

Python Starter

Умовні конструкції

Python Starter

План заняття

1. Поняття умовних конструкцій
2. Оператор if-else
3. Оператор if-elif-else
4. Тернарні оператори
5. Логічні значення не-булевих виразів
6. Конструкція match/case

Python Starter

Після уроку обов'язково



Повторіть цей урок у відео форматі на [ITVDN.com](http://itvdn.com)

Доступ можна отримати через керівництво вашого навчального центру



Перевірте, як Ви засвоїли цей матеріал на [TestProvider.com](http://testprovider.com)

Основи алгоритмізації. Поняття умовних конструкцій

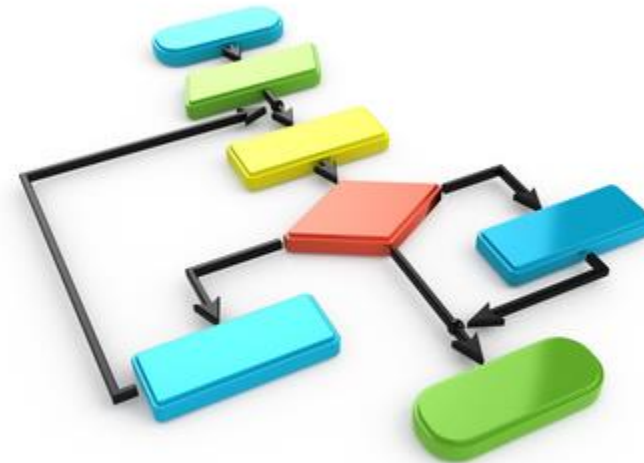
Python Starter

Поняття алгоритму

Алгоритм — набір інструкцій, що описують порядок дій виконавця для досягнення результату розв'язання задачі за кінцеве число дій.

Властивості:

- дискретність
- детермінованість
- зрозумілість
- завершуваність (кінцевість)
- масовість (універсальність)
- результативність



