

Python Starter

Функции

Python Starter

После урока обязательно



Повторите этот урок в видео формате на [ITVDN.com](http://itvdn.com)



Проверьте как Вы усвоили данный материал на [TestProvider.com](http://testprovider.com)

Python Starter

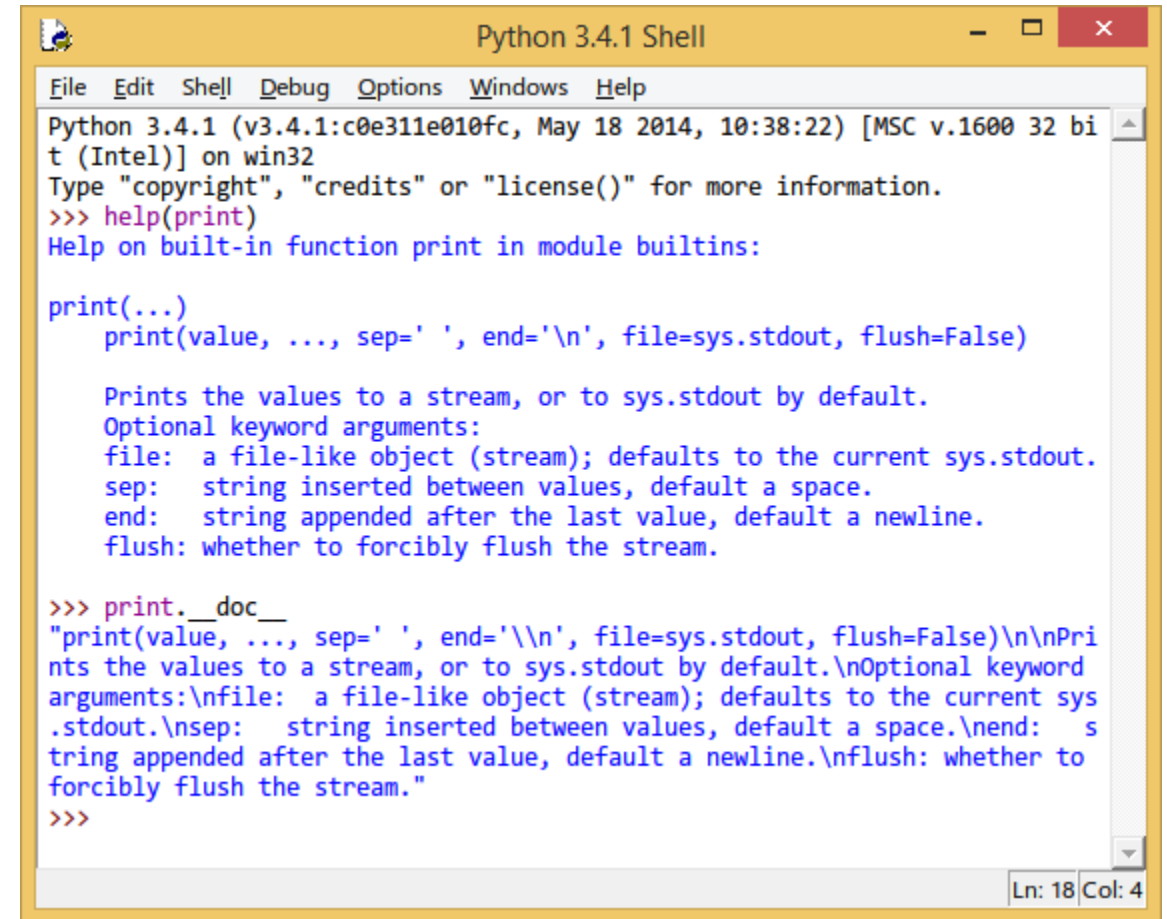
Введение в Python

Функции

Python Starter

Документационные строки

- Строка, стоящая в самом начале функции (а также модуля, класса или метода), играет роль особого вида комментариев – документационной строки (docstring).
- В отличие от обычных комментариев, к документационным строкам можно получить доступ во время выполнения программы.
- Напрямую доступ к документационным строкам осуществляется путём обращения к полю `__doc__` соответствующих объектов.
- При работе с интерпретатором в интерактивном режиме удобно использовать функцию `help`.



```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014, 10:38:22) [MSC v.1600 32 bi
t (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> help(print)
Help on built-in function print in module builtins:

print(...)
    print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

    Prints the values to a stream, or to sys.stdout by default.
    Optional keyword arguments:
    file: a file-like object (stream); defaults to the current sys.stdout.
    sep: string inserted between values, default a space.
    end: string appended after the last value, default a newline.
    flush: whether to forcibly flush the stream.

>>> print.__doc__
"print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)\n\nPri
nts the values to a stream, or to sys.stdout by default.\nOptional keyword
arguments:\nfile: a file-like object (stream); defaults to the current sys
.stdout.\nsep: string inserted between values, default a space.\nend: s
tring appended after the last value, default a newline.\nflush: whether to
forcibly flush the stream."
>>>
```

Python Starter

Стандартные функции

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

<u>abs()</u>	<u>dict()</u>	<u>help()</u>	<u>min()</u>	<u>setattr()</u>
<u>all()</u>	<u>dir()</u>	<u>hex()</u>	<u>next()</u>	<u>slice()</u>
<u>any()</u>	<u>divmod()</u>	<u>id()</u>	<u>object()</u>	<u>sorted()</u>
