

# Python Starter

Переменные и типы данных

# Python Starter

После урока обязательно



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

Доступ можно получить через руководство вашего учебного центра



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

# Python Starter

Введение в Python

## Переменные и типы данных

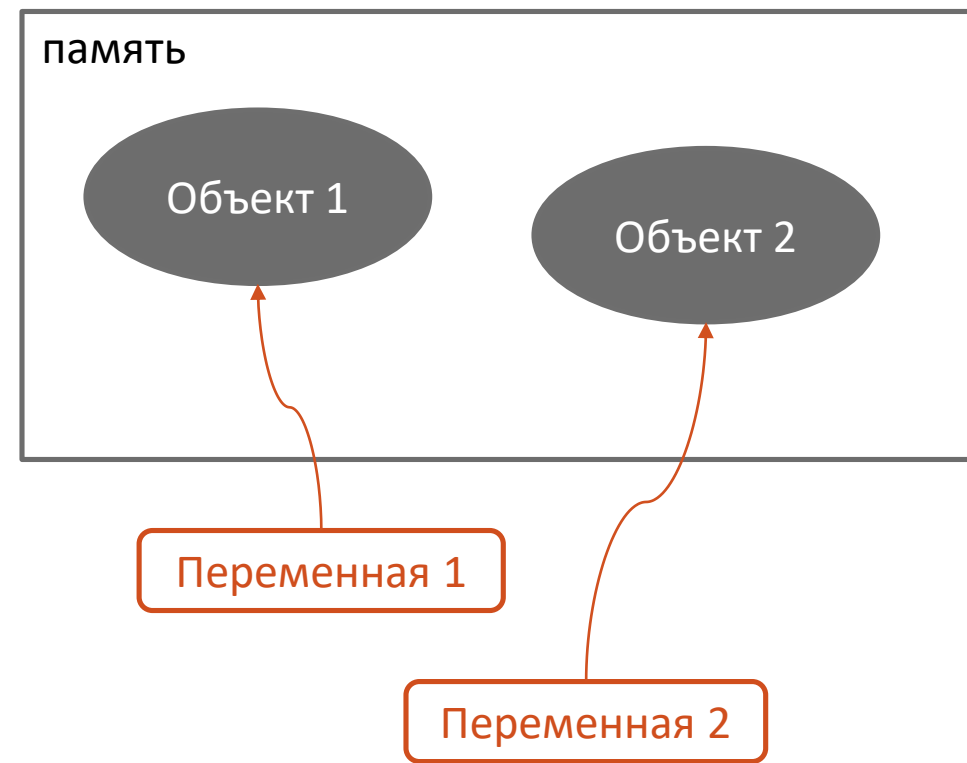
# Python Starter

## Переменные и константы

**Идентификатор** – это имя объекта. Он может состоять из больших и маленьких букв, цифр, знаков подчёркивания, не должен начинаться с цифры и не может совпадать с зарезервированными ключевыми словами языка. Регистр имеет значение.

*Переменная* в Python – ссылка на какой-либо объект. Согласно PEP 8, имена переменных должны записываться маленькими буквами через знаки подчёркивания: `my_variable`.

