

Програмният език Python

Красимир Светославов Стойков

ЦЕЛИ НА СТАТИЯТА.

В текущата статия ще намерите описание на това какво представлява езикът Питон(Python), как да започнете да пишете програми на него, какви са предимствата и недостатъците му, както и някои прости примери за програми, написани на него.

КАК ЗАПОЧВА ВСИЧКО.

От уважение към труда на създателя на Питон, статията ще започне с неговото име. То е Гуидо ван Росум. Гуидо, който е холандски програмист, завършва магистратура в Амстердамския Университет през 1982 година, а през декември 1989 започва работа по Python, или както сам се изразява „хоби програмен проект“. Езикът е създаден в началото на деветдесетте години и е кръстен е на телевизионното шоу на BBC „Monty Python’s Flying Circus“ (Летящият цирк на Монти Пайтън).

СЪЩНОСТ.

Питон, както вече знаете, е език за програмиране. При това обектно-ориентиран и интерпретируем. Той е изключително лесен за научаване и ползване. Една от причините за това е че синтаксисът му е прост и елегантен. Същевременно, обаче, езикът предлага висока функционалност и голяма мощ. Той притежава полезни инструменти като гъвкави масиви и речници, за които биха били необходими дни, за да се напишат ефикасно на някои от другите езици, като например „C“. Поради това, че е обектно-ориентиран Питон позволява даден проект да бъде разделен на модули, като след това те биха могли да бъдат използвани за други програми.

Езикът, както вече споменах по-горе, е интерпретируем. Той не се нуждае от компилиране и свързване, а голямо предимство е, че интерпретаторът на Python е реализиран на голяма част от платформите и операционните

системи. Вероятно може да се каже, че езикът е замислен и като лесно разширяем.

ОСОБЕНОСТИ.

Преди да продължим с това как да започнем да пишем на Питон и как да напишем първата си програма е хубаво да се спомене, че езикът притежава някои интересни и интригуващи(поне за мен) особености, които веднага хващат окото. Първо лексикалното групиране не става чрез начални и крайни скоби, begin и end, или каквито и да е било други ключови думи, а единствено чрез отстъп(празно пространство след началото на реда). И второ, за да използваме дадена променлива, не трябва да я декларираме или дефинираме преди това. Следователно в Питон не е нужно да опоменаваме типа на един обект преди да го използваме. Това, разбира се, си има и своите лоши страни, но кому е нужно да се притеснява за тях още преди да е започнал с писането на код. Нима не е приятно да не се занимаваме с грешки от сорта на недекларирани променливи? На всичкото отгоре липсата на ограниченията, идващи с типовете данни, които понякога се различават по размер в различните системи, правят езика още по-достъпен, а кодът, написан на него - преносим.

НЕОБХОДИМИ ЗА ПОЛЗВАНЕТО НА PYTHON СРЕДСТВА.

Сега пред нас остава една друга проста стъпка и тя е да си намерим текстов редактор, в който да пишем код, и да си изтеглим самия Python. Ако използвате Linux или BSD по всяка вероятност имате инсталиран Python както и подходящ текстов редактор(като Гедит например). Ако пък сте Windows потребители, то тотава notepad++ би бил разумен избор за редактор. Също така ако сте под Windows ще ви се наложи да си изтеглите сами PYTHON, който междувпрочем предлага IDLE, текстов редактор, който можете да ползвате и който е добър избор за начинаещи.

ПЪРВАТА НИ PYTHON ПРОГРАМА: „HELLO, WORLD!”

И така, стига с общите приказки. Стъпката, с която обикновено се започва приключението по навлизането в нов език, това е първата програма. Обикновено тя се състои в това на монитора да се изведе „Hello, world!”. За целта си отворете интерпретатора. За Linux просто напишете в терминала `python`, а в Windows отворете IDLE. Сега да напишем кода, който ще изведе на екрана желаното. В Python това става по следния необременяващ начин:

```
print "Hello, world!"
```

За да проверим дали действа, просто записваме кода в файл с окончание `.py` чрез текстовия редактор. После пишем `python името-на-файла` в терминала или ако сме под Windows след като сме влезли в IDLE и сме дали `File->New Window` и сме въвели кода в отворилия се прозорец, натискаме `Run->Run Module`. Пред очите ви би трябвало да се изведе `Hello, world!`. Ако не се е получи, значи най-вероятно сте допуснали някаква грешка в кода.

