



Курс-интенсив

Python для быстрого старта в IT

Обзор языка Python

academy.rubius.com

konstantin.dobrychev@rubius.com

Константин Добрычев

Текст программы

```
number = 42
```

```
it_is_a_list = True
```

```
value = [1, 2, 3, 8] if it_is_a_list else number
```

```
print(value)
```

Текст программы

```
name = 'Alice'
```

```
text = "it's a string"
```

```
multi_line_text = '''  
    line_1  
    line_2  
    line_3  
'''
```

Комментарии

```
# Это просто комментарий
```

```
a = 3 + 3 # Комментарии могут быть и тут
```

```
def user_info(name, age, city):
```

```
    """
```

```
    Возвращает информацию о пользователе.
```

```
    :param name: имя
```

```
    :param age: возраст в годах
```

```
    :param city: город
```

```
    """
```

```
    return name + ' ' + age + ', ' + city
```

```
#!/usr/bin/env python3
```

```
# -*- coding: utf-8 -*-
```

Ключевые слова

False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

Переменные

Переменная — поименованная, либо адресуемая иным способом область памяти, адрес которой можно использовать для осуществления доступа к данным. Данные, находящиеся в переменной (то есть по данному адресу памяти), называются **значением** этой переменной.

```
number: int = 42
```

```
it_is_a_list: bool = True
```

```
value: object = [1, 2, 3, 8] if it_is_a_list else number
```

```
print(value)
```

```
name: str = 'Alice'
```

Встроенные типы

<https://docs.python.org/3/library/stdtypes.html#boolean-operations-and-or-not>

<https://docs.python.org/3/library/stdtypes.html#typesnumeric>

<https://docs.python.org/3/library/stdtypes.html#sequence-types-list-tuple-range>

<https://docs.python.org/3/library/stdtypes.html#set-types-set-frozenset>

<https://docs.python.org/3/library/stdtypes.html#mapping-types-dict>

СПИСКИ

```
numbers = [1, 2, 3, 5, 6]  
users = ['Alice', 'Bob', 'Carlos']  
things = [1, True, 'Text', None, [1, 2, 3]]
```

```
numbers += [34, 35, 35]  
print(numbers[2])  
print(numbers[1:3], numbers[2:-2])
```


Кортежи

```
location = (56.484640, 84.947649, 'Tomsk')
```

```
latitude, longitude, label = location
```

```
assert location[0] == latitude
```

```
assert location[1] == longitude
```

```
assert location[2] == label
```

Словари

```
location = {  
    'latitude': 56.484640,  
    'longitude': 84.947649,  
    'label': 'Tomsk',  
}
```

```
assert location['latitude'] == 56.484640  
assert location['longitude'] == 84.947649  
assert location['label'] == 'Tomsk'
```

```
location['label'] = 'Tomsk, Russia'  
location['population'] = 500_000
```

Перечисления

<https://docs.python.org/3/library/enum.html>

```
from enum import Enum
```

```
class DefaultColor(Enum):
```

```
    Red = 0,  
    Green = 1,  
    Blue = 2,  
    Black = 3,  
    White = 4,  
    Transparent = 5
```

```
class
```

```
DefaultColor(Enum):
```

```
    Red = '#ff0000',  
    Green = '#00ff00',  
    Blue = '#0000ff',  
    Black = '#ffffff',  
    White = '#000000',
```

```
class DefaultColor(Enum):
```

```
    Red = (255, 0, 0),  
    Green = (0, 255, 0),  
    Blue = (0, 0, 255),  
    Black = (255, 255, 255),  
    White = (0, 0, 0),
```

Классы данных

<https://docs.python.org/3/library/dataclasses.html>

```
from dataclasses import dataclass
```

```
@dataclass
```

```
class Location:
```

```
    latitude: float
```

```
    longitude: float
```

```
    label: str
```

```
location = Location(56.484640, 84.947649, 'Tomsk')
```

```
assert location.latitude == 56.484640
```

```
assert location.longitude == 84.947649
```

```
assert location.label == 'Tomsk'
```

Ветвление

```
# main program
```

```
if condition:  
    # do something...
```

```
# main program
```

```
# main program
```

```
if condition:  
    # do something...  
else:  
    # do another thing...
```

```
if condition:  
    # ...  
elif:  
    # ...  
elif:  
    # ...  
else:  
    # ...
```

```
# main program
```

Циклы

```
# main program
```

```
while condition:  
    # do something...
```

```
# main program
```

```
# main program
```

```
for x in [1, 2, 3, 4, 5]:  
    # do something for x...
```

```
# main program
```

Ещё раз про списки

```
names = ['Alice', 'Bob', 'Carlos', 'David', 'Eva']
```

Составить список имён в верхнем регистре, убрав слишком короткие имена (от 3 символов и меньше).

Ещё раз про списки

```
names = ['Alice', 'Bob', 'Carlos', 'David', 'Eva']
```

Плохо

```
names_ = []
```

```
for name in names:
    if len(name) > 3:
        names_.append(name.upper())
```

Хорошо

```
names_ = [name.upper() for name in names if len(name) > 3]
```


Функции

Функция в программировании — фрагмент программного кода (подпрограмма), к которому можно обратиться из другого места программы.

```
def _sum(a, b):  
    return a + b  
  
def f(x: float) -> float:  
    return _sum(3, x) * exp(10) + sin(34)
```

```
_f = lambda x: _sum(3, x) * exp(10) + sin(34)
```

```
def _fib(max_value: int):  
    a, b = 0, 1  
  
    while a <= max_value:  
        print(a)  
        a, b = b, a + b
```

Генераторы

```
def _fib_gen():
```

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

```
    yield a
```

```
    yield b
```

```
    while True:
```

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

```
        yield b
```

```
for n in _fib_gen(10):
```

```
    print(n)
```

```
for n in range(10):
```

```
    print(n)
```

```
for n in (gen.__next__() for n in range(10)):
```

```
    print(n)
```

```
names_ = [name.upper() for name in names if len(name) > 3]
```

```
names_ = (name.upper() for name in names if len(name) > 3)
```



Курс-интенсив

Python для быстрого старта в IT

Обзор языка Python

academy.rubius.com

konstantin.dobrychev@rubius.com

Константин Добрычев