

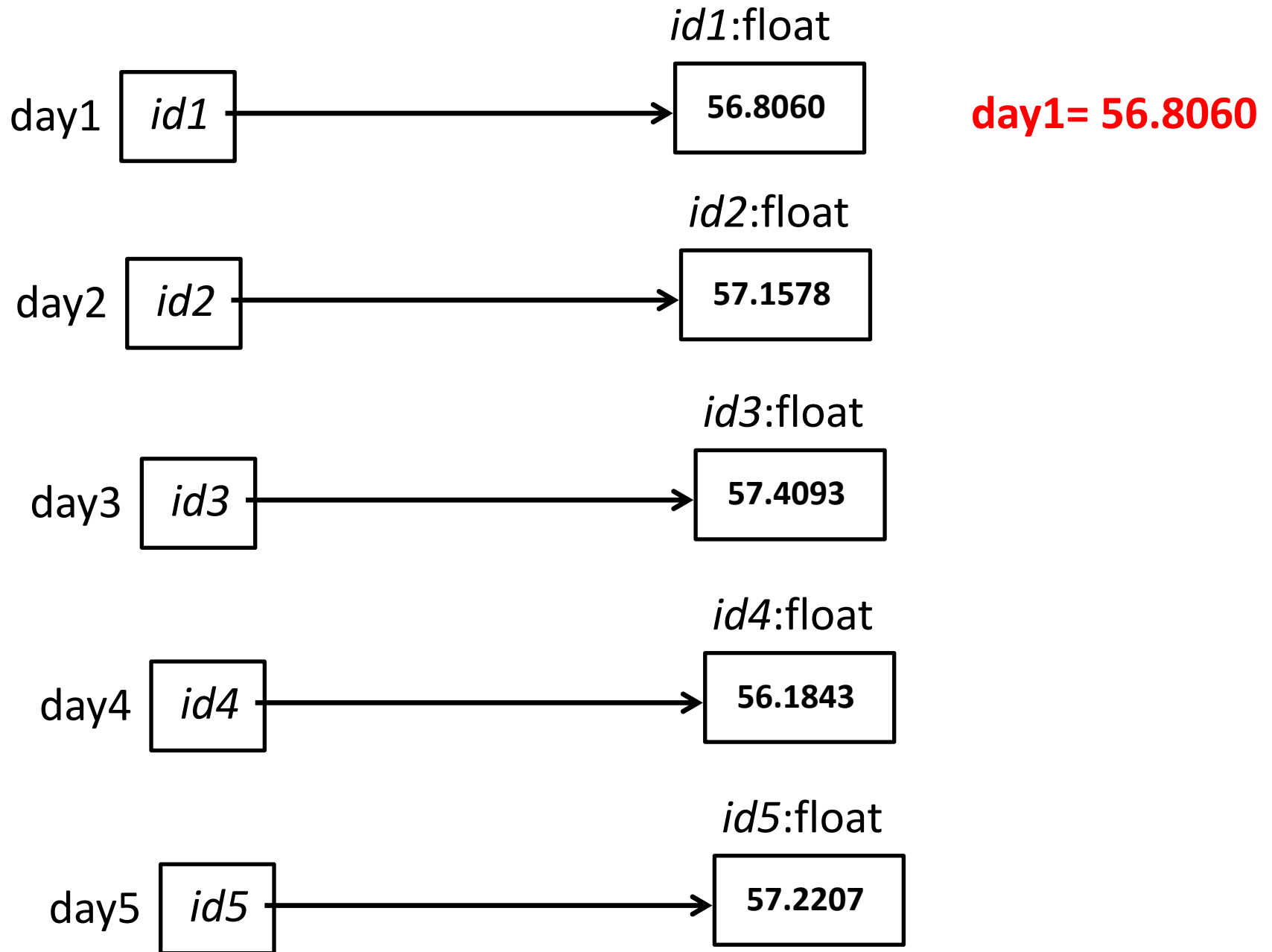


Python: быстрый старт.

