



Лекция 2

Коллекции

Кисляков Иван



Что такое коллекция?

Коллекция - это объект (переменная), хранящий набор значений одного или различных типов.





Строка



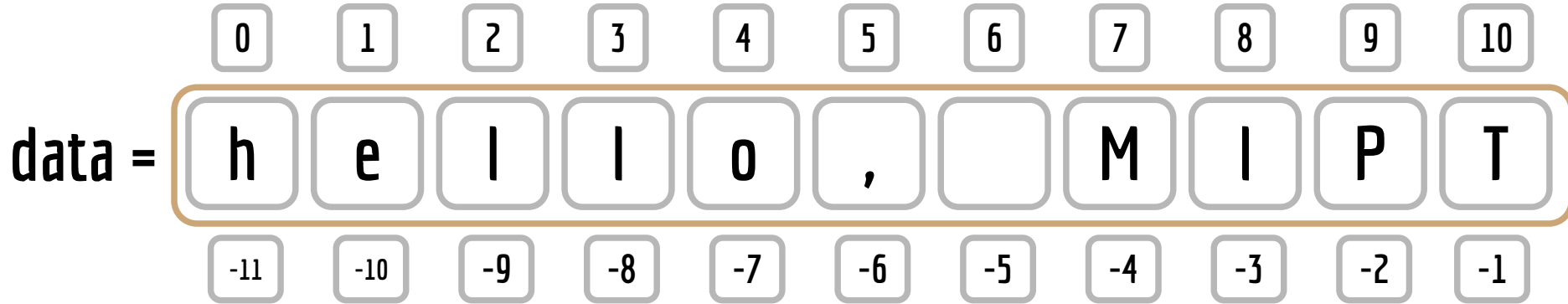
Строка

Строка - коллекция символов.

data =

0	1	2	3	4	5	6	7	8	9	10
h	e	l	l	o	,		M	I	P	T

Строка



```
symb = data[1]  
print(symb) # e
```

```
symb = data[-2]  
print(symb) # Р
```

```
symb = data[12]  
ошибка
```

Строка

data =

0	1	2	3	4	5	6	7	8	9	10
h	e	l	l	o	,		М	I	Р	Т

data[5] = '-'

ошибка

Строка - неизменяемая коллекция.

Строка - срез

data =

0	1	2	3	4	5	6	7	8	9	10
h	e	l	l	o	,		M	I	P	T

```
substr = data[1:5]  
print(substr) # "ello"
```

```
substr = data[:4]  
print(substr) # "hell"
```

```
substr = data[8:]  
print(substr) # "IPT"
```

Строка - срез

data =

0	1	2	3	4	5	6	7	8	9	10
h	e	l	l	o	,		M	I	P	T

```
substr = data[1:6:2]  
print(substr) # "el,"
```

```
substr = data[10:7:-1]  
print(substr) # "TPI"
```


Строка

data =

0	1	2	3	4	5	6	7	8	9	10
h	e	l	l	o	,		M	I	P	T

```
print(len(data)) # 11
```

Цикл

data =

0	1	2	3	4	5	6	7	8	9	10
h	e	l	l	o	,		М	І	Р	Т

Цикл по индексам

```
for i in range(len(data)):  
    print(i, data[i])
```

Цикл по элементам

```
for elem in data:  
    print(elem)
```

Методы

data =

0	1	2	3	4	5	6	7	8	9	10
h	e	l	l	o	,		М	l	P	T

Проверка символа на регистр

```
if data[7].isupper():  
if data[1].islower():
```