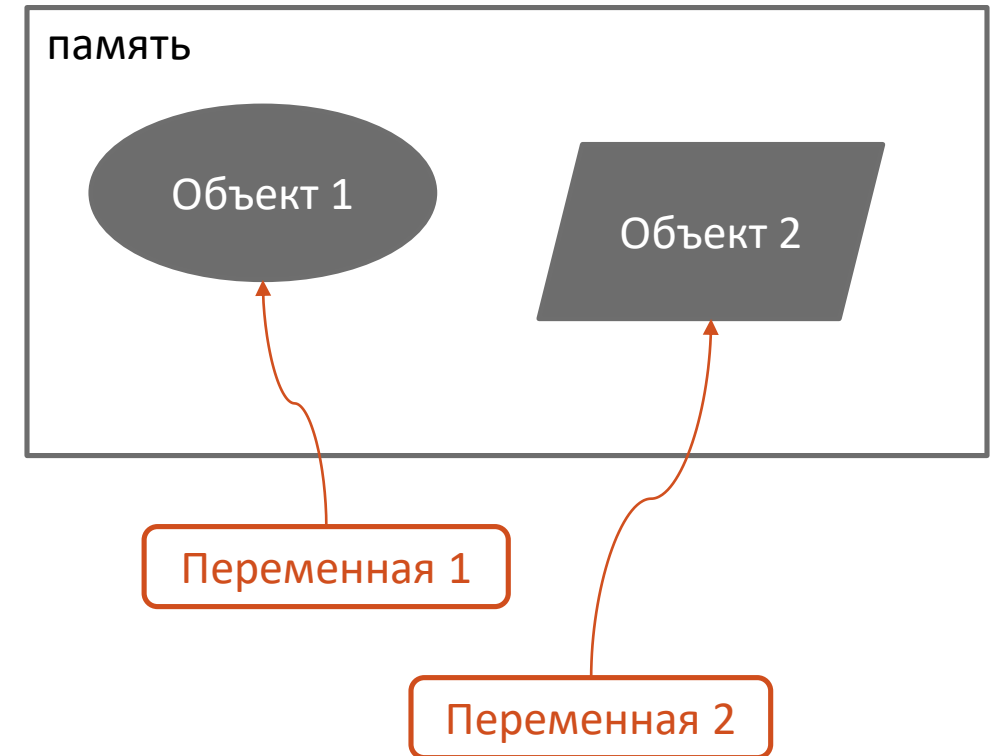
*Константа* – значение, которое не предназначено для изменения. В Python нет отдельной синтаксической конструкции для объявления констант. Принято называть их идентификаторами, написанными заглавными буквами: `MY_CONSTANT`.



# Python Starter

## Понятие типов данных

- *Тип данных* (тип) — множество значений и операций на этих значениях.
- Тип определяет возможные значения и их смысл, операции, а также способы хранения значений типа. Изучается теорией типов. Неотъемлемой частью большинства языков программирования являются системы типов, использующие типы для обеспечения той или иной степени типобезопасности.
- Операция назначения типа информационным сущностям называется *типизацией*.



