

آموزش گام به گام برنامه نویسی پایتون



گردآوری و تألیف: مرتضی غیائی

معرفی:

پایتون یک زبان برنامه نویسی همه منظوره، سطح بالا، شیء‌گرا و مفسر است که توسط خودو فان روسوم در سال ۱۹۹۱ در کشور هلند طراحی شد. فلسفه ایجاد آن تأکید بر دوهدف اصلی خوانایی بالای برنامه های نوشته شده و کوتاهی و بازدهی نسبی بالای آن است.

کلمات کلیدی و اصلی این زبان به صورت حداقلی تهیه شده اند و در مقابل کتابخانه هایی که در اختیار کاربر است بسیار وسیع هستند.

پایتون یکی از محدود زبان های برنامه نویسی است که می توان ادعا کرد ساختاری ساده و قدرتمند دارد، از این رو، یادگیری این زبان همواره به افراد مبتدی که شاید هیچ تجربه ای در برنامه نویسی نداشته باشند، توصیه می شود و از طرف دیگر، استفاده از این زبان برای حل مسائل مختلف و پیچیده انتخاب اول بسیاری از برنامه نویسان حرفه ای بوده است.

بر اساس رتبه بندی سایت Tiobe، زبان برنامه نویسی Python در سپتامبر سال ۲۰۱۵ با سه پله صعود نسبت به زمان مشابه در سال قبل در جایگاه پنجم قرار گرفته است که نشان دهنده ی رشد محبوبیت این زبان در میان برنامه نویسان سراسر دنیا است.

همان طور که می دانید هر زبان برنامه نویسی ویژگی ها و قابلیت های خاص خود را دارد که آن را از سایر زبان ها متمایز می سازد و علت شکل گیری زبان های مختلف نیز پاسخگویی به نیازهای متفاوت و متنوع کاربران با استفاده از همین قابلیت های متمایز است. به همین دلیل، پیش از شروع به یادگیری هر زبان ابتدا باید نیازها و هدف خود را از یادگیری آن زبان در کنار قابلیت هایش قرار دهیم و در صورت تطبیق آن ها باهم، قدم در راه یادگیری بگذاریم. بنابراین، برای آشنایی بیش تر با زبان پایتون، در ادامه به معرفی برخی از ویژگی ها و قابلیت های آن میپردازیم:

1. سادگی و صراحت: پایتون یک زبان ساده و کمینه گرا است. وقتی نگاهی به سورس کد یک برنامه ی نوشته شده به زبان پایتون بیاندازیم، احساس می کنیم که با یک متن انگلیسی صریح مواجه هستیم. شاید بتوان گفت این بزرگ ترین نقطه ی قوت پایتون است که به جای درگیر کردن برنامه نویس به جزئیات زبان به او اجازه می دهد تا روی حل مسئله تمرکز داشته باشد. همین موضوع سرعت کدنویسی و خوانایی این زبان را هم افزایش داده است.

2. منحنی یادگیری کم شیب: قطعاً عامل اصلی این موضوع که یادگیری پایتون به عنوان قدم اول به مشتاقان برنامه نویسی و حتی کودکان توصیه می شود سینتکس فوق العاده ساده ی آن است. همان طور که گفتیم صراحت زبان پایتون نه تنها خوانایی آن را افزایش داده است، بلکه با حذف پیچیدگی هاسهولت یادگیری آن را نیز بیش تر کرده است.

3. رایگان و متن باز بودن : توزیع های مختلف زبان برنامه نویسی پایتون کاملاً رایگان بوده و هر برنامه نویس می تواند سورس کد آن را بخواند، آن را تغییر دهد، و در برنامه های خود از اسکریپت های آن استفاده کند.

4. سطح بالا بودن : پایتون از جمله زبان های قدرتمند سطح بالا است که برنامه نویس را درگیر جزئیات سطح پایین مثل مدیریت حافظه یا کار با ثباتها و غیره نمی کند.

