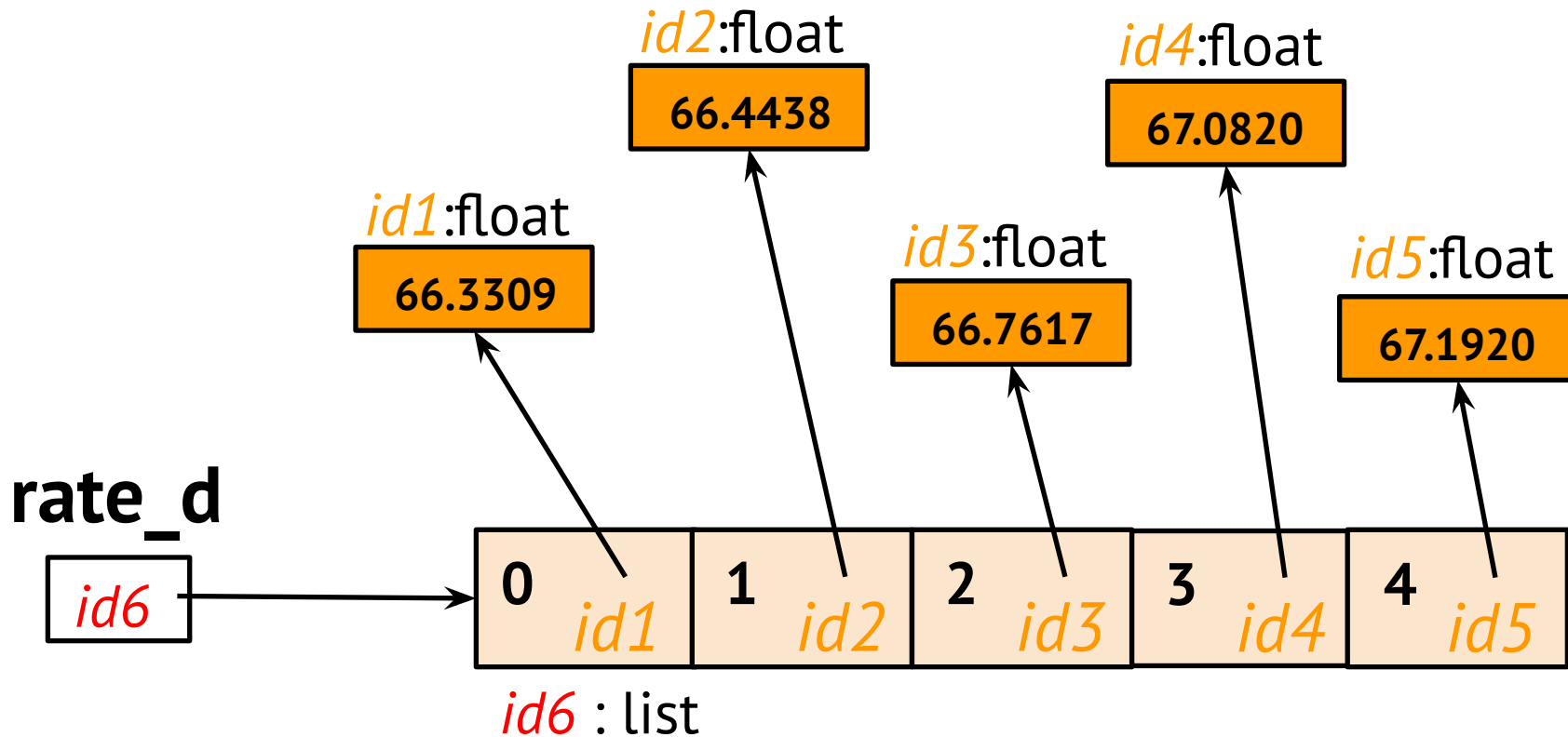


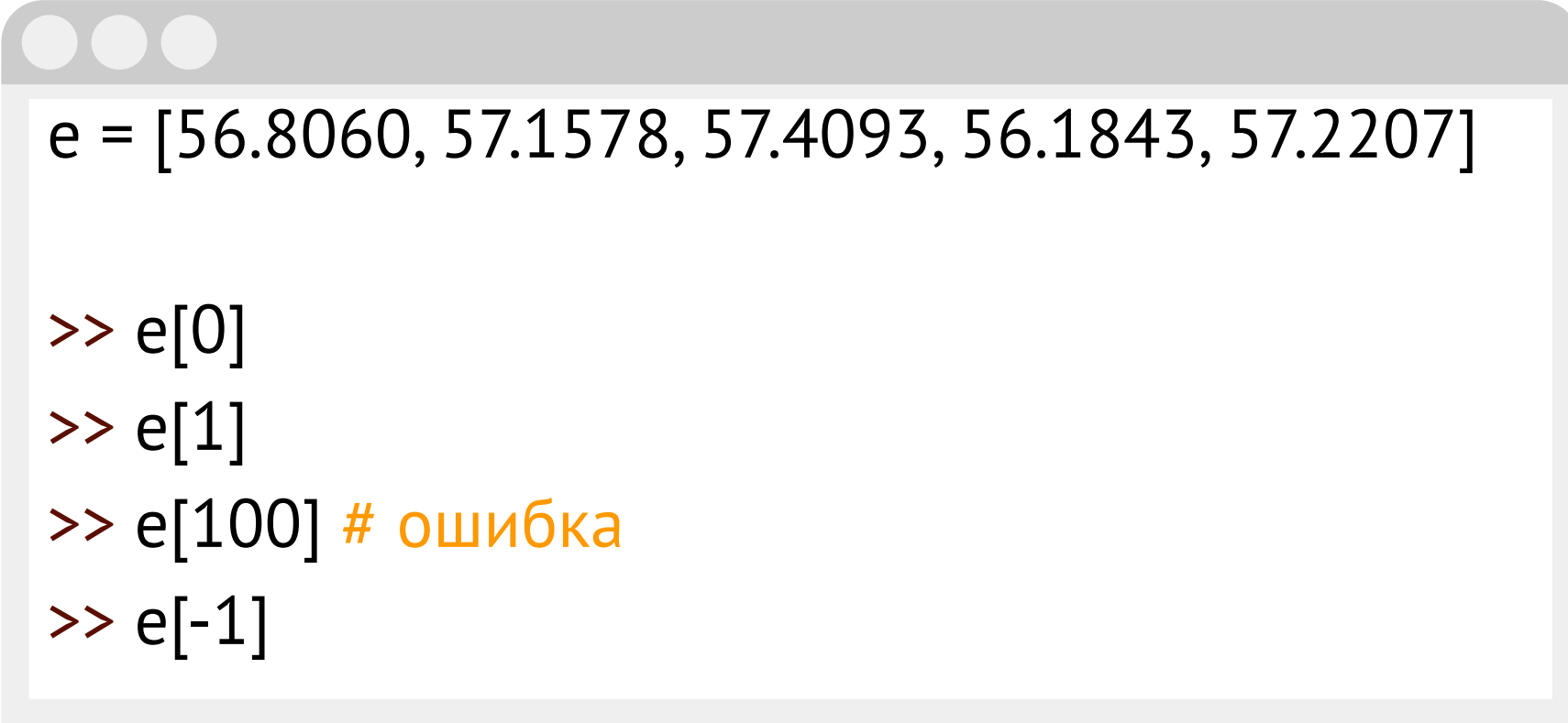


<https://dfedorov.spb.ru>

Список (list) является объектом и может быть присвоен переменной (переменная хранит адрес объекта типа список)

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





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

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

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

```
>> e[100] # ошибка
```

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

```
>>> h = ['Hi', 27, -8.1, [1, 2]]
```

```
>>> h[1] = 'hello' ←
```

```
>>> h
```

```
['Hi', 'hello ', -8.1, [1, 2]]
```

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

```
'hello'
```

```
>>>
```