Урок 7

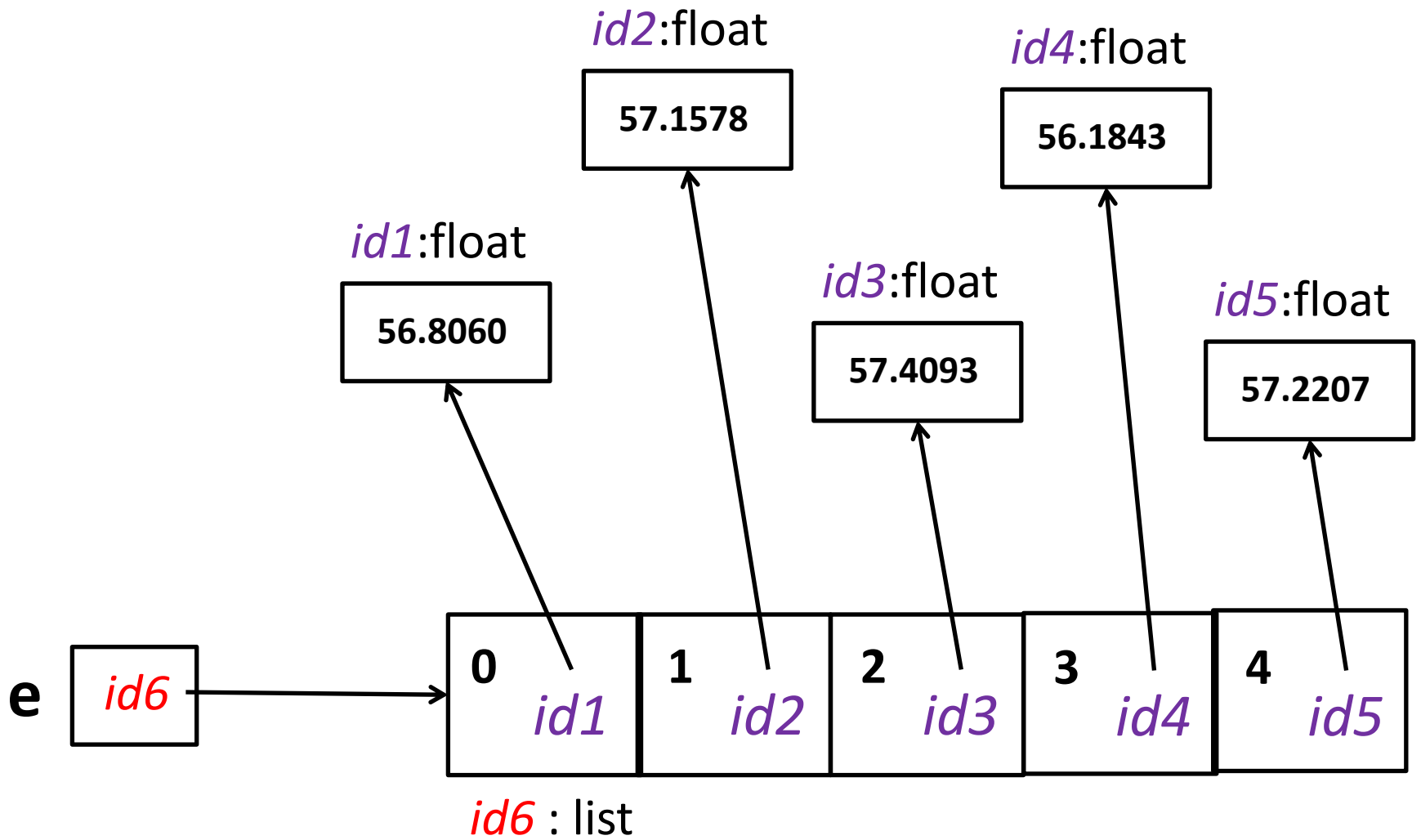
Списки в Python, операции над
списками, срезы,
преобразование типов,
вложенные списки