Изменение регистра строки

```
data = data.upper()  
data = data.lower()
```

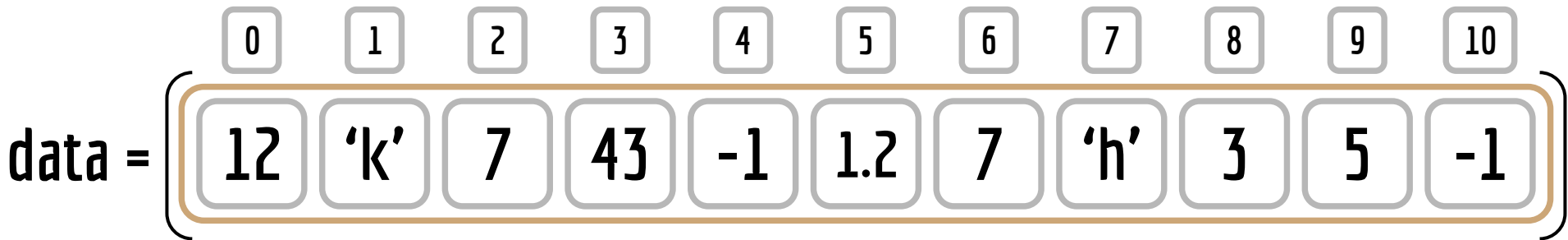


Список



Список

Список - изменяемая
коллекция элементов.



```
data = [12, 'k', 7, 43, -1]  
print(data[0]) # 12
```

```
data[1] = 'Y'  
print(data) # [12, 'Y', 7, 43, -1]
```

Список

Изменение
коллекции

+

```
data = [1, 2, 3, "MIPT"]  
data.append(8)  
print(data)  
[1, 2, 3, 'MIPT', 8]
```

-

```
data = [1, 2, 7, "MIPT", 7]  
data.remove(7)  
print(data)  
[1, 2, 'MIPT', 7]
```

Изменение списка

Изменение
коллекции

```
data.append(val)
```

```
data.insert(pos, val)
```

```
data[3] = 'a'
```

```
data = list()  
data = []
```

```
data.remove(val)
```

```
del data[3]
```

```
del data[1:7]
```

Список

Прекрасные
методы

data =

0	1	2	3	4	5	6	7	8	9	10
12	'k'	7	43	-1	1.2	7	'h'	3	5	-1

3 in data -> True
4 in data -> False

len(data) -> 11

Список - срез

data =

0	1	2	3	4	5	6	7	8	9	10
12	'k'	7	43	-1	1.2	7	'h'	3	5	-1

Аналогично срезу строк

```
substr = data[1:6:2]  
print(substr) # ['k', 43, 1.2]
```

```
substr = data[:7:-1]  
print(substr) # [-1, 5, 3]
```

Цикл

data =

0	1	2	3	4	5	6	7	8	9	10
12	'k'	7	43	-1	1.2	7	'h'	3	5	-1

Цикл по индексам

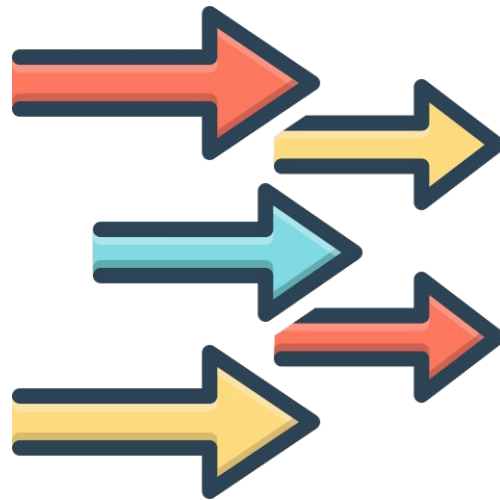
```
for i in range(len(data)):
    print(i, data[i])
```

Цикл по элементам

```
for elem in data:
    print(elem)
```

Сравнение

	string	list
изменить	-	+
разные т.д.	-	+
повтор значений	+	+
индексы	+	+
срезы	+	+
цикл	2	2





Кортеж



Кортеж

Кортеж - неизменяемая коллекция элементов.
(неизменяемая копия list)

```
data = (1, 2, 7, "MIPT", 7)
```

```
data_2 = 1, 2
```