Изменили
содержимое
списка

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

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

```
>>> h
```

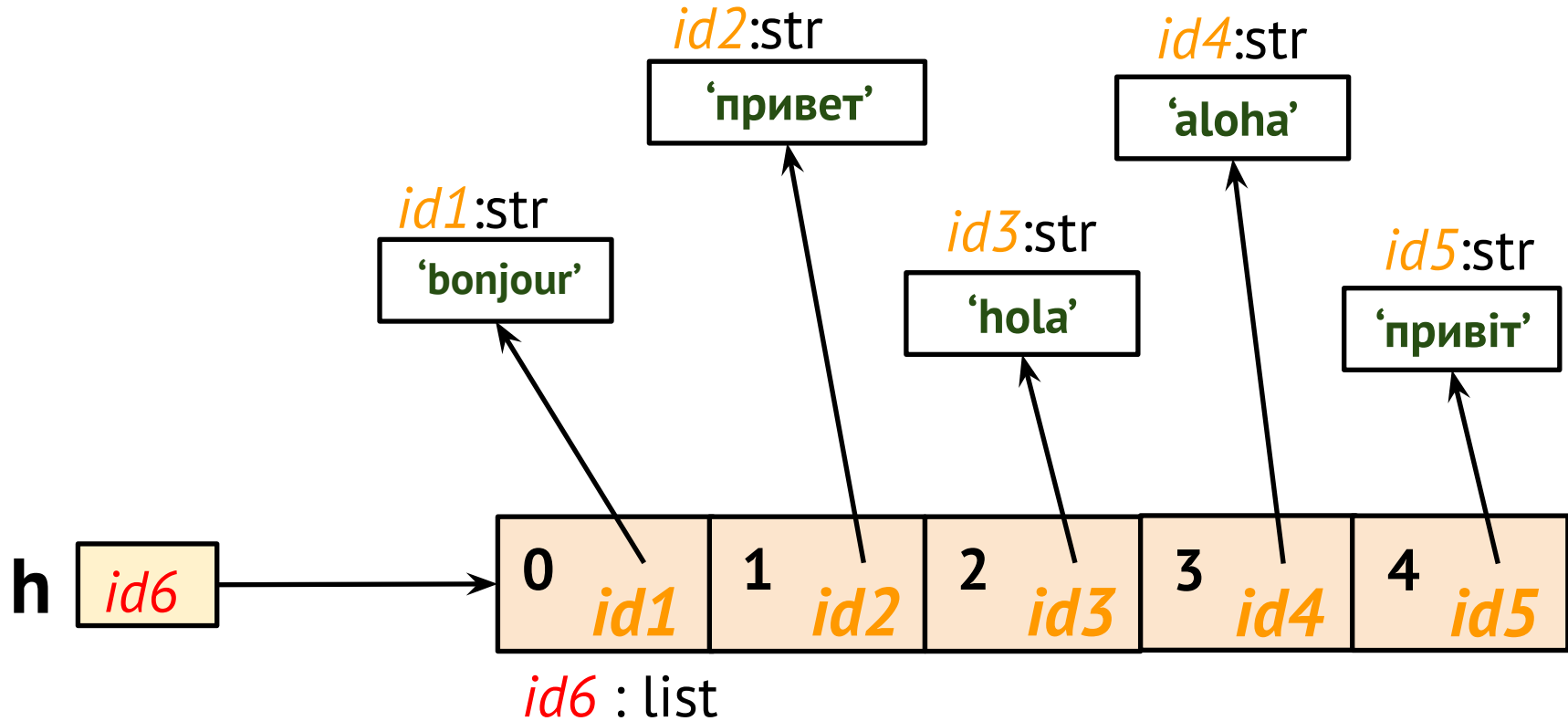
```
['bonjour', 'hello', 'hola', 'aloha', 'привіт']
```

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

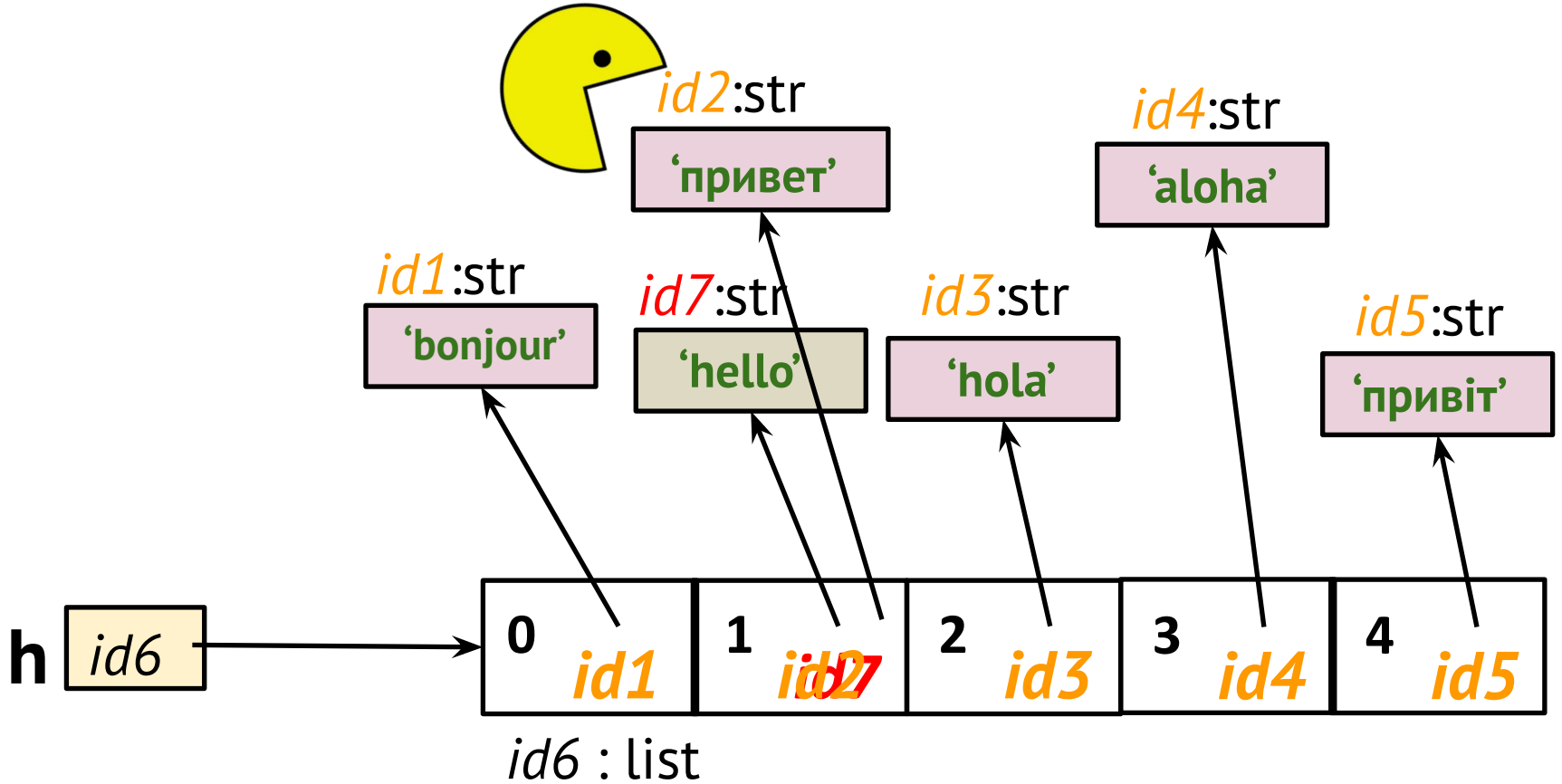
```
'hello'
```

```
>>>
```

```
>>> 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'] # НОВЫЙ СПИСОК
```

```
>>> 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
```

```
>>>
```

