

Python Starter

Условные конструкции

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

Основы алгоритмизации.
Понятие условных конструкций

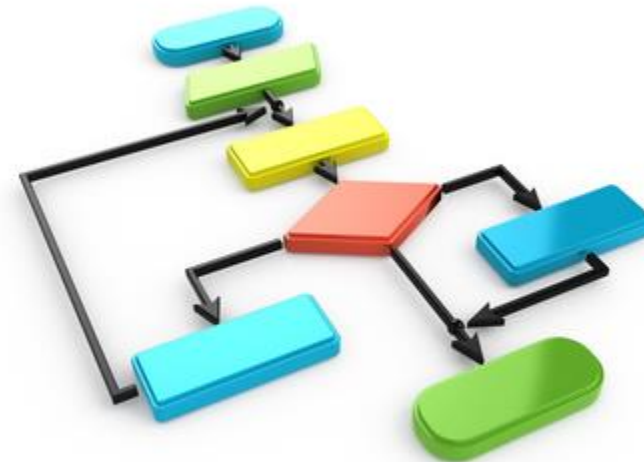
Python Starter

Понятие алгоритма

Алгоритм — набор инструкций, описывающих порядок действий исполнителя для достижения результата решения задачи за конечное число действий.

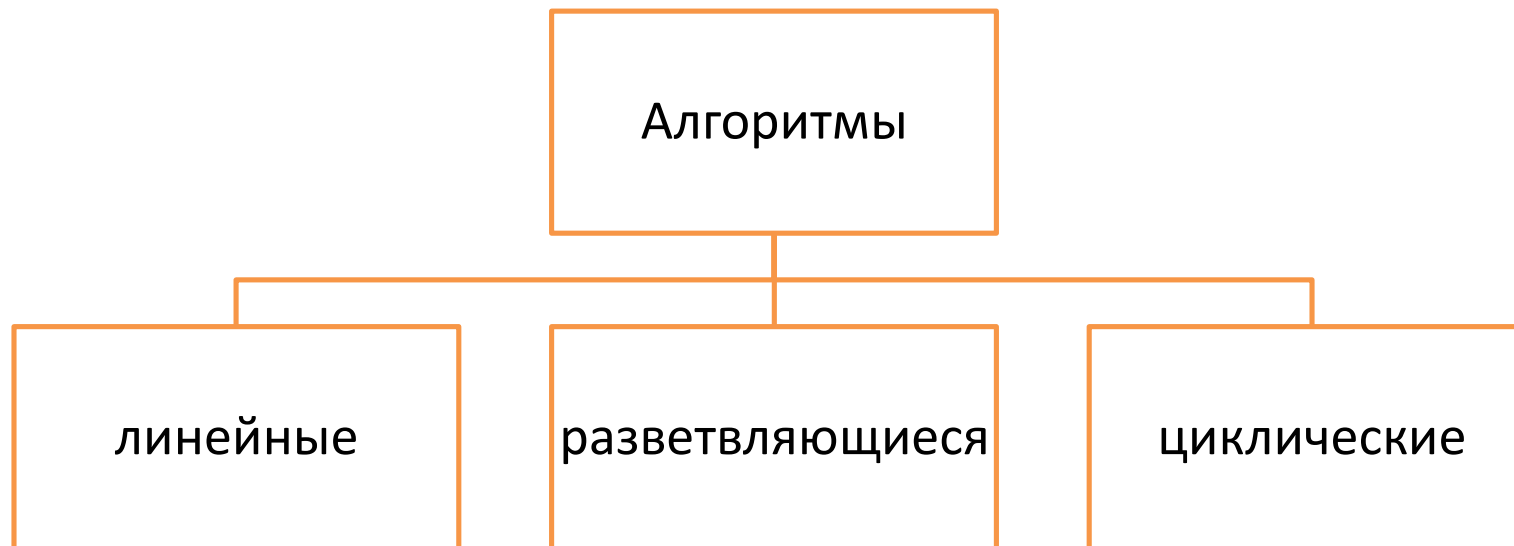
Свойства:

- дискретность
- детерминированность
- понятность
- завершаемость (конечность)
- массовость (универсальность)
- результативность