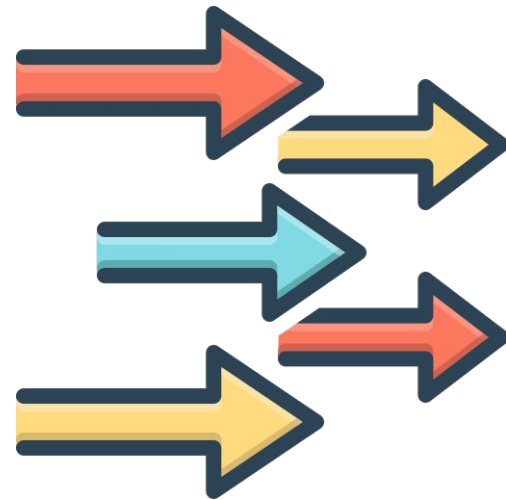
```
data_3 = 1,
```

```
print(data, data_2, data_3)
```

```
(1, 2, 7, 'MIPT', 7) (1, 2) (1,)
```

Сравнение

	string	list	tuple
изменить	-	+	-
разные т.д.	-	+	±
повтор значений	+	+	+
индексы	+	+	+
срезы	+	+	+
цикл	2	2	2





Множество



Множество

Множество - изменяемая неупорядоченная коллекция элементов.

```
data_1 = {1, 2, 3}
```

```
data_2 = set()
```

```
print(data_1, data_2)
```

```
{1, 2, 3} set()
```


Методы множества

Изменение
коллекции

`data.add(val)`

`data.remove(val)`

Уникальность
значений

`{1, 2, 3, 2, 3}`



`{2, 3, 1}`

Другие методы

`len(data)`

`elem in data`

Цикл

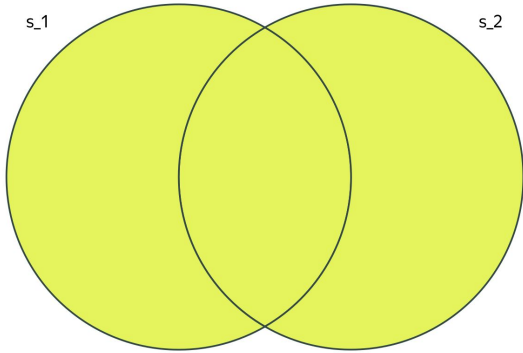
Цикл по индексам

```
for i in range(len(data)):  
    print(data[i])
```

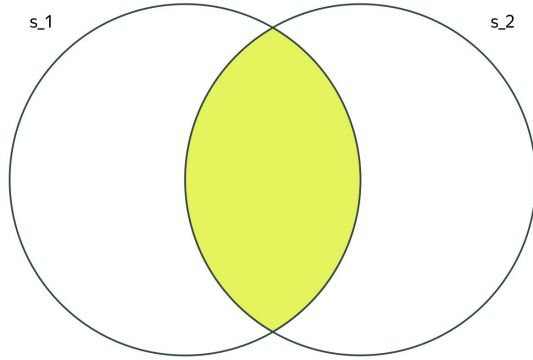
Цикл по элементам

```
for elem in data:  
    print(elem)
```

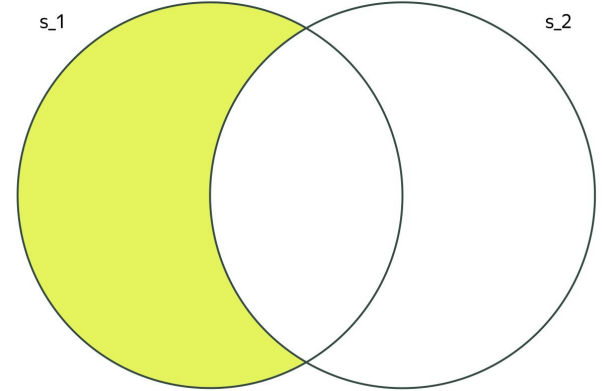
Операции над множествами



$s_3 = s_1.union(s_2)$



$s_3 = s_1.intersection(s_2)$



$s_3 = s_1.difference(s_2)$

Сравнение

	string	list	tuple	set
изменить	-	+	-	+
разные т.д.	-	+	±	+
повтор значений	+	+	+	-
индексы	+	+	+	-
срезы	+	+	+	-
цикл	2	2	2	1