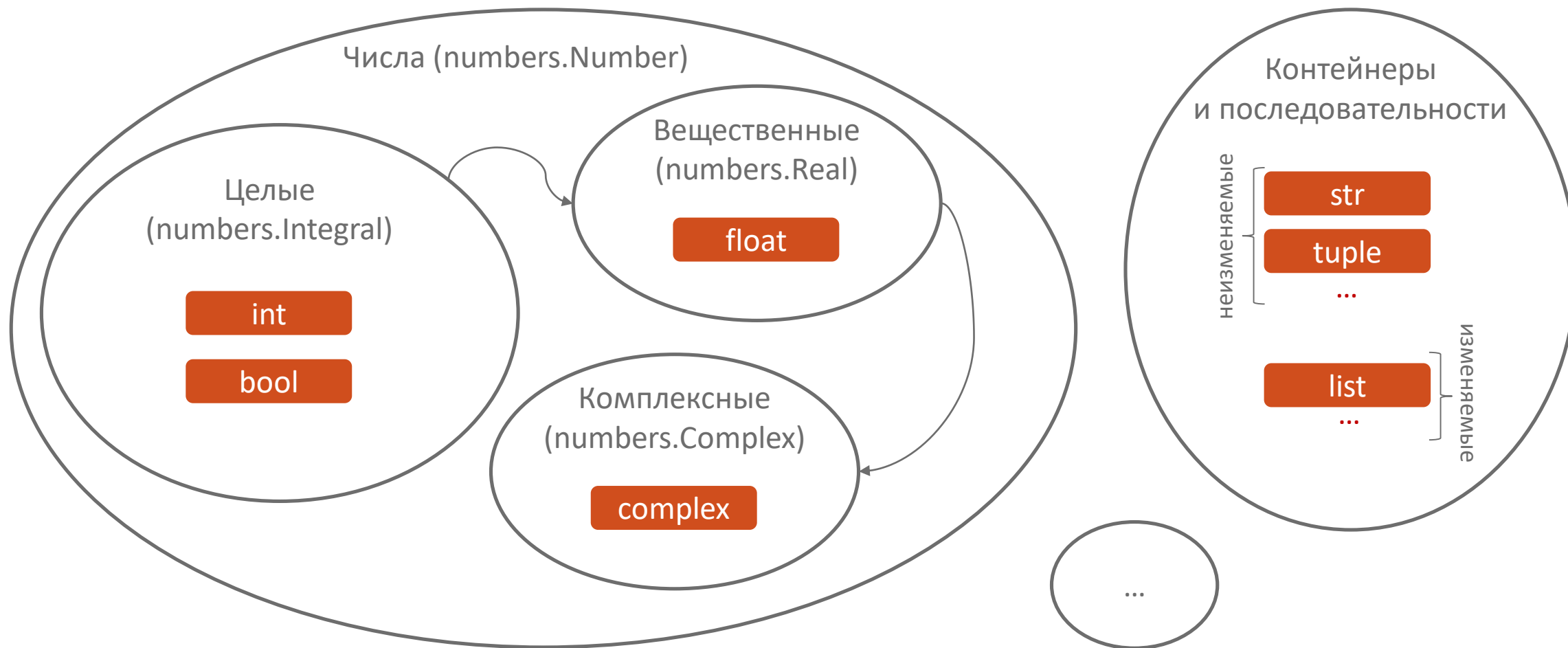
# Python Starter

## Типизация

Типизация	Статическая	Динамическая
Сильная (строгая)	C#, Java, Haskell, Scala	<b>Python</b> , Ruby
Слабая	C	JavaScript, PHP

# Python Starter

## Основные стандартные типы данных



# Python Starter

Язык программирования Python

Операции с числами



# Python Starter

## Операции с числами

$x + y$	сумма
$x - y$	разница
$x * y$	произведение
$x / y$	частное
$x // y$	операция целочисленного деления
$x \% y$	остаток от деления
$-x$	число, противоположное $x$
$+x$	$x$