Дата	Доллар США USD	ЕВРО EUR
16.05.2015	50.0115	56.9881
15.05.2015	50.0774	57.1383
14.05.2015	49.5366	55.7138
13.05.2015	50.9140	57.1102
09.05.2015	50.7511	56.8971
08.05.2015	50.3615	57.2207
07.05.2015	49.9816	56.1843
06.05.2015	51.7574	57.4093
01.05.2015	51.1388	57.1578
30.04.2015	51.7029	56.8060
29.04.2015	52.3041	56.9016
28.04.2015	51.4690	55.8747
25.04.2015	50.2473	54.6590
24.04.2015	51.6011	55.1255
23.04.2015	53.6555	57.7226
22.04.2015	53.9728	57.5998



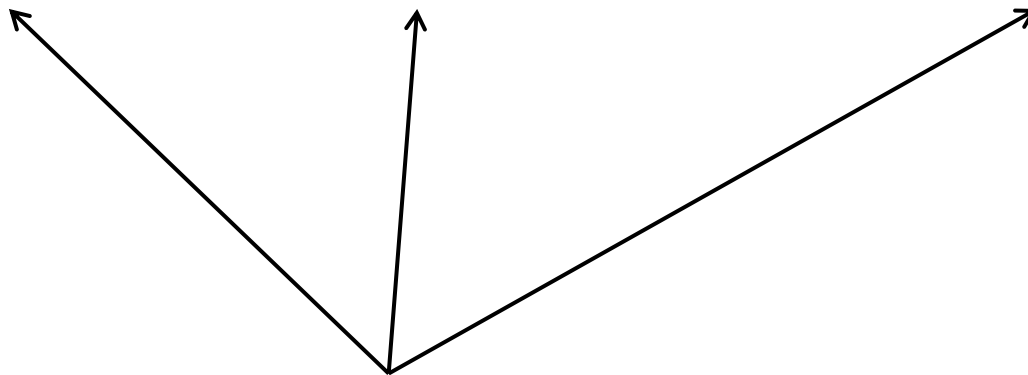
Список (list) является объектом, поэтому может быть присвоен переменной:

```
e=[56.8060, 57.1578, 57.4093, 56.1843, 57.2207]
```



Общая форма

[<<выражение1>>, <<выражение2>>, .. , <<выражениеN>>]



Элементы списка

Пустой список записывается [] и не содержит элементов.

```
e=[56.8060, 57.1578, 57.4093, 56.1843, 57.2207]
```

```
>> e[0]
```

```
>> e[1]
```

```
>> e[100] # ошибка
```

```
>> e[-1]
```

```
>>> h=['bonjour','привет','hola','aloha','привіт']
```

```
>>> h[1]='hello'
```

```
>>> h
```

```
['bonjour', 'hello', 'hola', 'aloha', 'привіт']
```