скорость

Преобразование коллекций

STR



SET

`set("abca") == {"a", "c", "b"}`

LIST



SET

`set([1, 2, 2, 3]) == {3, 1, 2}`

TUPLE



LIST

`list((1, 2, 3, 2)) == [1, 2, 3, 2]`

id

память ПК

переменная

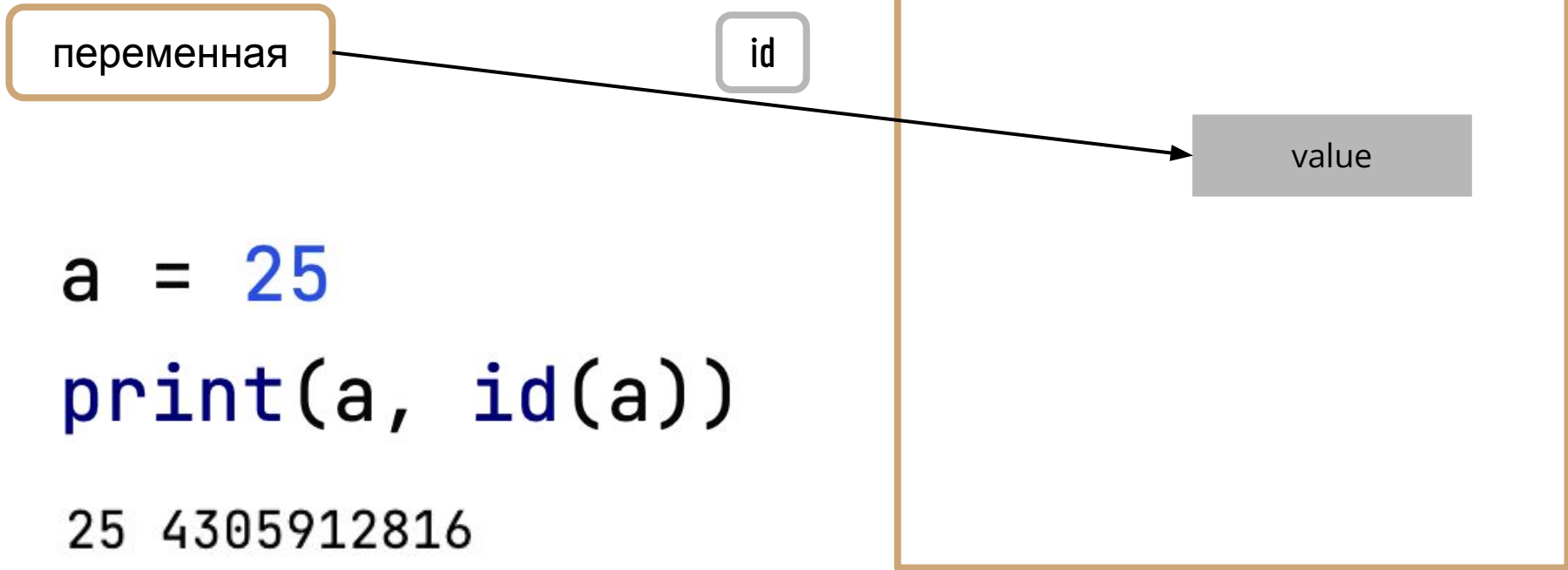