ИЗПЪЛНЕНИЕ НА КОДА.

Всъщност има два начина, по които може да се изпълни кодът. В Linux първият вариант е да се пише директно в интерпретатора. Резултатът ще се появи веднага. Недостатъкът тук е, че след затварянето му, кодът ще се изгуби. По тази причина кодът винаги трябва да се запазва в `.py` файлове. Другият начин е в интерпретатора да се въведе `python името-на-файла`. Интерпретаторът ще отвори файла, ще го изпълни и ще изведе резултата. За Windows е най-добре да се използва метода, при който изпълняваме кода чрез `Run->Run Module` през IDLE. Другият начин е през командния промпт, но няма особено голям смисъл да се спираме на него.

КОМЕНТАРИ

След малко ще продължим с някои основни неща за Python, но преди това, както подсказва и наименованието на секцията, ще си поговорим за коментарите. Те са нещо изключително полезно в болшинството от случаите. Коментарите не влияят на изпълнимия код или иначе казано интерпретаторът ги пренабрегва. Добрият стил

на програмиране изисква програмистът да пише коментари за човека след себе си, който ще чете кода (чрез коментари човек може да съобщи какво прави дадена сложна конструкция, да подчертае някаква особеност или дори грешка, която не е успял да оправи). Статията няма за цел да ви научи как се пишат добри коментари, а просто да ви покаже как се пишат такива. Първият начин е с поставянето на символа `#`. Целият текст след него до края на реда е коментар. Всеки нов ред изисква нов `#`. Другият начин е удобен за по-дълги коментари, които се разпростират на повече от един ред. Методът е прост: Всичко, след поставянето на тройни кавички `'''` (или `"""`) и преди повторението им, се води за коментар.

ТРИ ОСНОВНИ КОНЦЕПЦИИ: ИМЕНА, ОБЕКТИ, ОПЕРАТОРИ

Вече бе споменато, че езикът за програмиране Питон се отличава със по-особените си синтактични правила (или липсата на такива). Отбелязах, че една доста полезна особеност на езика е, че за да използваме дадена променлива не е нужно да я декларираме предварително. Дори и това да не е вашият първи програмен език, пак е наложително да изясним какъв е смисълът на имената, обектите и операторите, което ще ви помогне по-лесно да разберете колко практично е да не декларираме променливите, които ползваме, предварително. Вероятно всеки може да се досети какво представляват имената (променливите). В Python с тях назоваваме обекти, също както и в „реалния” свят, тоест имената са си просто имена. Може да се каже, че те са нещо като табели или надписи, указващи обекта, който манипулираме. Това е проста концепция за разлика от обектите, които съществуват в паметта на компютъра или някъде другаде. Има най-различни обекти, някои от които по-сложни от други. Тази статия няма за цел да ви изучи на Python, а да покаже накратко какво представлява езикът и основните му характеристики и поради тази причина няма да се спираме върху нещо по-сложно, а ще хвърлим поглед на някои прости обекти. Такива например, са числата, които всички познаваме от математиката. В Python те не са нещо по-

необикновено. В този случай не се сещам за по-добро обяснение от нагледното:

```
x = 10
y = 2.5
z = x**10
q = y*2
```

Виждайки този код на Python можем да разберем няколко неща. Първо, `x` и `y` са просто имена обозначаващи обектите 10 и 2.5 (може да се каже, че `x` има стойност 10, а `y` 2.5). Второ, `z` приема стойността `x` на степен десета, а `q` - `y` умножено по две. Различни променливи могат да имат еднаква стойност без да зависят една от друга. Също така една променлива може да променя съдържанието си.

