайте разницу между моментами окончания и начала.

Задание

Напишите программу, в которой используется две функции. В одной программа «спит» 2 секунды, в другой – 3 секунды.

Пусть каждая функция возвращает время, которое она «проспала».

Главная программа запускает цикл от 0 до 10, если число четное, то запускает функцию с 2 секундами, если нечетное, то функцию с 3 секундами.

Накапливает сон обеих функций отдельно и печатает две суммы.

Модуль datetime

- Класс **datetime.date**(year, month, day) - стандартная дата. Атрибуты: year, month, day. Неизменяемый объект.
- Класс **datetime.time**(hour=0, minute=0, second=0, microsecond=0) - стандартное время, не зависит от даты. Атрибуты: hour, minute, second, microsecond.
- Класс **datetime.timedelta**(days=0, seconds=0, microseconds=0, milliseconds=0, minutes=0, hours=0, weeks=0) - разница между двумя моментами времени, с точностью до микросекунд.
- Класс **datetime.datetime**(year, month, day, hour=0, minute=0, second=0, microsecond=0) - комбинация даты и времени.
- **datetime.datetime.today()** - объект datetime из текущей даты и времени.
- Сохраните в переменную **b** «сегодня» и напечатайте все его атрибуты, год, месяц, день и т.д.

Преобразование строки в дату – `strptime()`

- # Воспользуемся библиотекой `datetime` методом `strptime`
from `datetime` **import** `datetime`
`print(datetime.strptime('22 04 2020 19:33', '%d %m %Y %H:%M'))`
- 2020-04-22 19:33:00

Преобразование даты и времени в строки – strftime()

```
from datetime import date, time
```

```
my_date = date(2021, 8, 10)
```

```
my_time = time(7, 18, 34)
```

```
print(my_date)                # вывод в ISO формате
```

```
print(my_time)                # вывод в ISO формате
```

```
print(my_date.strftime('%d/%m/%y'))    # форматированный вывод даты
```

```
print(my_date.strftime('%A %d, %B %Y')) # форматированный вывод даты
```

```
print(my_time.strftime('%H.%M.%S'))    # форматированный вывод
```

Выполните эти операторы. Поменяйте разделители на другие.
Замените буквы на y на Y, A на a, B на b

Получение текущей даты и времени

Одним из классов, определенных в модуле `datetime`, является класс `datetime`.

После импортирования класса мы можем применить метод `now()` для создания объекта `datetime`, содержащего текущие локальные дату и время.

```
from datetime import datetime
datetime_object = datetime.now()
t0 = datetime_object
# После небольшой паузы
datetime_object = datetime.now()
t1 = datetime_object
print(t0, t1, t0 == t1, t0 < t1)
```

Что напечатает?

Создание даты и времени из кортежа

```
from datetime import datetime  
a = datetime(2017, 11, 28, 23, 55, 59)  
print("year =", a.year)  
print("month =", a.month)  
print("day =", a.day)  
  
print("hour =", a.hour)  
print("minute =", a.minute)  
print("second =", a.second)
```

Выполните это для сегодня

Получение текущей даты – еще один способ

Метод `today()`, определенный в классе `date`, чтобы получить объект `date`, содержащий текущую локальную дату.

```
from datetime import date  
today = date.today()  
print("Current date =", today)  
2022-04-22
```

Конструирование даты

```
import datetime
```

```
dt = datetime.date(2020, 6, 29)
```

```
print(dt)
```

```
2020-06-29
```

```
# получение значений
```

```
print("Current year:", dt.year)
```

```
print("Current month:", dt.month)
```

```
print("Current day:", dt.day)
```


Форматирование даты

```
from datetime import datetime
```

```
now = datetime.now()
```

```
t = now.strftime("%H:%M:%S")
```

```
print("time:", t)
```

```
time: 15:00:24
```

```
s1 = now.strftime("m/%d/%Y, %H:%M:%S")
```

```
print("s1:", s1)
```

```
S1: 04/22/2022, 15:00:24
```

```
s2 = now.strftime("d/%m/%Y, %H:%M:%S")
```

```
# dd/mm/YY H:M:S format
```

```
print("s2:", s2)
```

Основные коды для определения формата:

%Y — год [0001, ..., 2018, 2019, ..., 9999]

%m — месяц [01, 02, ..., 11, 12]

%d — день [01, 02, ..., 30, 31]

%H — час [00, 01, ..., 22, 23]

%M — минута [00, 01, ..., 58, 59]

%S — секунда [00, 01, ..., 58, 59]

Локализация

```
from datetime import date  
import locale
```

```
locale.setlocale(locale.LC_ALL, "ru") # иногда используется 'ru_RU.UTF-8'
```

```
my_date = date(2021, 8, 10)  
print(my_date.strftime("%A %d, %B %Y"))
```

Задание

```
from datetime import date  
import locale
```

```
locale.setlocale(locale.LC_ALL, 'ru') #, 'en_EN.UTF-8')  
birthday = date(1992, 10, 6)
```

```
print('Название месяца:', birthday.strftime('%B'))  
print('Название дня недели:', birthday.strftime('%A'))  
print('Год:', birthday.strftime('%Y'))  
print('Месяц:', birthday.strftime('%m'))  
print('День:', birthday.strftime('%d'))
```

Поставьте свой день рождения и выполните эти операторы

import calendar

Функция **weekday(year, month, day)** возвращает день недели в виде целого числа (по умолчанию 0 – понедельник, 6 – воскресенье) для заданной даты.

Функция **monthrange(year, month)** возвращает день недели первого дня месяца и количество дней в месяце в виде кортежа для указанного года year и месяца month

Функция **monthcalendar(year, month)** возвращает матрицу, представляющую календарь на месяц. Каждая строка матрицы представляет неделю.

Функция **month(year, month, w=0, l=0)** возвращает календарь на месяц в многострочной строке. Аргументами функции являются: year (год), month (месяц), w (ширина столбца даты) и l (количество строк, отводимые на неделю).

Функция **calendar(year, w=2, l=1, c=6, m=3)** возвращает календарь на весь год в виде многострочной строки. Аргументами функции являются: year (год), w (ширина столбца даты) и l (количество строк, отводимые на неделю), c (количество пробелов между столбцом месяца), m (количество столбцов).

Функция **isleap(year)** определяет, является ли год високосным.

Задание

Напишите программу, которая принимает в качестве аргумента номер года.

В качестве результата формирует словарь, в котором считается количество понедельников, вторников и т.д. в этом году.

Дни недели года

```
import calendar
year = int(input("Введите номер года:"))
dw = {0:0, 1:0, 2:0, 3:0, 4:0, 5:0, 6:0}
for month in range(1, 13):
    for day in range(1, calendar.monthrange(year, month)[1] + 1):
        dd = calendar.weekday(year, month, day)
        dw[dd] += 1 # dw[dd] = dw.get(dd, 0) + 1
print(dw)
```

Задача 11-1

Каждый третий четверг каждого месяца билеты в Эрмитаж бесплатны.

Напечатайте список дат в 2023 году, когда вход бесплатен.

Задача 11-2

Дан файл с расширением .csv, содержащий в каждой строке следующую информацию: номер, фамилия, имя, компания, зарплата.

Создайте Эксельный файл, в который перенесите эту информацию, предварительно отсортировав этот список по компании, по фамилии и имени.

В конце списка добавьте строку: ИТОГО и суммарное значение всех зарплат.

Задача 11-3

Напишите функцию, которая переводит арабские числа в римские.

Например: 2023 ->MMXXIII