id

value

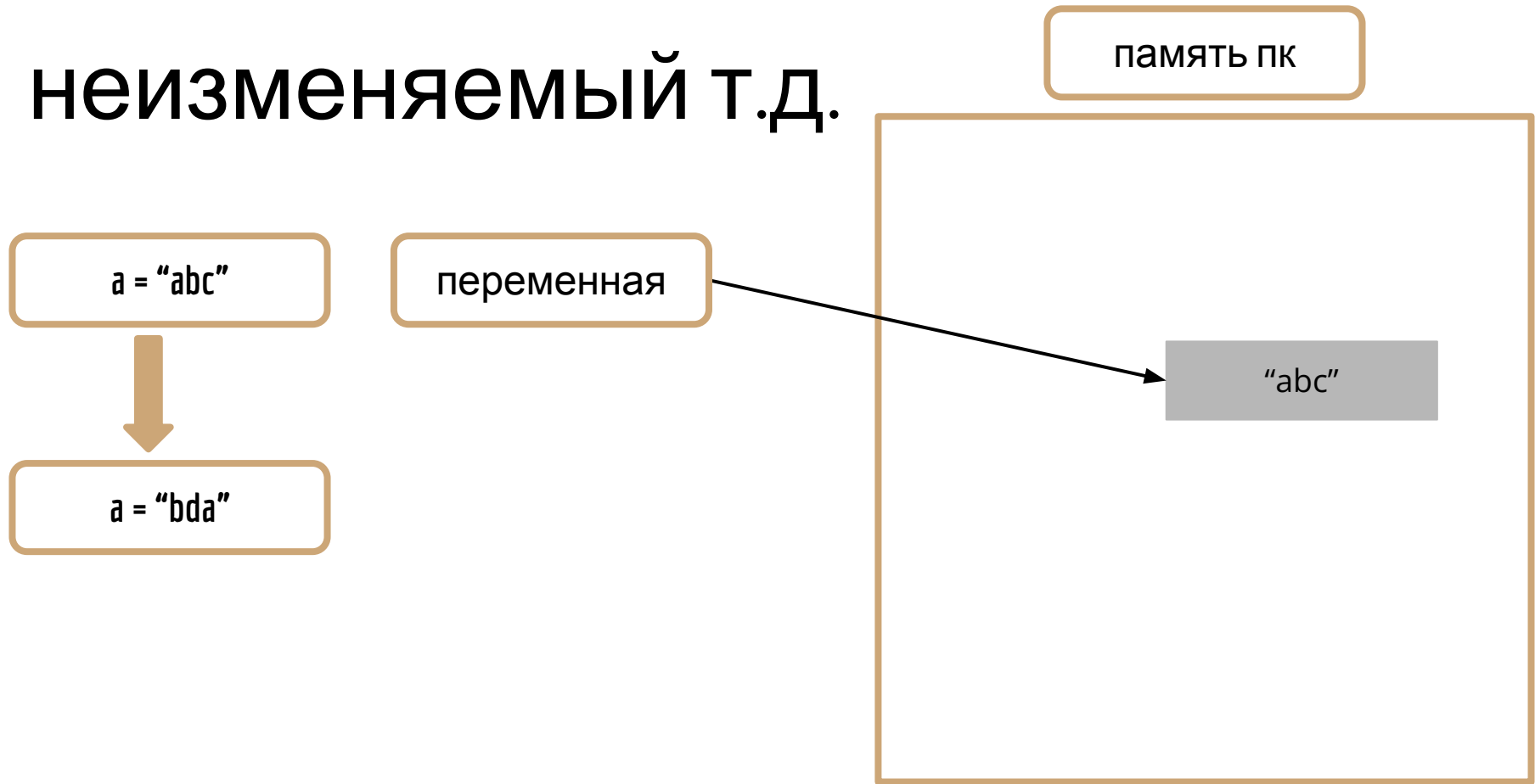
```
a = 25
```

```
print(a, id(a))
```

