Обобщение пройденного

Типы данных

Числа: 5 9892 10.2 0.342

Строки: ‘строка’ ‘число номер 45’

Списки: [‘строка’, 5, ‘Строка номер 2’]

Кортеж: 5, 3, 10.1 (5, 3, 10.1) (‘один элемент’,)

Словарь: { ‘Ключ’: ‘Значение’, ‘Ключ2’: ‘Значение2’ }

Множекства: { ‘строка1’, ‘строка2’, ‘строка3’ }

Булевые: True False

Отсутствие данных: None

Переменные

Имя переменной: perem1, color, colors, фрукт, фрукты

Присваивание и выражения :

chislo = 5 stroka = ‘Привет Мир’ spisok = [4, 5, 6]

a = b + c * d a = (b+c) * d stroka = ‘STRO‘ + ‘KA’

a = b = 3

Простые типы данных

Числа	Булевые
chislo = 5 float = 2.4	bool_type = True bool_type = False
chislo = chislo* 5 #25 chislo = chislo**2 #25	a = 1 +True # 2
chislo = 5/2 # 2.5 chislo = 5//2 #2	a = 1 + False # 1
# пример с условием	if a > 1: print(True) # пример с условием
a = a + 1 => a += 1	
a = a * 1 => a *= 1	
chislo = 1 + ‘23’ # error chislo = 1 + int(‘23’)	

Последовательности

Строки	Кортежи	Списки
S = ‘Привет мир’	tuple1 = (‘привет’, ‘мир’)	spisok = [‘привет’, ‘мир’]
S += ‘ Здравствуй’	tuple1 += (‘ Здравствуй’)	spisok += [‘ Здравствуй’]
S[1] S[7] = ‘п’ #error	tuple1[1] tuple[1] = ‘пир’ #error	spisok[1] spisok[1] = ‘пир’
S[7:10] # ‘мир’	tuple1[0:2] # (‘привет’, ‘мир’)	spisok[0:2] # [‘привет’, ‘мир’]
len(S) # 21	len(tuple1) # 3	len(spisok) # 3
S.count(‘р’) # 3	tuple1.count(‘мир’) # 1	spisok.count(‘мир’) # 1
S.index(‘и’) # 2 S.index(‘и’,3) #8	tuple1.index(‘мир’) # 1	spisok.index(‘мир’) # 1

Неупорядоченные массивы

Соварь	Множество
slovar = {‘ключ’: ‘значение’, ‘ключ2’: ‘значение2’}	mnoz = {‘значение’, ‘значение2’}
slovar = { } slovar = dict()	mnoz = set()
работает с for	работает с for
slovar.pop(‘ключ2’) # ‘значение2’	mnoz.pop(‘значение2’) # ‘значение2’
len(slovar) # 2	len(mnoz) # 2

Изменяемые и неизменяемые Типы данных

Изменяемые	Неизменяемые
Список	Числа
Словарь	Строки
Множества	Кортеж
	Булевые
	None

Изменяемые:

```
spisok = [5, 1, 4 , 2]
spisok.sort()
print(spisok)   # [1,2,4,5]
spisok2 = spisok.sort()
print(spisok2)  # None
```

Неизменяемые:

```
stroka = ‘‘Привет”
stroka.replace(‘Омл’, ‘Прив’)
print(stroka) # Привет
stroka = stroka.replace(‘Омл’, ‘При’)
print(stroka) # Омлет
```

Инструкции

https://pythonworld.ru/osnovy/klyuchevye-slova-modul-keyword.html

Инструкция удаление: del perem del spisok[1:4] del slovar[‘ключ’]

Условия: if, else, elif

Проверка в условии: in, is

Инструкции в условии: or, and, not

Циклы: for in, while, continue, break

Подключение пакетов, библиотек: import, from import

Другие: global, def, return, class, with

Блочные инструкции

Блочные инструкции: if, elif, else, while, for, def, class, with

Правила: 1)Блок начинается с :

2)Все, что внутри блока должно иметь отступ

3)Блок завершается возвратом от отступа

Вложенности в блоке

```
if a>b:
        print(a)
        if c>b:
                print(c)
        else:
                print(b)
else:
        print(d)
```

Условия

Пять форм условия:

1.
```
if a > b:
    print(a)
```

2.
```
if a > b:
    print(a)
else:
    print(b)
```

3.
```
if a > b:
    print(a)
elif a<b:
    print(b)
else:
    print(c)
```

Однострочный if:

4. print(a) if a > b

5. print(a) if a > b else print(b)

Аналог пункта 3:

```
if a > b:
    print(a)
else:
    print(b) if a < b else print(c)
```

Выражение в условии

Где применяются:

```
if (True или False):
        print(True)

while (True или False):
        print(True)
```

Виды выражений:

1) a>b, a>b>c , a>=b
2) a is b
3) a in b # stroka = ‘мир’ ‘и’ in stroka
4) a
5) a>b and b<c a<b and c
6) a>b or b<c
7) not a not a>b

Циклы

while (условие):
    блок

for (переменная) in (итерируемый тип данных):
    блок

break - останавливает выполнение итераций

continue - пропускает оставшиеся инструкции и выполняет итерации дальше

Итерируемые типы данных:

Строки	проходит по символам
Списки	проходит по элементам
Словари	проходит по ключам
Кортежи	проходит по элементам
Множества	проходит по элементам
range()	проходит по диапозону чисел (на самом деле, это тип данных схожий со списком)

Функции

1) Встроенные функции - функции доступные в питоне из коробки
2) Функции из библиотек - функции которые импортируем из пакетов python
3) Встроенные методы - функции привязанные к типу данных(к объекту)
4) Пользовательские функции - наши собственные функции
5) Пользовательские методы - наши собственные методы

Функции

function(аргумент1, аргумент2)

функция возвращает ответ подобно выражению.

возвращает ответ инструкцией return

ответом может быть любой из типов данных

используя встроенные функции мы их просто вызываем. Описание встроенных функций спрятано от нас.

аргументов может быть и ноль и сколь угодно

Встроенные функции

https://pythonworld.ru/osnovy/vstroennye-funkcii.html

print(), input()	функции ввода и вывода
int(), float(), str(), list(), dict(), list(), set(), tuple()	функции типов данных
min(), max()	минимальный и максимальные элемент послед-ти
len()	возвращает длину строки или массива
abs()	Возвращает абсолютную величину (модуль числа).
sum()	Возвращает сумму элементов последоват-ти
type()	Возвращает тип данных
eval()	Выполняет инструкцию из строки

Методы

Методы это функции, которые привязаны к типу данных (объекту)

Обращаемся к методам через точку.

spisok.sort()

Методы строк

S.find(str, [start],[end])	Поиск подстроки в строке. Возвращает номер первого вхождения или -1
S.rfind(str, [start],[end])	Поиск подстроки в строке. Возвращает номер последнего вхождения или -1
S.index(str, [start],[end])	Поиск подстроки в строке. Возвращает номер первого вхождения или вызывает ValueError
S.rindex(str, [start],[end])	Поиск подстроки в строке. Возвращает номер последнего вхождения или вызывает ValueError
S.replace(шаблон, замена)	Замена шаблона
S.split(символ)	Разбиение строки по разделителю
S.upper(), S.lower()	Преобразование строки к верхнему и нижнему регистру
S.join(список)	Сборка строки из списка с разделителем S
S.format()	форматирование строки
S.strip([chars])	удаление символов.

Методы списков

list.append(x)	Добавляет элемент в конец списка
list.extend(L)	Расширяет список list, добавляя в конец все элементы списка
list.insert(i, x)	Вставляет на i-ый элемент значение x
list.remove(x)	Удаляет первый элемент в списке, имеющий значение x.
list.pop([i])	Удаляет i-ый элемент и возвращает его
list.index(x, [start [, end]])	Возвращает положение первого элемента со значением x
list.count(x)	Возвращает количество элементов со значением x
list.sort()	Сортирует список
list.reverse()	Разворачивает список
list.clear()	Очищает список

Методы словарей

dict.clear()	очищает словарь
dict.get(key[, default])	возвращает значение ключа
dict.items()	возвращает список кортежей (ключ, значение)
dict.keys()	возвращает список ключей словаря
dict.values()	возвращает список значений словаря
dict.update([other])	добавляет в словарь ключ значения из другого
dict.pop(key[, default])	удаляет указанный элемент в словаре