<u>ascii()</u>	<u>enumerate()</u>	<u>input()</u>	<u>oct()</u>	<u>staticmethod()</u>
<u>bin()</u>	<u>eval()</u>	<u>int()</u>	<u>open()</u>	<u>str()</u>
<u>bool()</u>	<u>exec()</u>	<u>isinstance()</u>	<u>ord()</u>	<u>sum()</u>
<u>bytearray()</u>	<u>filter()</u>	<u>issubclass()</u>	<u>pow()</u>	<u>super()</u>
<u>bytes()</u>	<u>float()</u>	<u>iter()</u>	<u>print()</u>	<u>tuple()</u>
<u>callable()</u>	<u>format()</u>	<u>len()</u>	<u>property()</u>	<u>type()</u>
<u>chr()</u>	<u>frozenset()</u>	<u>list()</u>	<u>range()</u>	<u>vars()</u>
<u>classmethod()</u>	<u>getattr()</u>	<u>locals()</u>	<u>repr()</u>	<u>zip()</u>
<u>compile()</u>	<u>globals()</u>	<u>map()</u>	<u>reversed()</u>	<u>__import__()</u>
<u>complex()</u>	<u>hasattr()</u>	<u>max()</u>	<u>round()</u>	
<u>delattr()</u>	<u>hash()</u>	<u>memoryview()</u>	<u>set()</u>	

Python Starter

Вложенные функции

В Python можно объявлять функции внутри других функций. Они имеют доступ к переменным и аргументам внешней функции и недоступны за пределами тех функций, в которых были определены.

```
def outer_function():  
  
    def inner_function():  
        print('Внутренняя функция')  
  
    print('Внешняя функция')  
    inner_function()
```

Python Starter

Область видимости


- Область видимости (англ. scope) обозначает область программы, в пределах которой идентификатор (имя) некоторой переменной продолжает быть связанным с этой переменной и возвращать её значение. За пределами области видимости тот же самый идентификатор может быть связан с другой переменной, либо быть свободным (не связанным ни с какой из них).
- В языках, поддерживающих структурное программирование, переменные обычно разделяются на два типа по области видимости:
 - глобальные переменные;
 - локальные переменные.

Python Starter

Глобальные переменные

- Глобальные переменные объявляются вне всех функций и доступны отовсюду.
- Использование глобальных переменных имеет недостатки: глобальная переменная может быть изменена в любой точке программы, что может повлиять на работу других частей программы. По этой причине глобальные переменные имеют неограниченный потенциал для создания взаимных зависимостей, что приводит к усложнению программы.