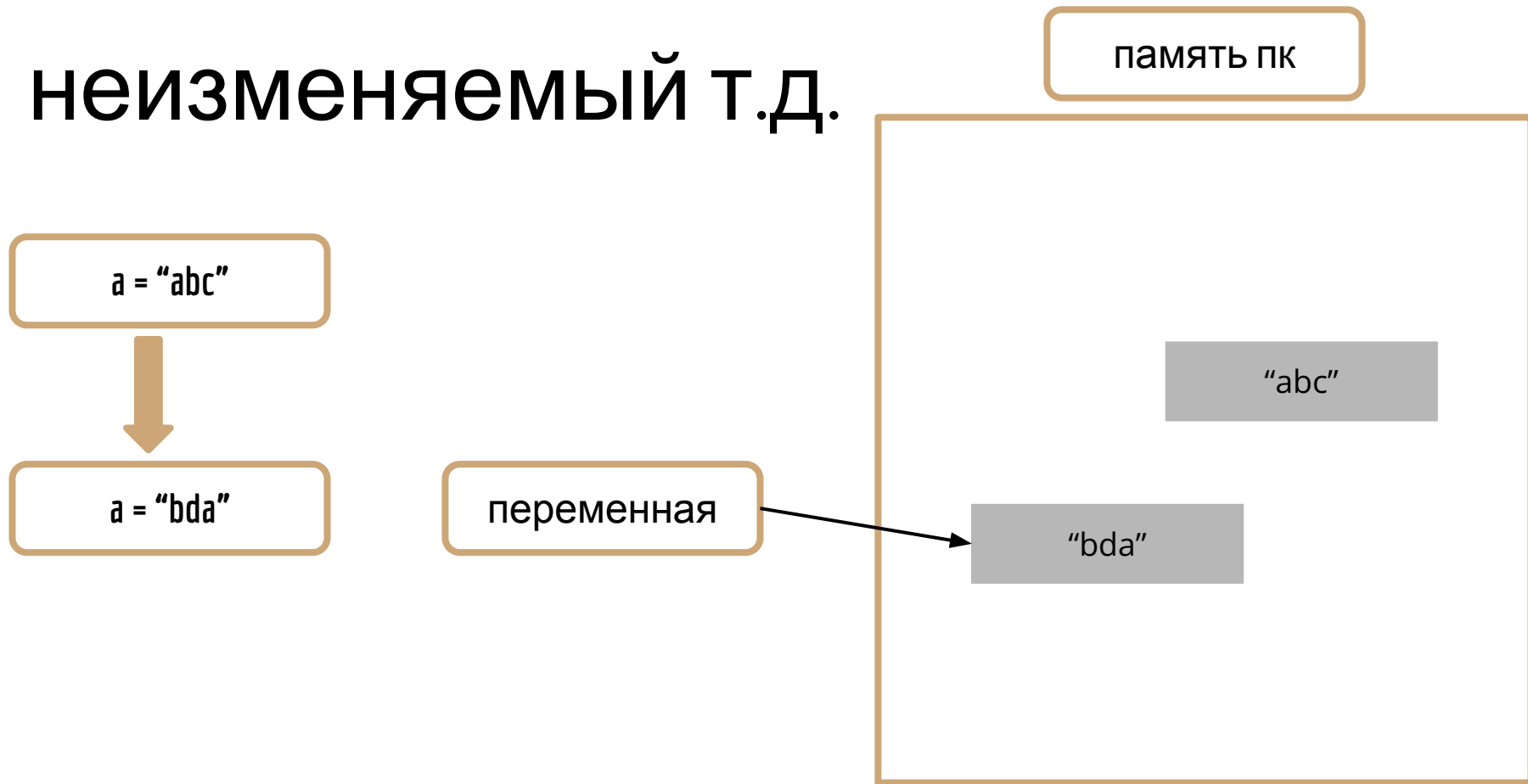
```
25 4305912816
```



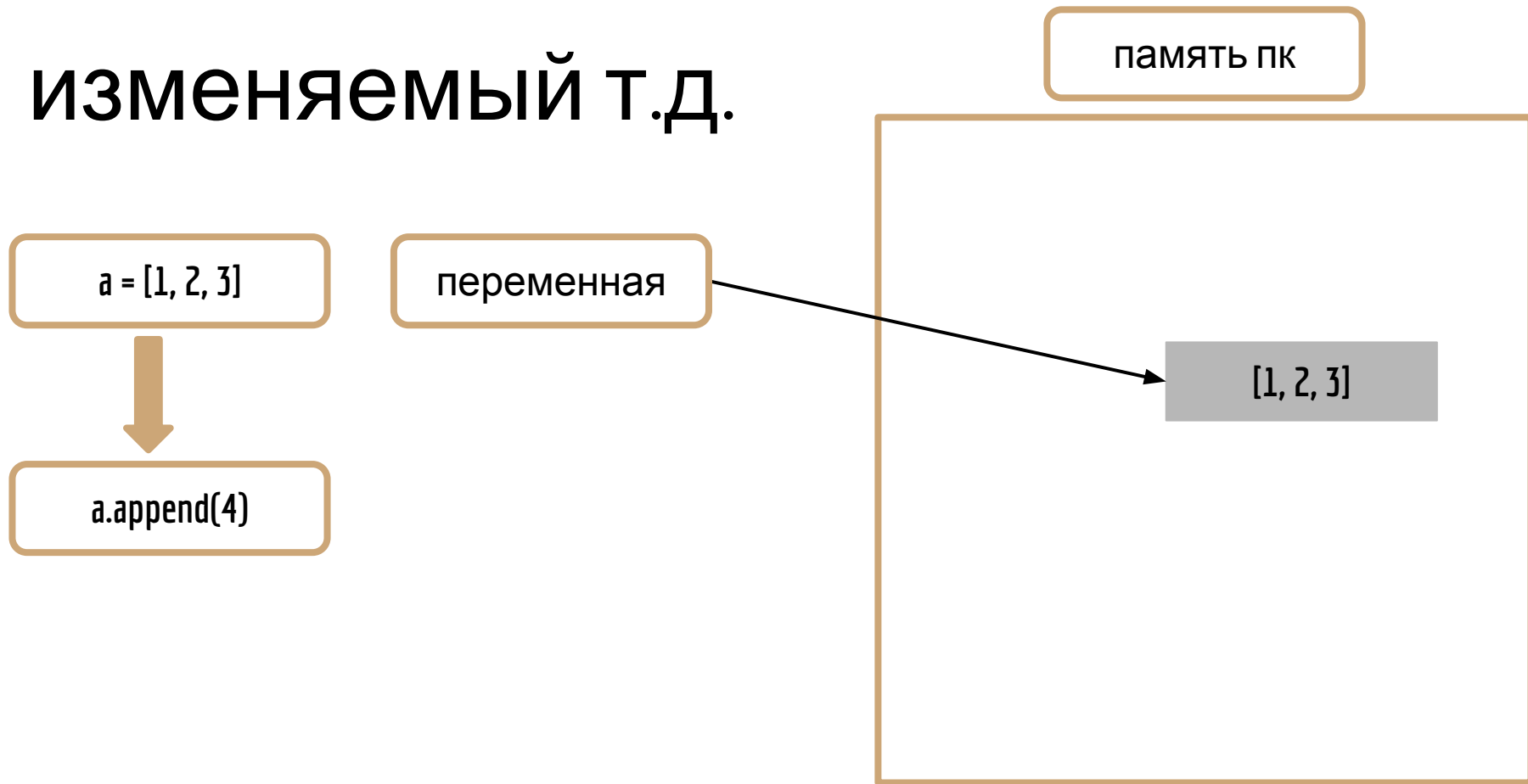
неизменяемый т.д.



