

# PYTHON

---

управляющие конструкции  
списки, кортежи

# If in Python

```
>>> x = int(raw_input("Введите число "))
>>> if x < 0:
...     x=0
...     print 'отрицательное'
... elif x == 0:
...     print '0'
... elif x == 1:
...     print '1'
... else:
...     print 'больше 1'
```

# If in Python

- elif, else может не быть
- ==, !=, >=, <= как в C
- сокращенная запись  
x = a if (condition) else b

# Логические выражения

- Логические значения: True, False
- Логические операции: and, or, not

A or not B and C эквивалентно (A or ((not B) and C))

- 0, [], "" - ложь, иначе — истина

# While

```
>>> # ряд Фибоначчи
```

```
... a, b = 0, 1
```

```
>>> while b < 10:
```

```
...     print b,
```

```
...     a, b = b, a+b
```

```
...
```

```
1 1 2 3 5 8
```

# Pass

Не делает ничего, вставляется туда, где должна быть инструкция

```
>>> while 1:  
...     pass
```

# Списки

```
>>> a = ['spam', 'eggs', 100, 1234]
```

```
>>> a[0]  
'spam'
```

```
>>> a[-2]  
100
```

```
>>> a[1:-1]  
['eggs', 100]
```

`a[:]` - создание полной копии списка

# Списки

Легко объединять, можно умножать

```
>>> a = ['spam', 'eggs', 100, 1234]
```

```
>>> a[:2] + ['bacon', 2*2]  
['spam', 'eggs', 'bacon', 4]
```

```
>>> 3*a[:3] + ['Boe!']  
['spam', 'eggs', 100, 'spam', 'eggs', 100, 'spam',  
'eggs', 100, 'Boe!']
```



# Списки. Изменяемы

```
>>> a = ['spam', 'eggs', 100, 1234]
```

```
>>> a[2] = a[2] + 23
```

```
>>> a
```

```
['spam', 'eggs', 123, 1234]
```

```
>>> a[0:2] = [1, 12]          #Присваивание срезу
```

```
>>> a
```

```
[1, 12, 123, 1234]
```

```
>>> a[0:2] = []
```

```
>>> a
```

```
[123, 1234]
```

# Списки. Добавление элемента

```
>>> a_list = ['a']
```

```
>>> a_list = a_list + [2.0, 3]
```

```
>>> a_list
```

```
['a', 2.0, 3]
```

```
>>> a_list.append(True)
```

```
>>> a_list
```

```
['a', 2.0, 3, True]
```

## Списки. Добавление элемента

```
>>> a_list.extend(['four', 'Ω'])
```

```
>>> a_list
```

```
['a', 2.0, 3, True, 'four', 'Ω']
```

```
>>> a_list.insert(0, 'Ω')
```

```
>>> a_list
```

```
['Ω', 'a', 2.0, 3, True, 'four', 'Ω']
```

# Списки. Удаление элемента

```
>>> a_list = ['a', 'b', 'new', 'mpilgrim', 'new']
```

```
>>> del a_list[1]
```

```
>>> a_list  
['a', 'new', 'mpilgrim', 'new']
```

```
>>> a_list.remove('new')
```

```
>>> a_list  
['a', 'mpilgrim', 'new']
```

# Списки. Удаление элемента

```
>>> a_list.pop()
```

```
'new'
```

```
>>> a_list
```

```
['a', 'mpilgrim']
```

```
>>> a_list.pop(1)
```

```
'mpligrim'
```

# Кортежи

Значения менять нельзя!

```
>>> t = 12345, 54321, 'hello!'
```

```
>>> t[0]
```

```
12345
```

```
>>> t
```

```
(12345, 54321, 'hello!')
```

```
>>> u = t, (1, 2, 3, 4, 5)
```

```
>>> u
```

```
((12345, 54321, 'hello!'), (1, 2, 3, 4, 5))
```

# Кортежи

```
>>> empty = ()
```

```
>>> singleton = 'hello',      # запятая обязательна
```

```
>>> singleton  
('hello',)
```

# Кортежи. Распаковка.

```
>>> # упаковка
```

```
... t = 12345, 54321, 'hello!'
```

```
>>> # распаковка
```

```
... a, b, c = t
```

```
>>> a
```

```
12345
```

```
>>> c
```

```
'hello!'
```



# Множественное присваивание

MONDAY, TUESDAY, WEDNESDAY, THURSDAY,  
FRIDAY, SATURDAY, SUNDAY = range(7)

! обязательно слева столько элементов, сколько  
в кортеже

# Кортежи и списки.

Можно преобразовывать друг в друга

`tuple() # list → tuple`

`list() # tuple → list`

## For. Range.

```
>>> list = ['It', 'is an', 'interesting', 'lecture']
```

```
>>> for x in list:
```

```
...     print x,
```

```
...
```

```
It is an interesting lecture
```

```
>>> range(10)
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
>> range(5, 10)
```

```
# диапазон [5, 6, 7, 8, 9]
```

```
>>> range(0, 10, 3)
```

```
# задаем шаг [0, 3, 6, 9]
```

минусы range?

# For. Xrange.

псевдосписок — объект, для которого мы можем получить значения “элементов”, но не можем изменить их или порядок их следования.

```
>>> xrange(0, 10, 3)
```

```
(0, 3, 6, 9)
```

```
>>> xrange(-10, -100, -30)
```

```
(-10, -40, -70)
```

```
>>> a = ['Что-то', 'новенькое', '?']
```

```
>>> for i in xrange(len(a)):
```

```
...     print a[i],
```

```
...
```

```
Что-то новенькое ?
```

# Break. Continue. Else.

break, continue – как в C

```
>>> for element in list:
...     if element_to_find == element:
...         print element, 'элемент найден'
...         break
...     else:
...         print element1, 'элемент не найден'
```

Блок else выполняется, если вышли **не** через break

## Еще о списках

```
>>> li = [1, 9, 8, 4]
```

```
>>> [elem*2 for elem in li]  
[2, 18, 16, 8]
```

```
>>> li
```

```
[1, 9, 8, 4]
```

# Импорты

```
>>> from urllib2 import urlopen
```

```
>>> from urllib2 import *
```

```
>>> import urllib2.urlopen
```