В началото на тази секция бе казано, че освен имена и обекти в Python има и оператори. В горния пример оператори са `'='`, `'**'`, и `'*'`. Чрез операторите манипулираме обектите. В този случай символът `'='` е оператор за присвояване на стойност (наименование на обект), `'**'` е оператор за повдигане на степен, а `'*'` за умножение. Както се досещате има оператори и за някои от останалите манипулации с числа като например `'/'` (деление), `'%'` (деление, като се взима остатъкът от самото деление), `'//'` (деление без остатък), `'+'` (събиране) и `'-'` (изваждане).

Вероятно знаете, че числата се групират по най-различен начин (на цели, дробни и така нататък). По технически причини в програмирането има и различни типове числа. Два от тези типове са `integer`, известни и като целочислени, и `float`, числа с плаваща запетая. В горния пример `y` е име на число с плаваща запетая, а `x` на `integer`. Тук идва и логичният въпрос – какво става ако например разделим пет на две? Така делим целочислени числа, но се получава `float`. По принцип Python връща две и пренабрегва остатъка, но е възможно в по-новите версии автоматично да превръща отговора в `float`. Когато делим смесени типове (например `5/2.5`) получаваме число с плаваща запетая. Също така ако използваме операторът `'/'` (деление без остатък) дори и с `float` числа, пак ще получим отговор без остатък:

```
x = 5.0/2.0 # x тук ще приеме стойност 2
```

СИМВОЛНИ НИЗОВЕ.

Питон предлага много удобно средство и това е низът. Низът е структура данни, която се използва за манипулиране на текст. Както и в случая с числата, тук пак можем да присвояваме низове на просто една променлива или пък да извършваме основните действия чрез познатите ни вече оператори (`+`, `-`, `*`, `/`, `%`, `//`).

```
x = "frodo"
y = 'frodo'
```

Двата низа `'frodo'` и `"frodo"` са напълно еднакви. Както сами се досещате низове могат да се записват и чрез единични кавички и чрез двойни – разлика няма. Ето и малко действия с умножение и събиране:

```
"sauron"*5
"Darth"+"Vader"
```

Първият низ ще се превърне в `'sauronsauronsauronsauronsauron'`, а вторият в `'DarthVader'`.

Както виждате `“+”` създава нов низ състоящ се от първите два низа, а `“*”` създава низ който повтаря старият пет пъти. Не може да се умножава низ по низ или пък да се събират низ и число. Символните низове са вид последователности от низове с по един символ. Като последователности ние имаме възможност да пулчуваме достъп до различните елементи на низовете, чрез техника, наречена индексване и нахъсване: последователностите си имат обекти, като всеки обект си има номерче, започвайки от нула. `'asmodean'`, ако си мислим за него като за последователност:

```
a s m o d e a n
0 1 2 3 4 5 6 7
```

Възможно е да използваме и обратно индексване от типа: `x[-1]` и така вземаме последния елемент. Както виждате, за да се доберем до даден обект от низа използваме името-на-променливата[индекс].

Има още неща, които може да се кажат за низовете, но тази секция беше просто пример за едно от средствата на езика – последователността и за индексването.

ЗАКЛЮЧЕНИЕ

Текущата статията не обхваща кой знае колко неща от езика Python, но описва някои от

особеностите на езика като например, че кодът не се компилира и свързва и че отстъпите са от значение за групирането на му. Езикът разполага с много повече от средствата, описани тук, но този текст няма за цел да ви научи да програмирате на езика, а да хвърли един поглед

върху част от него. Ще завърша с това, че Python е доказателство, че в простотата се крие много красота, а код, който е същевременно лесен за оформяне, поради необвързаността с тежки синтактични правила, и красив, е великолепно нещо.

Съставил статията:

Красимир Светославов Стойков, email:

lukistar.kr@gmail.com, 11Б клас,

Технологично училище „Електронни системи” към

ТУ, София, email: www.elsys-bg.org

Източници:

1.Python: Червеното хапче; Автор: Павел Пенев;

email: pavelludiq@gmail.com (оригинален текст на

основния източник -

<http://store2.data.bg/pavelludiq/Python-draft.pdf>

2.<http://bg.wikipedia.org/wiki/Python>

3.<http://www.download.bg/?cls=articles&mtd=single&id=584304>