глобальная переменная



```
variable = 42
```

```
def function():  
    pass
```


Python Starter

Локальные переменные

- Локальные переменные объявляются внутри функции и недоступны снаружи неё.
- Переменные локальной области видимости используются, чтобы избежать проблем с побочными эффектами, которые могут произойти с глобальными переменными.
- В Python областью видимости локальной переменной является функция. В некоторых языках любой блок кода может иметь свои локальные переменные.
- Операция присвоения в функции создаёт локальную переменную. Если необходимо изменить значение переменной из другой области видимости, следует воспользоваться операторами `global` или `nonlocal`.

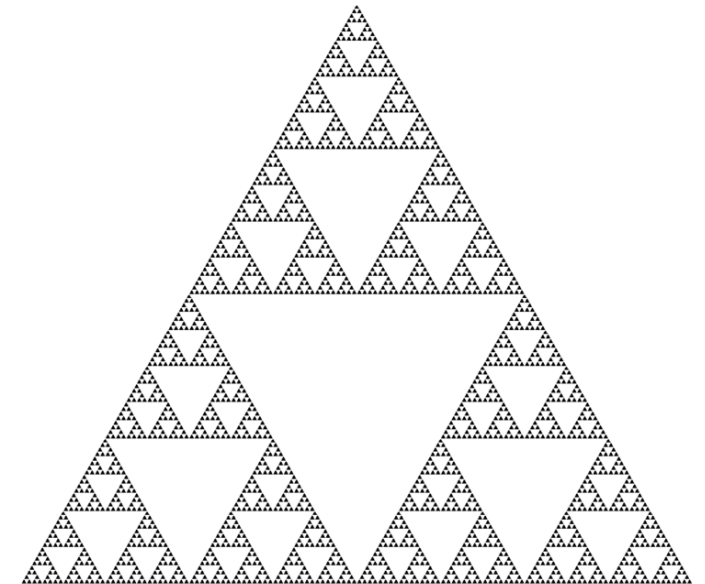
```
def function():  
    variable = 42
```