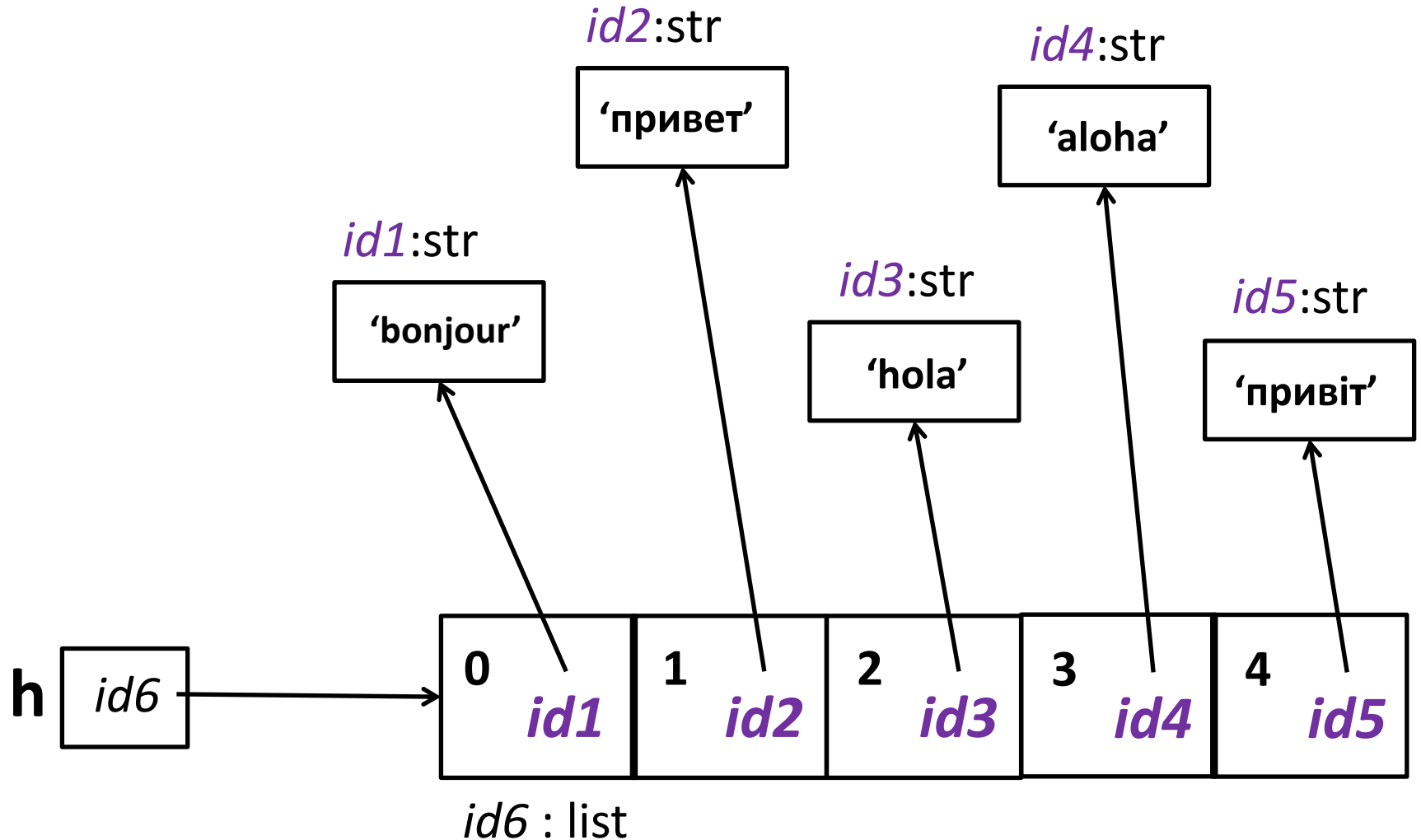
```
>>> h[1]
```

```
'hello'
```

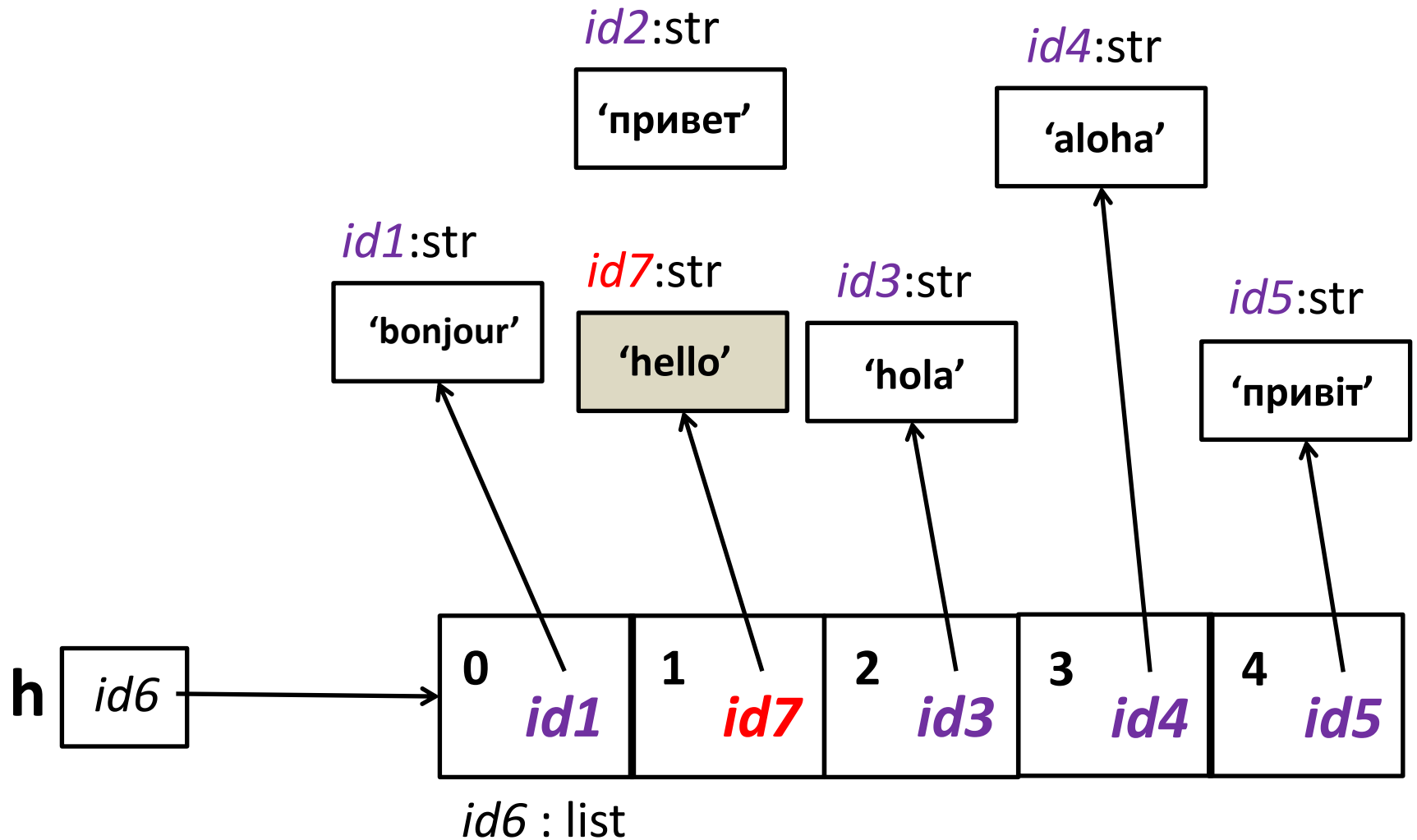
```
>>>
```



```
>>> h=['bonjour','привет','hola','aloha','привіт']
```



```
>>> h[1]='hello'
```



Операции над списками

len(L) – возвращает число элементов в списке L

max(L) – возвращает максимальное значение в списке L

min(L) – возвращает минимальное значение в списке L

sum(L) – возвращает сумму значений в списке L

sorted(L) – возвращает копию списка L, в котором элементы упорядочены по возрастанию. Не изменяет список L.

```
>>> e=[56.8060, 57.1578, 57.4093, 56.1843, 57.2207]
```

```
>>> e
```

```
[56.806, 57.1578, 57.4093, 56.1843, 57.2207]
```

```
>>> len(e)
```

```
5
```

```
>>> max(e)
```

```
57.4093
```

```
>>> min(e)
```

```
56.1843
```

```
>>> sum(e)
```

```
284.7781
```

```
>>> sorted(e)
```

```
[56.1843, 56.806, 57.1578, 57.2207, 57.4093]
```

```
>>> e
```

```
[56.806, 57.1578, 57.4093, 56.1843, 57.2207]
```

```
>>> original=['H','B']
```

```
>>> final=original+['T']
```

```
>>> final
```

```
['H', 'B', 'T']
```

```
>>> final=final*5
```

```
>>> final
```

```
['H', 'B', 'T', 'H', 'B', 'T', 'H', 'B', 'T', 'H', 'B', 'T', 'H', 'B', 'T']
```

```
>>> del final[0]
```

```
>>> final
```

```
['B', 'T', 'H', 'B', 'T', 'H', 'B', 'T', 'H', 'B', 'T', 'H', 'B', 'T']
```

Оператор *in*

```
>>> h=['bonjour',7,'hola',-1.0,'привіт']
```

```
>>> if 7 in h:
```

```
    print('Значение есть в списке')
```

Значение есть в списке

```
>>>
```

В отличие от строк оператор *in* в списке не позволяет проверять подписки.