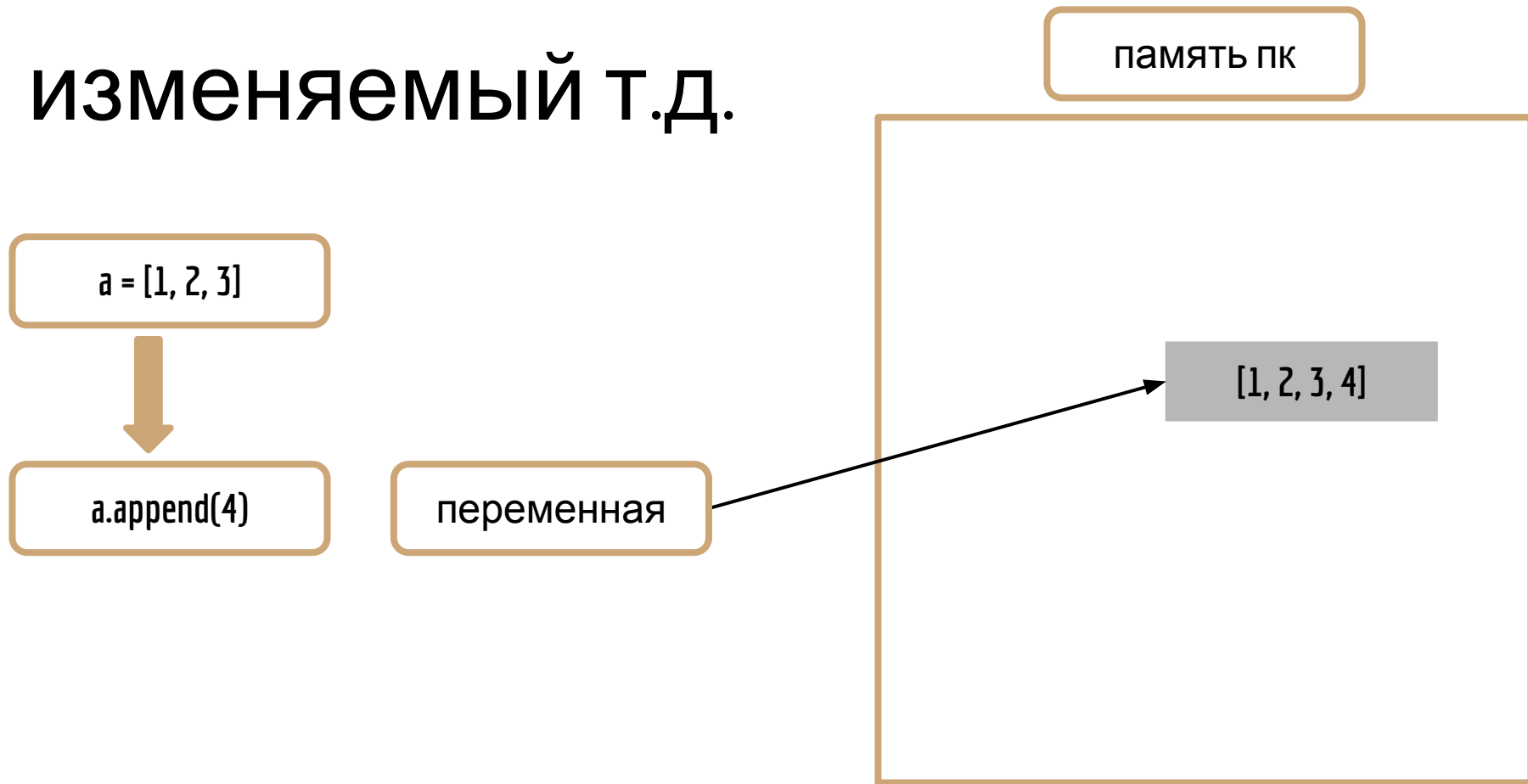
неизменяемый т.д.



ИЗМЕНЯЕМЫЙ Т.Д.



изменяемый т.д.





Словарь



Словарь

неизм. тд, уникальное

любое

key	value
0	"Hello"
'a'	[1, 2, 3, 4, 5, 2]
(1,2)	'Hello'
13	1.2

Словарь -
изменяемая
неупорядоченная
коллекция пар
ключ-значение.

Создание словаря

```
dct = {}
```

```
dct_2 = dict()
```

```
data = {1: "hello", "a": [1, 2, 3]}
```

Изменение значений

```
data = {1: "hello", "a": [1, 2, 3]}  
data["a"] = 13  
print(data)
```

```
{1: 'hello', 'a': 13}
```

Создание значений

```
data = {1: "hello", "a": [1, 2, 3]}  
data["b"] = 7  
print(data)
```

```
{1: 'hello', 'a': [1, 2, 3], 'b': 7}
```

Удаление значений

```
data = {1: "hello", "a": [1, 2, 3]}
```

```
del data['a']
```

```
print(data)
```

```
{1: 'hello'}
```


Интересные методы

`key in data`

`data.keys()`

`len(data)`

`data.values()`



Цикл

Цикл по парам

```
for key, elem in data.items():  
    print(key, elem)
```

Сравнение

	string	list	tuple	set	dict
изменить	-	+	-	+	+
разные т.д.	-	+	±	+	+
повтор значений	+	+	+	-	+
индексы	+	+	+	-	++
срезы	+	+	+	-	-
цикл	2	2	2	1	3