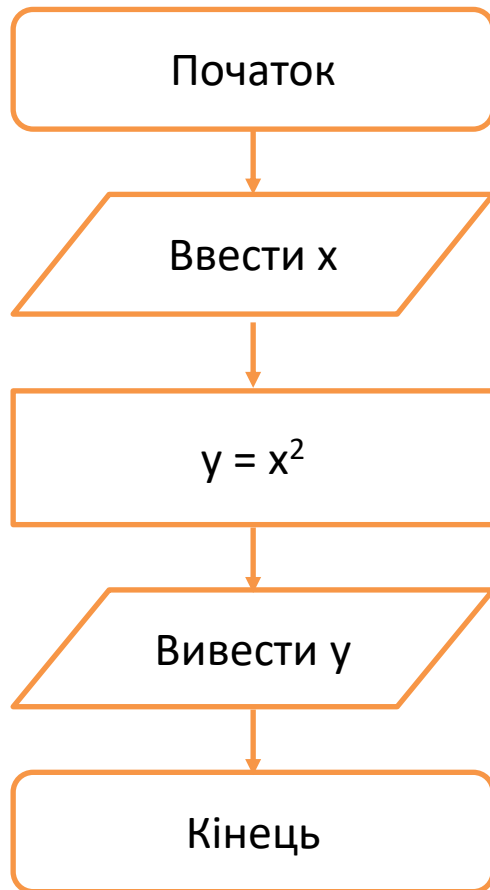
Python Starter

Основні алгоритмічні конструкції



Python Starter

Лінійні алгоритми

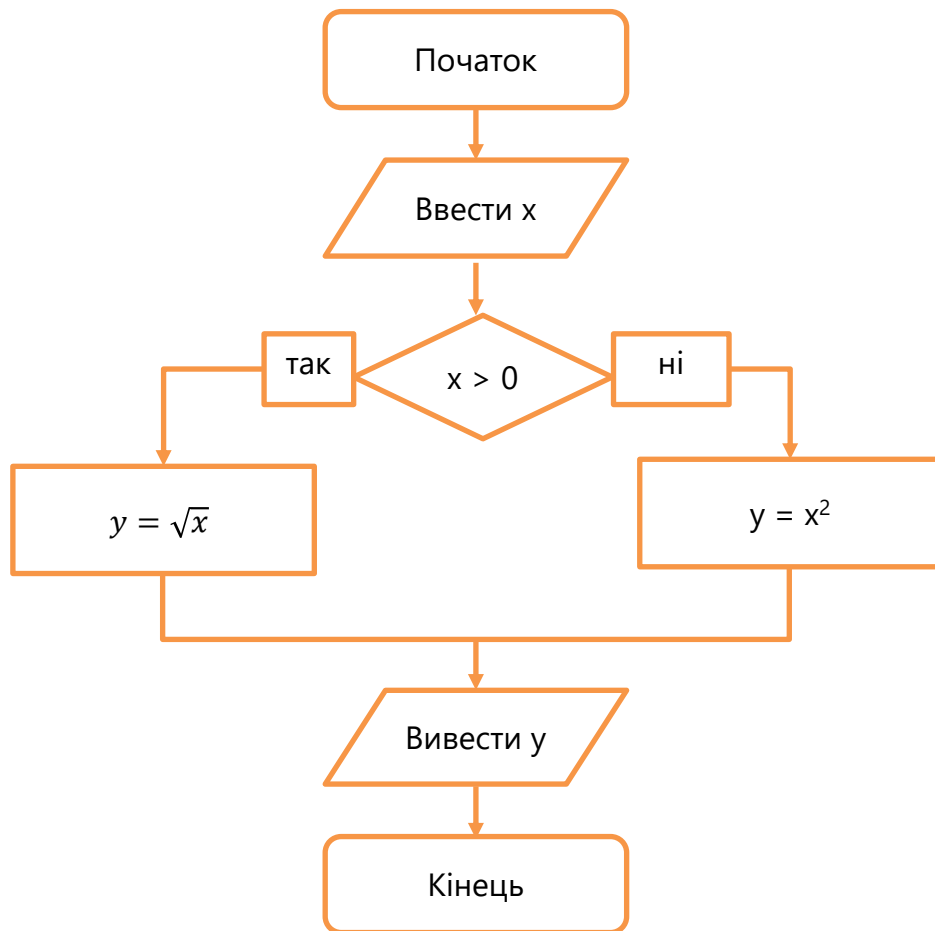


Лінійний алгоритм — набір команд (вказівок), що виконуються послідовно в часі одна за одною.

Саме алгоритми такого типу ми реалізовували на минулому уроці.

Python Starter

Алгоритми з розгалуженнями



Алгоритм, що розгалужується — алгоритм, що містить хоча б одну умову, в результаті перевірки якої може здійснюватися поділ на кілька паралельних гілок алгоритму.

Тобто, алгоритм, що розгалужується, містить умови, в залежності від яких виконується одна або інша послідовність дій. Кожна з них, у свою чергу, також може (але не зобов'язана) мати розгалуження.

Python Starter

Введення в Python

Оператор розгалуження

Python Starter

Оператор if

if *умова*:
оператори

Тут:

- *умова* – це логічний вираз,
- *оператори* – це послідовність будь-яких інших команд.

Блок операторів не може бути порожнім. Якщо така необхідність виникає, можна скористатися оператором `pass`, який нічого не робить.

Однорядкова форма (небажана для використання):

if *умова*: оператори

Тут оператори, якщо їх декілька, розділяються символом крапки з комою.

Python Starter

Оператор if-else

```
if умова:  
    блок_операторів_1  
else:  
    блок_операторів_2
```

Тут, якщо умова є істинною, виконуються оператори з першого блоку, інакше з другого.

Python Starter

Вкладеність операторів

Оскільки блоки операторів всередині if можуть містити будь-які оператори та їх послідовності, оператори if можуть бути вкладеними. Приклад :

```
if 0 < x < 7:
    print('Значення x входить до заданого діапазону, продовжуємо')
    y = 2 * x - 5
    if y < 0:
        print('Значення y негативне')
    else:
        if y > 0:
            print('Значення y позитивне')
        else:
            print('y = 0')
```

Python Starter

Введення в Python

Оператор розгалуження з кількома
умовами

Python Starter

Каскадування операторів розгалуження

```
if умова1:
    команди1
else:
    if умова2:
        команди2
    else:
        if умова3:
            команди3
        ...
        else:
            if умоваN:
                командиN
            else:
                командиM
```

Часто виникає необхідність по черзі перевірити кілька умов (найчастіше – рівність якогось виразу певним значенням).

