Методические указания

Урок 7.1 Коллекции и их методы

Задачи урока:

- Изучить механизм вывода списков и строковые методы split() и join()

0. Подготовка к уроку

До начала урока преподавателю необходимо:

1) Просмотреть, как ученики справились с домашним заданием
2) Прочитать методичку

1. Повторение пройденного материала

Учитель: Давайте вспомним методы списков и как с ними работать. Задает вопросы по прошлому занятию ученикам

2. Вывод содержимого списков

Учитель: При использовании встроенной функции **print()** со списками выводится список в квадратных скобках, и все его элементы разделены запятой. Для просмотра содержимого списков такой вывод достаточно удобен, однако бывают ситуации и задачи где вывод должен соответствовать определенным требованиям. Например, я могу попросить вас вывести каждый элемент списка на отдельной строке, или разделить элементы символом пробела, без запятых, и вывести список без квадратных скобок по бокам.

Другими словами, нужно уметь адаптировать вывод содержимого списка под задачу.. Это несложно. Цикл for нам в помощь, как говорится.

```
a = [1, 2, 3, 'a', True]
for i in a:
    print(i)
```

```
a = [1, 2, 3, 'a', True]
for i in range(len(a)):
```

```
print(a[i])
```

```
a = [1, 2, 3, ‘a’, True]
print(*a) # распаковка списка(выводит без квадратных скобок)
```

1. Отличается ли вывод списка от вывода содержимого строки?
   Ответ: нет, можно использовать два вида цикла, для вывода содержимого списка. Первый удобно использовать, когда нужны индексы элементов, второй, когда нужны сами элементы. Демонстрация на слайде.

2. А было ли что-нибудь новое, что мы до этих пор не использовали со строками?
   Ответ: да, выводить содержимое списка можно еще и с помощью так называемой распаковки списка.

   Замечание. У учеников может возникнуть вопрос, как работает распаковка списков. Ответить пока можно так: это магия Python и на текущем уровне владения языком нам не обязательно понимать подробности ее реализации.

2. Строковые методы split и join

Учитель: При изучении **строковых методов** мы намеренно обошли стороной два очень важных метода split() и join(). Обошли мы их по очень простой причине, они работают со строками и списками в паре. Сейчас мы знаем и о строках, и о списках, поэтому можем изучить их.

При создании (конструировании) списков мы считывали построчно элементы списка, а затем добавляли их в список. Но что если начальные данные расположены в одной строке и разделены символом пробела? Такой способ представления данных достаточно удобен и часто встречается в реальной жизни. Как нам сконструировать список из такой строки? Другими словами, как нам разбить такую строку на элементы списка?

Метод split

```
languages = 'Python C# Java'.split()
print(languages)  # ['Python', 'C#', 'Java']

numbers = '1 2 3 4 5'.split()
print(numbers)    # ['1', '2', '3', '4', '5']
```

```
words = 'To    be or not    to   be that is the     question'.split()
print(words)      # ['To', 'be', 'or', 'not', 'to', 'be', 'that', 'is',
'the', 'question']

ip = '192.168.1.1'.split('.')
print(ip)         # ['192', '168', '1', '1']

terms = '1 + 2 + 3 + 4 = 10'.split(' + ')
print(terms)      # ['1', '2', '3', '4 = 10']
```

Функция split сканирует всю строку и разделяет ее в случае нахождения разделителя. В строке должен быть как минимум один разделитель. Им может выступать в том числе и символ пробела. Пробел — разделитель по умолчанию.

Метод join

```
languages = ' '.join(['Python', 'C#', 'Java'])
print(languages)  # 'Python C# Java'

numbers = ' '.join(['1', '2', '3', '4', '5'])
print(numbers)    # '1 2 3 4 5'

words = '   '.join(['To', 'be', 'or', 'not', 'to', 'be', 'that', 'is',
'the', 'question'])
print(words)      # 'To   be   or   not   to   be   that   is   the
question'

ip = '.'.join(['192', '168', '1', '1'])
print(ip)         # '192.168.1.1'

terms = ' + '.join(['1', '2', '3', '4 = 10'])
print(terms)      # '1 + 2 + 3 + 4 = 10'
```

Метод join в Python отвечает за объединение списка строк с помощью определенного указателя. Часто это используется при конвертации списка в строку. Например, так можно конвертировать список букв алфавита в разделенную запятыми строку для сохранения. Метод принимает итерируемый объект в качестве аргумента, а поскольку список отвечает этим условиям, то его вполне можно использовать. Также список должен состоять из строк. Если попробовать использовать функцию для списка с другим содержимым, то результатом будет такое сообщение: TypeError: sequence item 0: expected str instance, int found.

3. Решение задач

Задача 1

Написать программу, запрашивающую у пользователя на ввод имя и класс(данные вводятся в одну строку через пробел) и вывести класс пользователя(Пример: 8А)

Решение

```
user_list = input().split()
print(user_list[1])
```

Задача 2

Есть список слов, необходимо составить из него строку через цикл, а также с помощью метода join

Решение

```
a = ['hello', 'world']
b = ' '.join(a)
c = ''
for letter in a:
    c += letter
    c += ' '
print(b)
print(c)
```

Дополнительно

Если на уроке остается время, то ученикам можно предложить начать прорешивать домашнее задание.

Домашняя работа

Задача 1