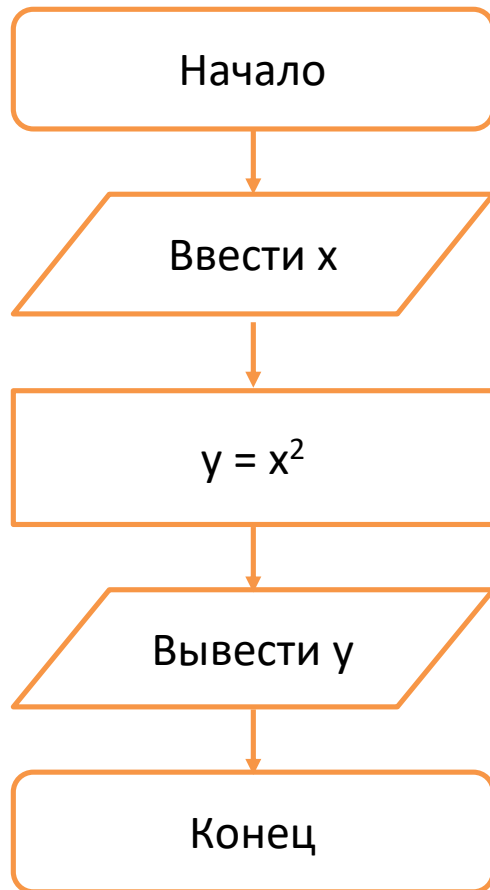
Python Starter

Основные алгоритмические конструкции



Python Starter

Линейные алгоритмы

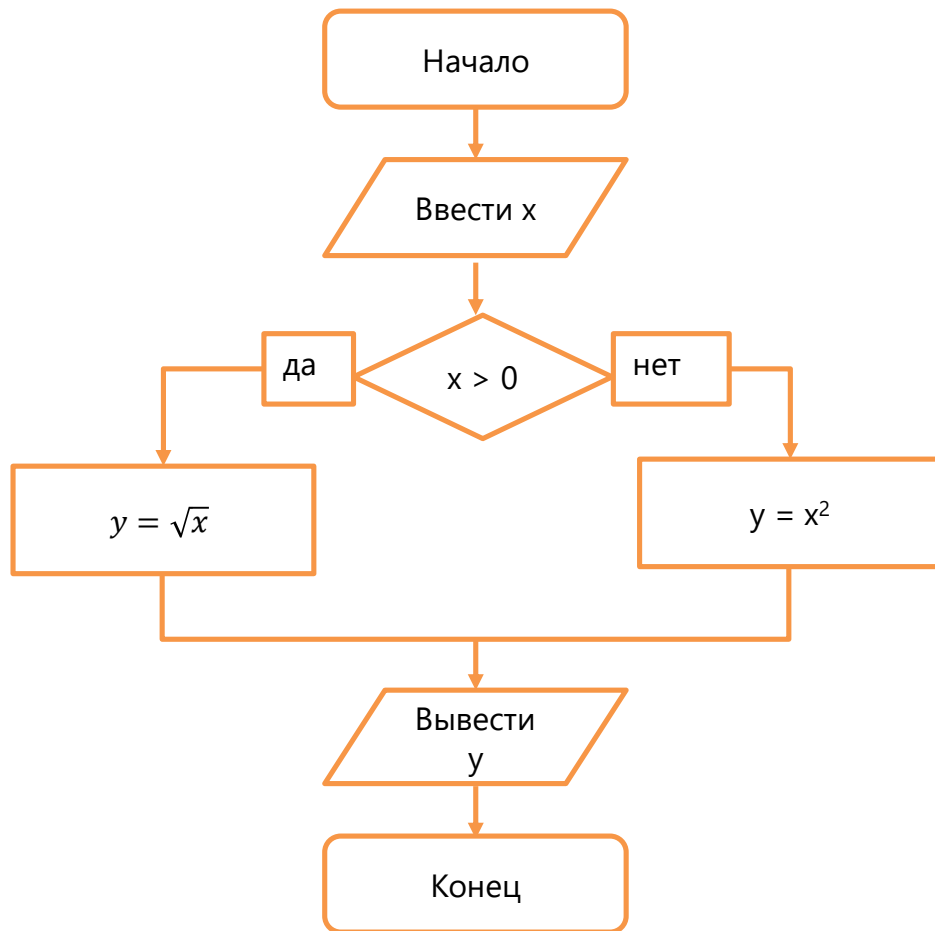


Линейный алгоритм — набор команд (указаний), выполняемых последовательно во времени друг за другом.

Именно алгоритмы такого типа мы реализовывали на прошлом уроке.

Python Starter

Алгоритмы с ветвлениями



Разветвляющийся алгоритм — алгоритм, содержащий хотя бы одно условие, в результате проверки которого может осуществляться разделение на несколько параллельных ветвей алгоритма.