Срезы

```
>>> h=['bonjour',7,'hola',-1.0,'привіт']
```

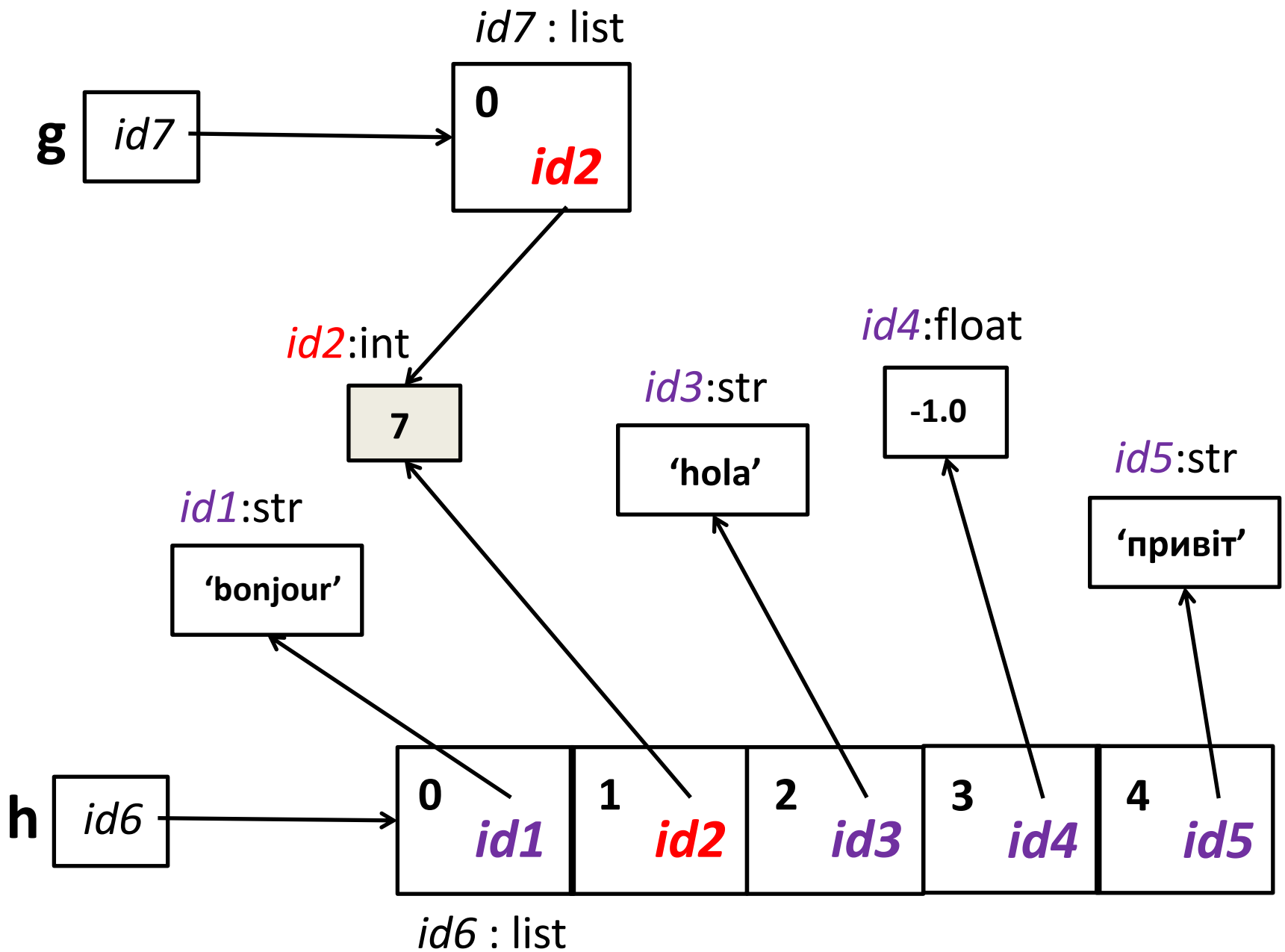
```
>>> h
```

```
['bonjour', 7, 'hola', -1.0, 'привіт']
```

```
>>> g=h[1:2]
```

```
>>> g
```

```
[7]
```