Однак величезна кількість (каскад) вкладених умовних операторів виглядає нечитабельною і при написанні такого коду легко припуститися помилки.

Варіанти вирішення цієї проблеми у різних мовах програмування :

- оператор перемикання;
- механізм співставлення зі зразком;
- оператор розгалуження з кількома умовами.

Python Starter

Оператор розгалуження з кількома умовами

```
if умова1:  
    оператори1  
elif умова2:  
    оператори2  
elif умова3:  
    оператори3  
...  
else:  
    операториN
```

Як і в операторі розгалуження з однією умовою, гілка else не є обов'язковою.

Python Starter

Введения в Python

Умовний вираз
(Тернарний оператор)

Python Starter

Тернарний оператор

- Досить часто виникає ситуація, коли нам потрібно використовувати той чи інший вираз в залежності від певної умови. У Python існує конструкція, яка за своєю дією аналогічна конструкції if-else, але при цьому є умовним виразом (або, інакше, «**тернарний оператор**»).
- Синтаксис умовного виразу в Python:
`вираз1 if умова else вираз2`
- Результатом всього цього виразу є значення першого виразу, якщо умова **істинна**, або **другого**, якщо хибна.

Python Starter

Тернарний оператор

Тернарний оператор — єдиний у своєму роді оператор, що потребує три операнди:
<expression on true> if <predicate> else <expression on false>.

```
number = int(input("Input your digit: "))  
if number >= 0:  
    print(number)  
else:  
    print(-number)
```

Чи можна записати більш лаконічно?

```
print(number if number >= 0 else -number)
```

Python Starter

Введення в Python

Логічні значення виразів, які не є булевими

Python Starter

Логічні значення об'єктів

- Будь-який об'єкт у Python може бути розглянутий як логічне значення для використання в умові оператора if, конструюванні значення типу bool або використання як операнда логічних операцій.
- Вивчені на попередньому уроці типи (NoneType, bool, int, float, complex, str) відображаються на логічні значення так: None, False, 0, 0.0, 0j і "" (порожній рядок) вважаються хибними значеннями, решта – істинними.
- Приклад. Замість

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

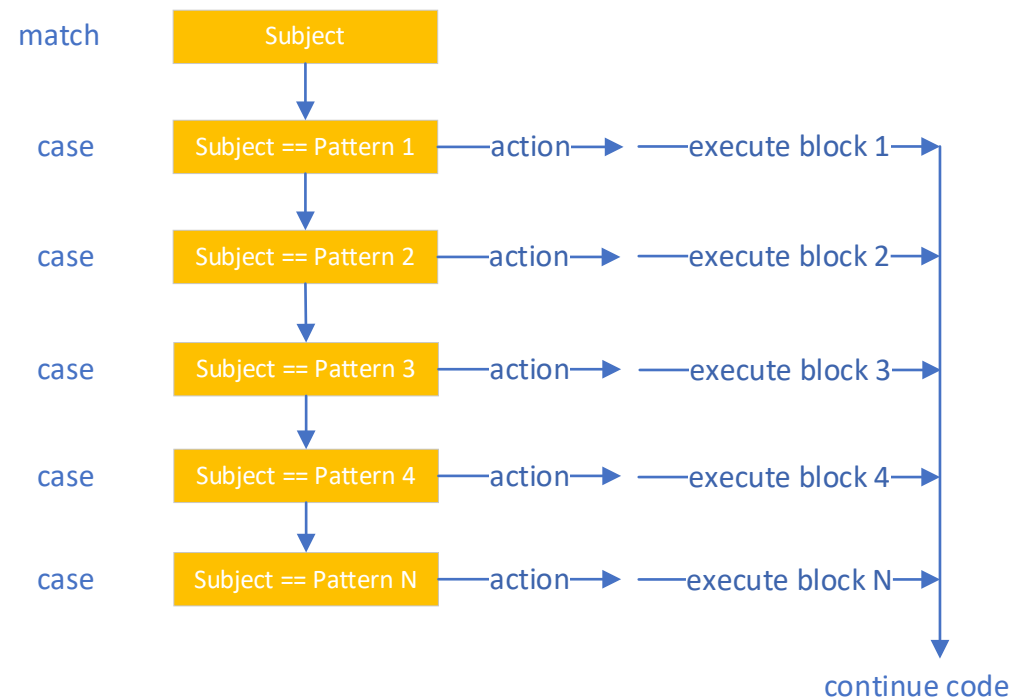
можна записати

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

Python Starter

Довгоочікувані інструкції match/case в Python

```
match variable_name:  
    case 'pattern1' : //statement1  
    case 'pattern2' : //statement2  
    ...  
    case 'pattern n' : //statement n
```