Модуль statistics

Модуль [statistics](#) содержит функции для осуществления статистических операций. Как правило, все функции поддерживают целые и вещественные типы.

```
import statistics
```

```
A = [1, 1, 2, 3, 3, 3, 4, 4, 5, 8, 8, 9]
```

```
print(statistics.mean(A))      # 4.25
```

```
print(statistics.median(A))    # 3.5
```

```
print(statistics.mode(A))      # 3
```

```
print(statistics.variance(A))  # 7.4772727272727275
```

```
print(statistics.stdev(A))     # 2.7344602259445514
```

[Описание](#)

[PEP 450 -- Adding A Statistics Module To The Standard Library](#)

Множественное присваивание

```
>>> x, y, z = [1, 2, 3]
```

```
>>> x
```

```
1
```

```
>>> z
```

```
3
```

```
>>> x, y = [1, 2, 3]
```

```
Traceback (most recent call last):
```

```
File "<pyshell#3>", line 1, in <module>
```

```
    x, y = [1, 2, 3]
```

```
ValueError: too many values to unpack (expected 2)
```

```
>>> x, *y = [1, 2, 3]
```

```
>>> x
```

```
1
```

```
>>> y
```

```
[2, 3]
```

***x, y = [1, 2, 3]**

Упражнение

Предположим, список состоит из 10 элементов. Как переместить три последних элемента из конца в начало списка без нарушения их исходного порядка.

[1, 2, 3, 4, 5, 6, 7, **8, 9, 10**]



[**8, 9, 10**, 1, 2, 3, 4, 5, 6, 7]

```
>>> x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
>>> x = x[-3:] + x[:-3]
```



```
>>> a = [1, 3, 5]
>>> b = a
>>> a[:] = [7, 8, 10]
>>> a
[7, 8, 10]
>>> b
[7, 8, 10]
>>> a[:] = [7, 8, 10, 11]
>>> a
[7, 8, 10, 11]
>>>
```

```
>>> lst = [1, 2, 3, 4]
```

```
>>> lst[:3] = [9, 7]
```

```
>>> lst
```

```
[9, 7, 4]
```

Упражнение

1, 2, 3, 4, 5, 6, 7, 8, 9, 10



1, 0, 3, 0, 5, 0, 7, 0, 9, 0

```
>>> x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
>>> x[1::2] = [0] * len(x[1::2])
```

```
>>> x = [1, 2, 3, 4]
>>> x[len(x):] = [5, 6, 7]
>>> x
[1, 2, 3, 4, 5, 6, 7]
>>> x[:0] = [-1, 0]
>>> x
[-1, 0, 1, 2, 3, 4, 5, 6, 7]
>>> x[1:-1] = []
>>> x
[-1, 7]
```



<https://dfedorov.spb.ru>