Срезы и клонирование

```
>>> h
```

```
['bonjour', 7, 'hola', -1.0, 'привіт']
```

```
>>> t=h[:] # теперь t – клон h, т.е. ссылается на одинаковые объекты
```

```
>>> t # если модифицировать t, что произойдет с h?
```

```
['bonjour', 7, 'hola', -1.0, 'привіт']
```

```
>>> y=h[:3]
```

```
>>> y
```

```
['bonjour', 7, 'hola']
```

Псевдонимы

Псевдонимы – альтернативные имена чего-либо.

В Python две переменные называются псевдонимами, когда они содержат одинаковые адреса памяти.

```
>>> h
['bonjour', 7, 'hola', -1.0, 'привіт']
>>> p=h # содержат указатель на один и тот же список
>>> p
['bonjour', 7, 'hola', -1.0, 'привіт']
>>> p[0]=1 # модифицируем одну из переменных
>>> h      # изменилась другая переменная!
[1, 7, 'hola', -1.0, 'привіт']
>>> p
[1, 7, 'hola', -1.0, 'привіт']
>>>
```

Надо внимательно обращаться с модифицируемыми объектами, например, со строками такая манипуляция не пройдет, т.к. они не изменяются.

Методы списка

```
>>> colors=['red', 'orange', 'green']
>>> colors.extend(['black','blue'])
>>> colors
['red', 'orange', 'green', 'black', 'blue']
>>> colors.append('purple')
>>> colors
['red', 'orange', 'green', 'black', 'blue', 'purple']
>>> colors.insert(2,'yellow')
>>> colors
['red', 'orange', 'yellow', 'green', 'black', 'blue', 'purple']
>>> colors.remove('black')
>>> colors
['red', 'orange', 'yellow', 'green', 'blue', 'purple']
>>> colors.count('red')
1
>>> colors.index('green')
3
```

Список методов:

<https://docs.python.org/3/tutorial/datastructures.html#more-on-lists>

Методы списка

```
>>> colors
['red', 'orange', 'yellow', 'green', 'blue', 'purple']
>>> colors.pop()
'purple'
>>> colors
['red', 'orange', 'yellow', 'green', 'blue']
>>> colors.reverse()
>>> colors
['blue', 'green', 'yellow', 'orange', 'red']
>>> colors.sort()
>>> colors
['blue', 'green', 'orange', 'red', 'yellow']
>>> colors.clear()
>>> colors
[]
>>>
```

Преобразование типов

```
>>> n=73485384753846538465
```

```
>>> list(str(n))
```

```
['7', '3', '4', '8', '5', '3', '8', '4', '7', '5', '3', '8', '4',  
'6', '5', '3', '8', '4', '6', '5']
```

```
>>> s='d a dd dd gg rr tt yy rr ee'.split()
```

```
>>> s
```

```
['d', 'a', 'dd', 'dd', 'gg', 'rr', 'tt', 'yy', 'rr', 'ee']
```

```
>>>
```

Вложенные списки

```
>>> lst=[['A', 1], ['B',2], ['C',3]]
```

```
>>> lst
```

```
[['A', 1], ['B', 2], ['C', 3]]
```