# Python Starter

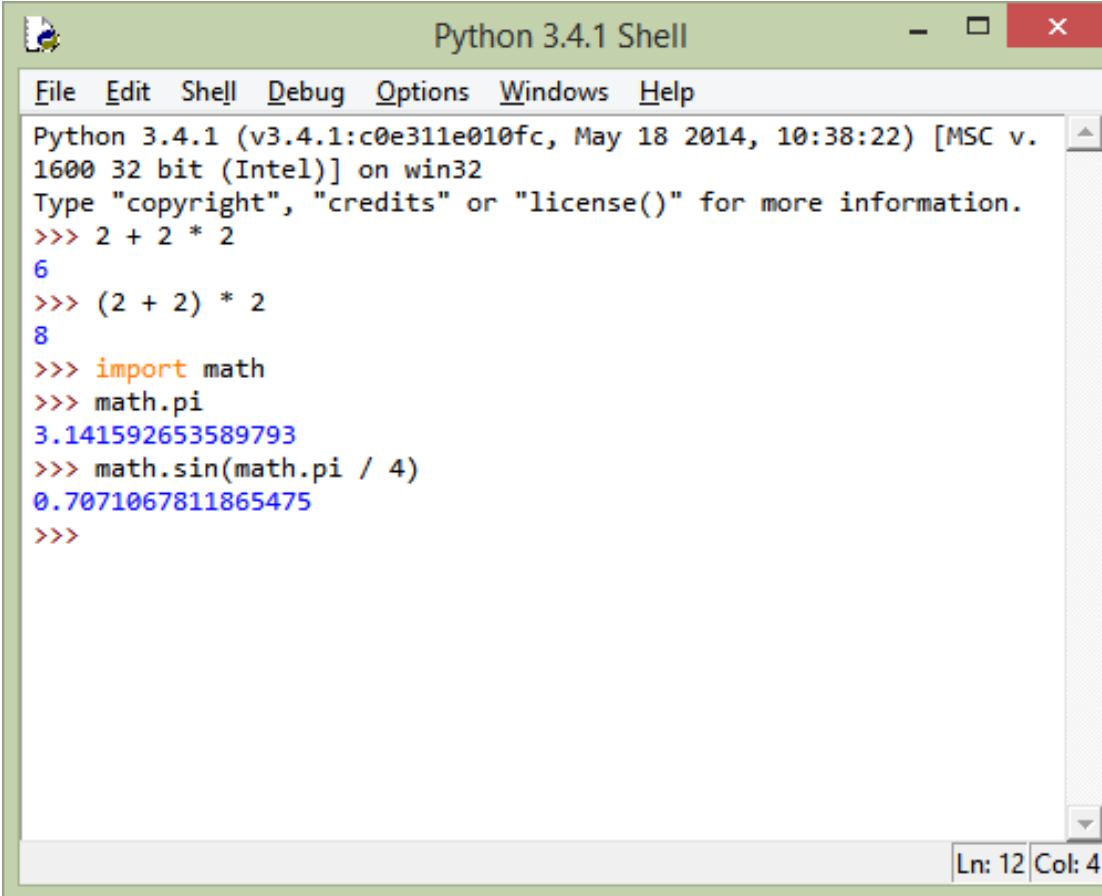
## Операции с числами

<b>abs(x)</b>	модуль числа x
<b>int(x)</b>	преобразовать x в целое число
<b>float(x)</b>	преобразовать x в действительное число
<b>complex(re, im)</b>	создать комплексное число $re + im*i$
<b>c.conjugate()</b>	число, сопряжённое комплексному числу c
<b>x ** y</b> <b>pow(x, y)</b>	x в степени y
<b>round(x)</b> <b>round(x, n)</b>	округлить действительное число x (до n цифр после запятой, если n указано)

# Python Starter

## Операции с числами

- У арифметических операций есть приоритеты (такие же, как и в математике). Чтобы изменить порядок вычислений, следует использовать круглые скобки.
- Можно импортировать модуль `math`, написав `import math`, и использовать его функции и константы.



```
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 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 2 + 2 * 2
6
>>> (2 + 2) * 2
8
>>> import math
>>> math.pi
3.141592653589793
>>> math.sin(math.pi / 4)
0.7071067811865475
>>>
```

Ln: 12 Col: 4

# Python Starter

## Некоторые функции и константы модуля math

<code>math.trunc(x)</code>	отбросить дробную часть действительного числа $x$ , возвращает целое число
<code>math.floor(x)</code>	наибольшее целое число (целое в математическом смысле, а не как тип данных), которое не превосходит данное вещественное число
<code>math.ceil(x)</code>	наименьшее целое число, большее или равное данному вещественному
<code>math.pi</code> , <code>math.e</code>	константы $\pi$ , $e$
<code>math.sin</code> , <code>math.cos</code> и т.д.	математические функции; полный их список можно посмотреть в документации ( <a href="https://docs.python.org/3/library/math.html">https://docs.python.org/3/library/math.html</a> ) или набрав в консоли интерпретатора: <pre>import math dir(math)</pre>

# Python Starter

## Введение в Python

### Логические операции и сравнения

# Python Starter

## Логические операции

- *Логическая операция* — операция над выражениями логического (булевского) типа, соответствующая некоторой операции над высказываниями в алгебре логики. Как и высказывания, логические выражения могут принимать одно из двух истинностных значений — «истинно» или «ложно».
- Логические операции служат для получения сложных логических выражений из более простых. В свою очередь, логические выражения обычно используются как условия для управления последовательностью выполнения программы.

# Python Starter

## Таблицы истинности

x	y	x and y
False	False	False
False	True	False
True	False	False
True	True	True

x	y	x or y
False	False	False
False	True	True
True	False	True
True	True	True

x	not x
False	True
True	False

# Python Starter

## Логические операции в Python

<b>x or y</b>	если x – ложь, то y, иначе x
<b>x and y</b>	если x – ложь, то x, иначе y
<b>not x</b>	если x – ложь, то True, иначе False



# Python Starter

## Операции сравнения

<code>x &lt; y</code>	x строго меньше y
<code>x &gt; y</code>	x строго больше y
<code>x &lt;= y</code>	x меньше или равен y
<code>x &gt;= y</code>	x больше или равен y
<code>x == y</code>	x равен y
<code>x != y</code>	x не равен y
<code>x is y</code>	x и y – это один и тот же объект
<code>x is not y</code>	x и y не являются одним и тем же объектом в памяти

С операциями `<`, `>`, `<=`, `>=` можно использовать двойные сравнения, например, `-2 <= x < 3`.