То есть, разветвляющийся алгоритм содержит условия, в зависимости от которых выполняется одна или другая последовательность действий. Каждая из них, в свою очередь, также может (но не обязана) содержать ветвления.

Python Starter

Введение в Python

Оператор ветвления

Python Starter

Оператор if

if *условие*:
операторы

Здесь:

- *условие* – это логическое выражение,
- *операторы* – это последовательность каких-либо других команд.

Блок операторов не может быть пустым. Если такая необходимость возникает, можно воспользоваться оператором *pass*, который не делает ничего.

Однострочная форма (нежелательна для использования):

if *условие*: операторы

Здесь операторы, если их несколько, разделяются символом точки с запятой.

Python Starter

Оператор if-else

```
if условие:  
    блок_операторов_1  
else:  
    блок_операторов_2
```

Здесь, если условие истинно, выполняются операторы из первого блока, иначе из второго.

Python Starter

Вложенность операторов

Так как блоки операторов внутри if могут содержать любые операторы и их последовательности, операторы if могут быть вложенными. Пример:

```
if 0 < x < 7:
    print('Значение x входит в заданный диапазон, продолжаем')
    y = 2 * x - 5
    if y < 0:
        print('Значение y отрицательно')
    else:
        if y > 0:
            print('Значение y положительно')
        else:
            print('y = 0')
```

Python Starter

Введение в Python

Оператор ветвления с несколькими условиями

Python Starter

Каскадирование операторов ветвления

```
if условие1:
    команды1
else:
    if условие2:
        команды2
    else:
        if условие3:
            команды3
        ...
        else:
            if условиеN:
                командыN
            else:
                командыM
```

Часто возникает необходимость поочерёдно проверить несколько условий (чаще всего – равенство какого-то выражения определённым значениям).

Однако огромное количество (каскад) вложенных условных операторов выглядит нечитабельно и при написании подобного кода легко допустить ошибку.

Варианты решения этой проблемы в разных языках программирования:

- оператор переключения;
- механизм сопоставления с образцом;
- оператор ветвления с несколькими условиями.

Python Starter

Оператор ветвления с несколькими условиями

```
if условие1:  
    операторы1  
elif условие2:  
    операторы2  
elif условие3:  
    операторы3  
...  
else:  
    операторыN
```

Как и в операторе ветвления с одним условием, ветка else не является обязательной.

Python Starter

Введение в Python

Условное выражение

Python Starter

Условное выражение

- Довольно часто возникает ситуация, когда нам нужно использовать то или иное выражение в зависимости от определённого условия. Поэтому существует такая конструкция, как условное выражение (или, иначе, «тернарный оператор»).
- Синтаксис условного выражения в Python:
выражение1 **if** условие **else** выражение2
- Результатом всего этого выражения является значение первого выражения, если условие истинно, или второго, если ложно.

Python Starter

Введение в Python

Логические значения выражений, которые
не являются булевыми

Python Starter

Логические значения объектов

- Любой объект в Python может быть рассмотрен как логическое значение для использования в условии оператора `if`, конструирования значения типа `bool` или использования как операнда логических операций.
- Изученные в предыдущем уроке типы (`NoneType`, `bool`, `int`, `float`, `complex`, `str`) отображаются на логические значения так: `None`, `False`, `0`, `0.0`, `0j` и `''` (пустая строка) считаются ложными значениями, все остальные – истинными.