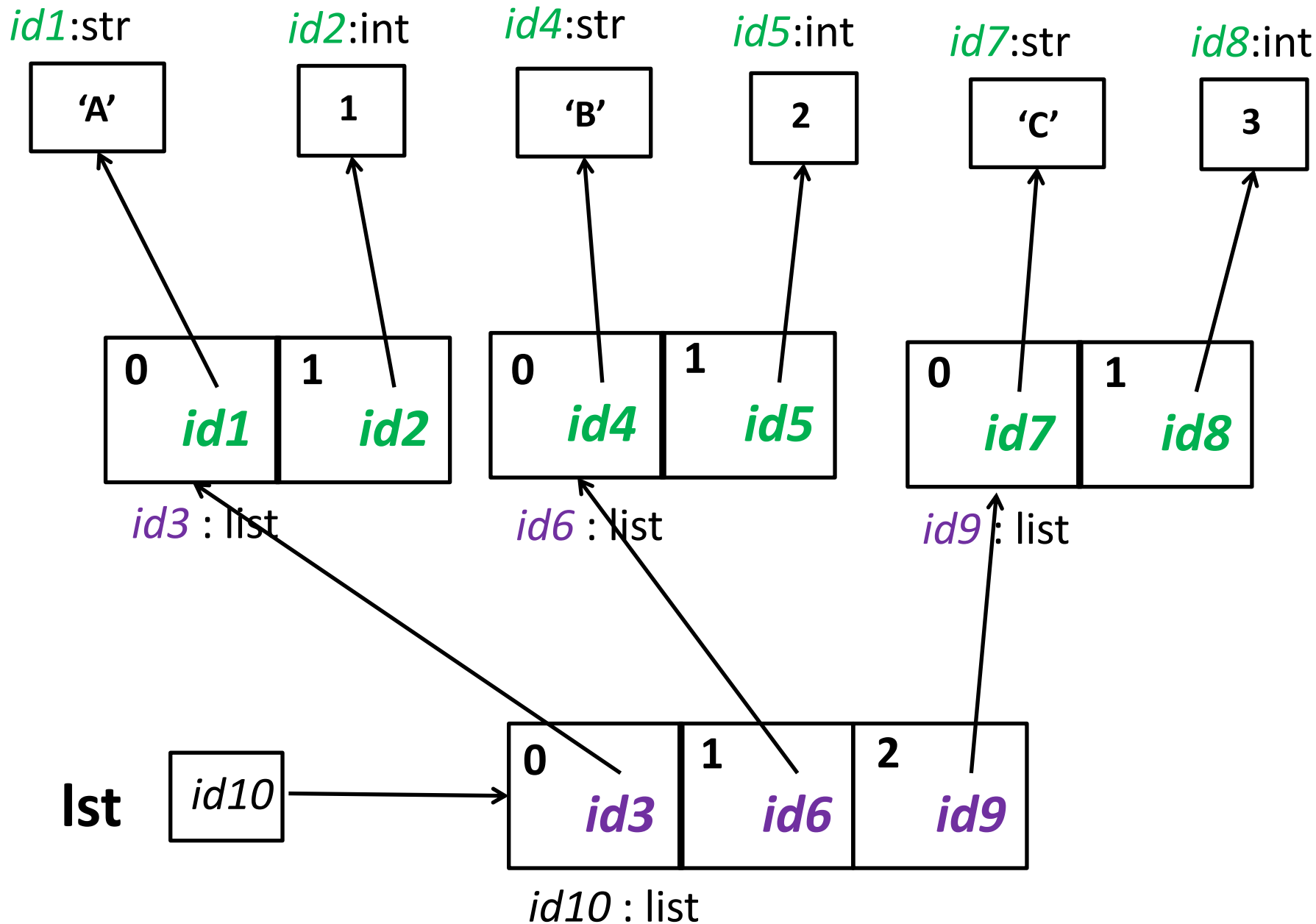
```
>>> lst[0]
```

```
['A', 1]
```

```
>>> lst[0][1]
```

```
1
```

```
>>>
```



Вопросы?

info@dfedorov.spb.ru

Исходные тексты: <http://dfedorov.spb.ru/python3/>