←
локальная переменная

Python Starter

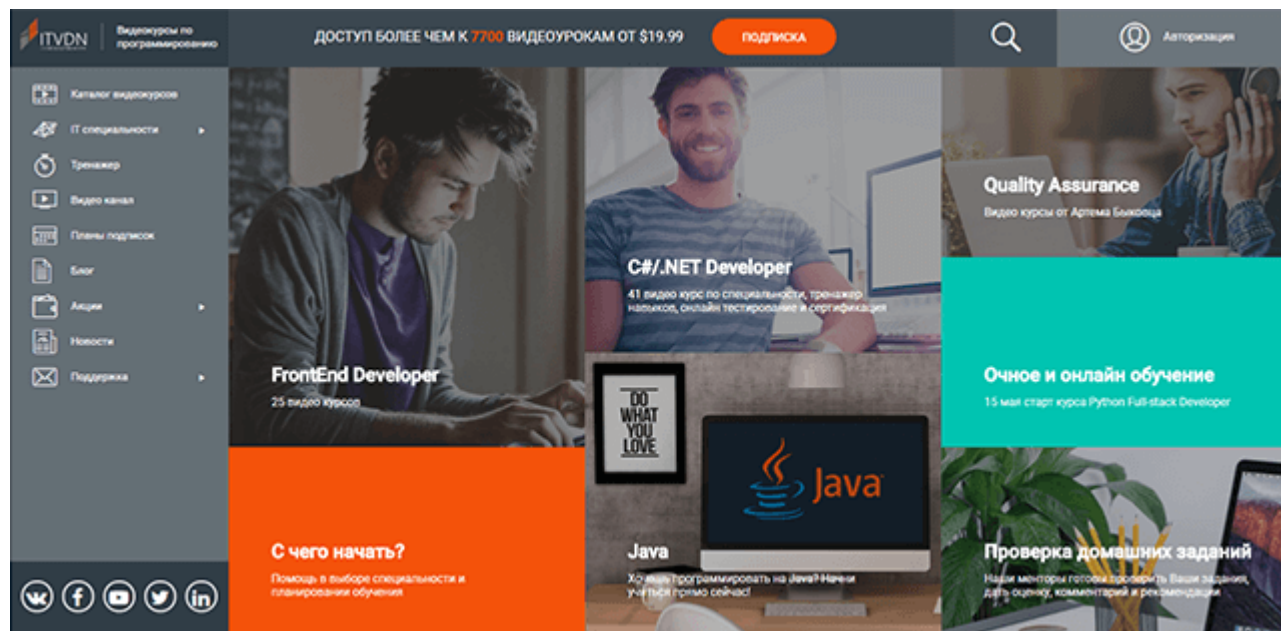
Рекурсия

- Память под локальные переменные выделяется при каждом вызове функции. Это делает возможным рекурсию.
- Рекурсия — вызов функции из неё же самой, непосредственно (простая рекурсия) или через другие функции (сложная или косвенная рекурсия), например, функция А вызывает функцию В, а функция В — функцию А.
- Количество вложенных вызовов функции или процедуры называется глубиной рекурсии.
- Рекурсивная программа позволяет описать повторяющееся или даже потенциально бесконечное вычисление, причём без явных повторений частей программы и использования циклов.



Смотрите наши уроки в видео формате

ITVDN.com



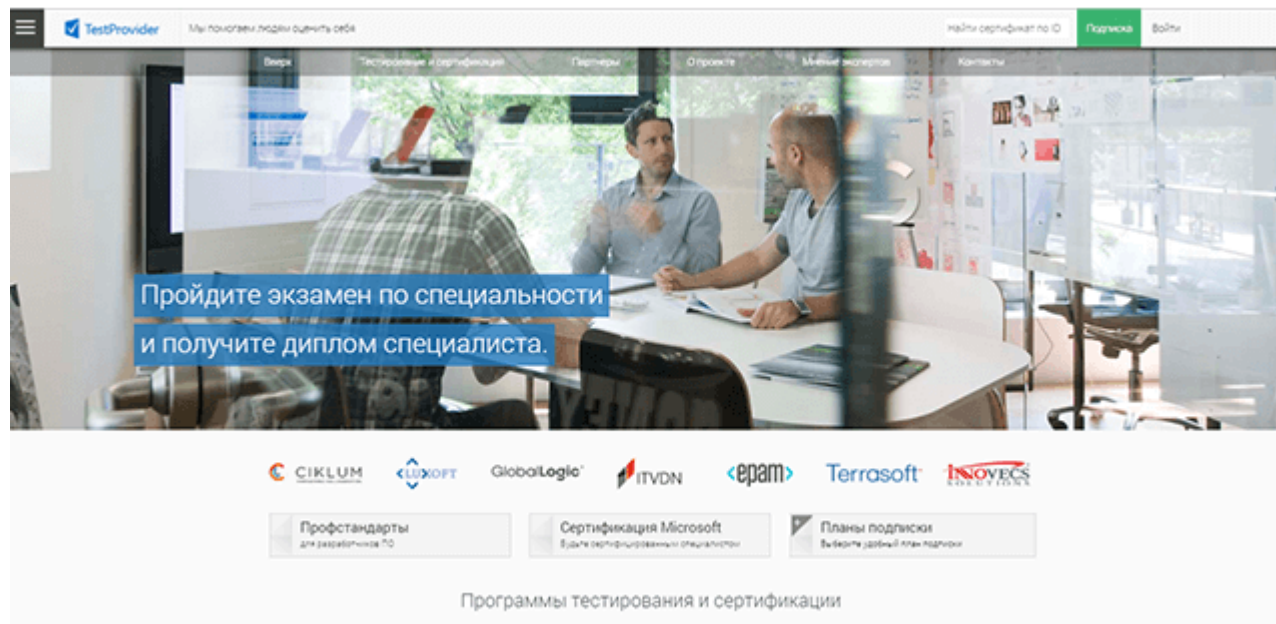
Посмотрите этот урок в видео формате на образовательном портале ITVDN.com для закрепления пройденного материала.

Курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics и другими высококвалифицированными разработчиками.



Проверка знаний

TestProvider.com



TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и для общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на [TestProvider.com](https://testprovider.com)

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.



Python Starter

Q&A

Информационный видеосервис для разработчиков программного обеспечения