# Python Starter

Введение в Python

## Работа со строками

# Python Starter

## Строки

- *Строки* – текстовые данные типа `str`.
- Строковые литералы (значения, которые задаются в коде) обрамляются с обеих сторон одинарными или двойными кавычками.
- Несколько строковых литералов подряд распознаются как одна строка.
- Строки, которые выделяются одной парой кавычек, должны располагаться на одной строке кода. Если необходимо создать строку из нескольких строчек текста, можно:
  - использовать специальную последовательность символов `\n` ;
  - расположить несколько строковых литералов на разных строках кода, не добавляя между ними никаких операций;
  - использовать специальный вид строковых литералов, который обрамляется тремя парами одинарных или двойных кавычек. Любая часть исходного текста программы, располагающаяся между ними, включая переводы строк и отступы, считается строкой.

# Python Starter

## Некоторые операции со строками

<code>s1 + s2</code>	конкатенация (объединение) строк
<code>s % x</code> <code>s % (x<sub>1</sub>, x<sub>2</sub>, ..., x<sub>n</sub>)</code>	форматирование строки в стиле C s – форматная строка, x <sub>1</sub> ...x <sub>n</sub> – значения <a href="https://docs.python.org/3/library/stdtypes.html#printf-style-string-formatting">https://docs.python.org/3/library/stdtypes.html#printf-style-string-formatting</a>
<code>s.format(args)</code>	форматирование строки в стиле C# <a href="https://docs.python.org/3/library/stdtypes.html#str.format">https://docs.python.org/3/library/stdtypes.html#str.format</a>
<code>s[i]</code> (вернуться в седьмом уроке)	символ, который стоит в строке s на позиции i (нумерация начинается с нуля)

# Python Starter

## Вывод

Для вывода значений на экран служит функция print:

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

где

- objects – это объекты, которые необходимо вывести
- sep – разделитель
- end – строка, которую необходимо вывести после всех объектов
- file – файл, в который необходимо вывести данные
- flush – нужно ли сразу после вывода сбросить содержимое буфера в файл. Если вы выводите информацию на одной строке через продолжительные промежутки времени, и она не появляется на экране, пока вы не выведете символ новой строки, добавьте параметр flush=True

Ни один из этих параметров не является обязательным.

# Python Starter

## Ввод

- Для ввода данных с клавиатуры можно использовать функцию `input`:

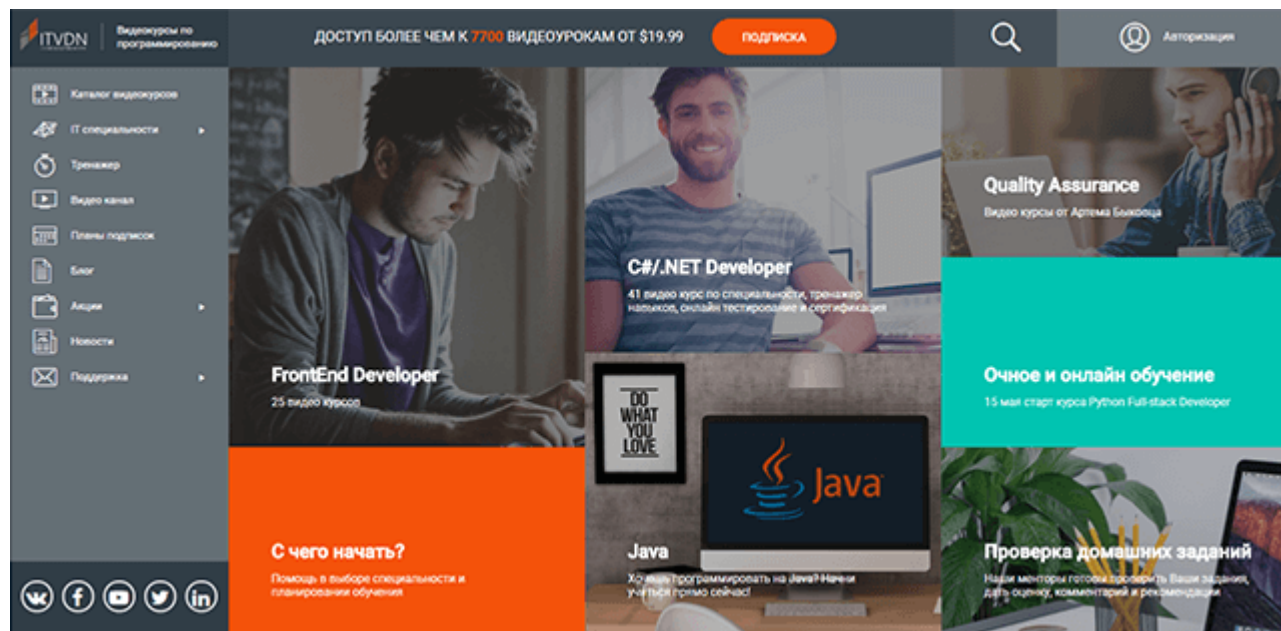
`input(prompt)`

`input()`

- Если параметр `prompt` задан, то он выводится, как поясняющий текст, приглашение ко вводу.
- Функция приостанавливает выполнение программы, пока пользователь не введёт строку текста, считывает её и возвращает.
- Обратите внимание, что она возвращает именно строку, то есть значение типа `str`, поэтому если необходимо ввести число, то нужно воспользоваться одной из функций, которые были рассмотрены выше, чтобы сконструировать число из его текстового представления.

# Смотрите наши уроки в видео формате

ITVDN.com



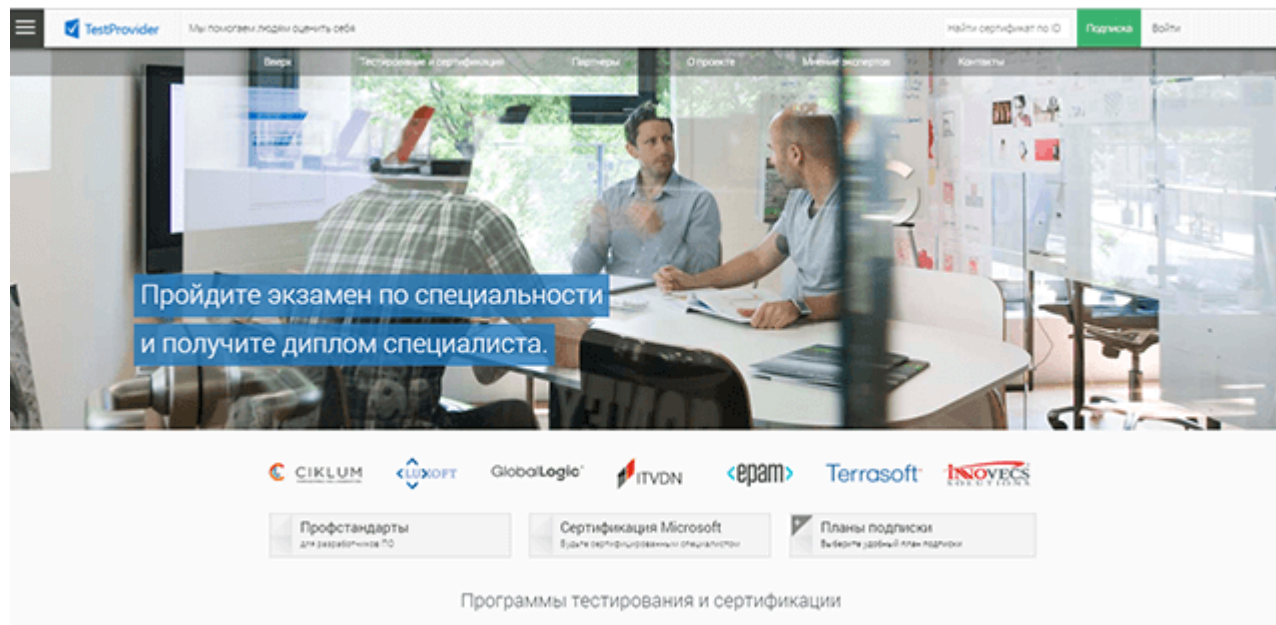
Посмотрите этот урок в видео формате на образовательном портале [ITVDN.com](http://ITVDN.com) для закрепления пройденного материала.

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