- Пример. Вместо

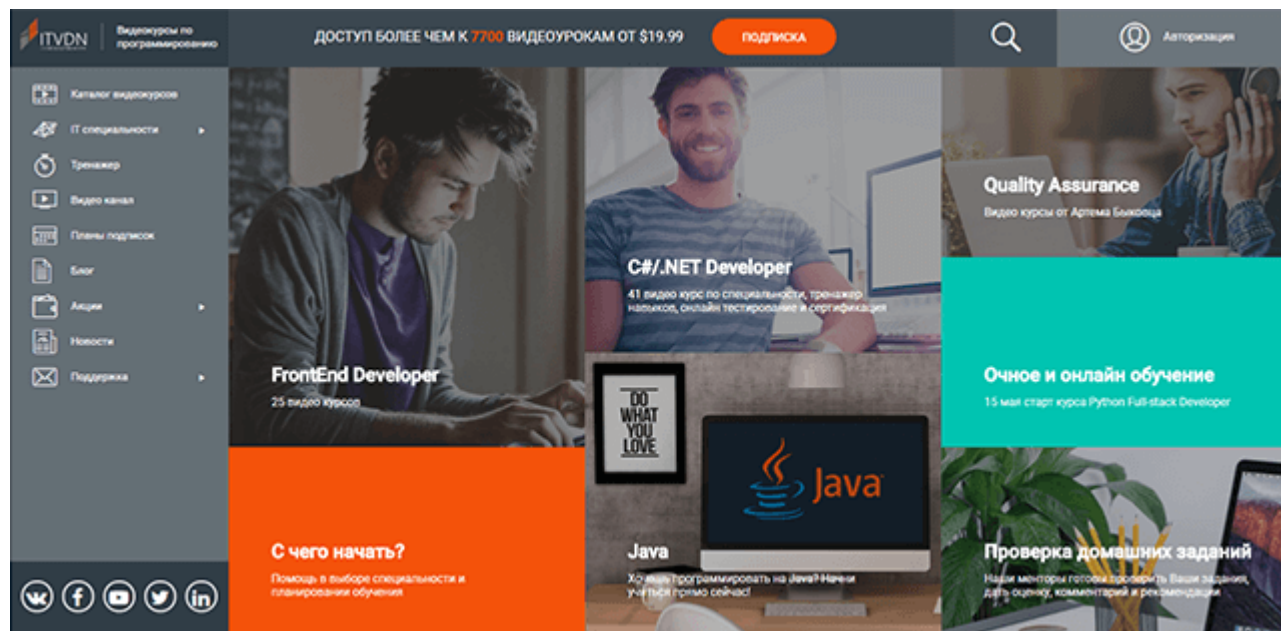
```
if string is not None and string != '':  
    num = int(string)
```

можно записать

```
if string:  
    num = int(string)
```

Смотрите наши уроки в видео формате

ITVDN.com



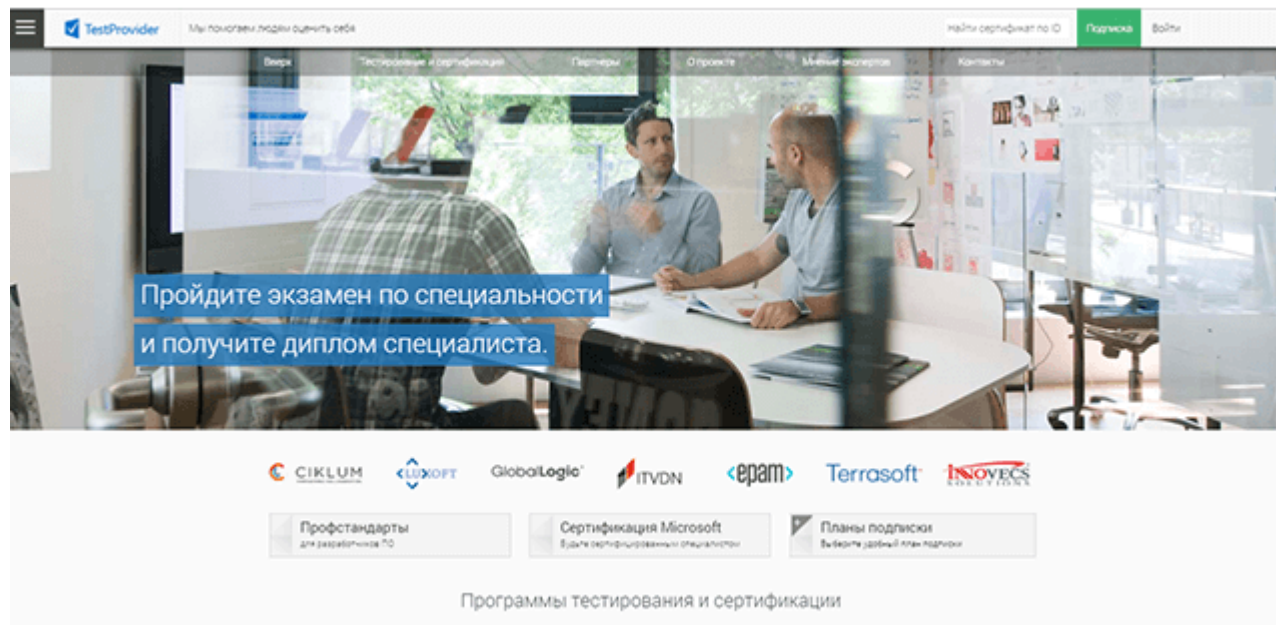
Посмотрите этот урок в видео формате на образовательном портале ITVDN.com для закрепления пройденного материала.

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