- Співставлення означає визначення при операторі match шуканого значення, після якого можна перерахувати кілька потенційних кейсів, кожен з оператором case. У місці виявлення співпадиння між match та case виконується відповідний код.

Python Starter

Довгоочікувані інструкції match/case в Python

Застосування конструкції if-elif-else:	Застосування конструкції match/case:
<pre>day = input('Введіть день тижня: ').lower() if day == 'понеділок': print('Початок робочого тижня') elif day == 'вівторок': print('Другий робочий день') elif day == 'середа': print('Середина тижня') elif day == 'четвер': print('Передостанній робочий день') elif day == 'п'ятниця': print('Останній робочий день тижня') elif day == 'субота' or day == 'неділя': print('Сьогодні вихідний') else: print('Немає такого дня тижня') print()</pre>	<pre>day = input("Введіть день тижня: ").lower() # Конструкція match/case: match day: case 'понеділок': print('Початок робочого тижня') case 'вівторок': print('Другий робочий день') case 'середа': print('Середина тижня') case 'четвер': print('Передостанній робочий день ') case 'п'ятниця': print('Останній робочий день тижня') case 'субота' 'неділя': print('Сьогодні вихідний') case _: print('Немає такого дня тижня')</pre>

- Тут відбувається перевірка кількох умов та виконання різних операцій на основі значення, яке ми знаходимо всередині **day**.
- Тим не менш, за допомогою інструкцій match-case ми позбавляємося від повторення **day ==**, що підвищує чистоту коду при тестуванні на відповідність багатьом різним умовам.

Дивіться наші уроки у відео форматі

ITVDN.com



Перегляньте цей урок у відео форматі на освітньому порталі [ITVDN.com](http://itvdn.com) для закріплення пройденого матеріалу.

Усі курси записані сертифікованими тренерами, які працюють у навчальному центрі CyberBionic Systematics



Перевірка знань

TestProvider.com

TestProvider

Мы помогаем людям оценить себя

Главная Услуги и цены Центр Тестирования Поддержка О нас

Регистрация Войти

Поиск сертификата

Мы в социальных сетях

Тестирование

Языки программирования и информационные технологии

Microsoft

C# ASP.NET MVC JavaScript Patterns OF Design SQL Architecture Guide WCF HTML&CSS XML SEO WPF HTML&CSS3 JQuery XNA SharePoint GUI for Android Windows Azure Platform Microsoft Patterns&Practices TFS SCRUM ReSharper TDD WWF LINQ Entity Framework Windows Forms Refactoring Microsoft Expression Blend 4 Windows Phone 8 Windows 8 AppStore Visual Studio Tips&Tricks MSF MEF SilverLight AJAX MEF Service Oriented Architecture

Добро пожаловать на TestProvider.com!

Сайт перенесен на новую облачную платформу с использованием системы единой авторизации Single Sign On. Если вы хотите восстановить статистику по предыдущим экзаменам обратитесь в [службу поддержки](#). Для восстановления информации с предыдущей версии сайта, просба написать в службу поддержки Ваш старый и новый логины.

ITVDN PROMETRIC TEST CENTER CyberBionic Microsoft Partner Learning Windows Azure Cloud Partner EBA

TestProvider – це online сервіс перевірки знань з інформаційних технологій. За його допомогою Ви можете оцінити Ваш рівень та виявити слабкі місця. Він буде корисним як у процесі вивчення технології, так і для загальної оцінки знань IT спеціаліста.

Після кожного уроку проходите тестування для перевірки знань на TestProvider.com

Успішне проходження фінального тестування дозволить Вам отримати відповідний Сертифікат.



Python Starter

Q&A

Python Starter

Дякую за увагу!

Інформаційний відеосервіс для розробників програмного забезпечення