5. قابل حمل بودن : ماهیت متن باز پایتون موجب شده است که این زبان با پلتفرم های مختلف سازگار باشد. بنابر اعلام رسمی سایت پایتون، در حال حاضر این زبان روی ۶ پلتفرم از جمله Windows، GNU/Linux، Macintosh، Solaris، Android، iOS، و ... کار می کند و برنامه های نوشته شده به این زبان بدون نیاز به تغییر یا با تغییرات بسیار جزئی روی تمام پلتفرم ها اجرا می شوند.

6. زبانی مفسری : برخلاف زبان های کامپایلری مانند C یا جاوا، زبان برنامه نویسی پایتون یک زبان مفسری است و سورس کد برنامه های نوشته شده به این زبان با استفاده از یک مفسر اجرا می شود که همین موضوع قابل حمل بودن آن را افزایش می دهد.

7. شیء گرایی : پایتون در مقایسه با زبان هایی مانند جاوا یا ++C، روش قدرتمندتر و ساده تری را برای اجرا برنامه های شیء گرا به کار می گیرد.

8. توسعه پذیری : یکی از مشکلات زبان مفسری پایتون سرعت پایین اجرا در مقایسه با زبان های کامپایلری مانند C یا جاوا است. حال اگر بخواهید قطعه ای از کدها سریع تر اجرا شود یا اگر بخواهید بخشی از الگوریتم برنامه ی خود را پنهان کنید می توانید آن بخش را به زبان C، ++C یا جاوا بنویسید و آن را در میان کدهای پایتون برنامه ی خود قرار دهید.

9. تعبیه پذیری : علاوه بر این که می توان کدهای زبان های دیگر را در برنامه های نوشته شده باه زبان پایتون قرار داد، می توان قطعه کدهایی را به زبان پایتون نوشت و در سورس کد برنامه های C، ++C یا جاوا نشانند و به این ترتیب قابلیت های اسکریپتی به سورس کد مدنظر اضافه نمود.

10. کتابخانه ی گسترده : پایتون از یک کتابخانه ی استاندارد غنی بهره می برد و در کنار این کتابخانه ی وسیع، کتابخانه های سایر توسعه دهندگان نیاز به سرعت در حال توسعه می باشند که در مجموع ابزارهای مناسبی را برای ایجاد اسناد، رابط های گرافیکی کاربر GUI، مرورگرهای وب، رمزنگاری، هوش مصنوعی، پست الکترونیکی، بازی سازی، داده کاوی، ایجاد و مدیریت وب سایت، و بسیاری کاربردهای دیگر در اختیار برنامه نویسان قرار می دهد.

11. همه منظور بودن : پایتون یک زبان برنامه نویسی با طیف گسترده ای از کاربردها است که در حوزه های مختلف و متنوع کاربرد داشته است که از جمله مهم ترین کاربردهای آن در طی سالیان گذشته می توان به موارد زیر اشاره کرد:

✓ موتور جستجوگر گوگل و موتور گرافیکی یوتیوب

✓ ساخت برنامه های کاربردی علمی در سازمان فضایی ناسا، Fermilab

✓ بخشی از سرویس ایمیل یاهو

✓ تست سخت افزار در Cisco، Intel، IBM

✓ ابزارهای نصب لینوکس در نسخه ی Redhat

✓ سرویس ابری Dropbox

و بسیاری کاربردهای دیگر نظیر طراحی سایت های دینامیک، تولید نرم افزارهای دسکتاپ، انیمیشن سازی، بازی سازی، شبکه، امنیت، پایگاه داده، داده کاوی، ساخت برنامه های محاسباتی و کاربردی در رشته های مختلف نظیر ریاضی، فیزیکی، آمار، زیست و غیره.

در نهایت می توان گفت که پایتون ابزاری مهیج و قدرتمند در اختیار برنامه نویسان است که کار با آن ساده و سرگرم کننده می باشد و تسلط بر آن کاربران را وارد دنیایی شگفت انگیز و بی نهایت می کند که هرکس می تواند متناسب با توانایی هایش از امکانات آن برای حل مسائل خود بهره مند شود.

فصل اول – شروع آموزش برنامه نویسی در پایتون:

۱- سطرها:

مفسر پایتون و همچنین کاربران، کدهای درون هر ماژول را به صورت تعدادی سطر مشاهده می کنند. در پایتون دو نوع سطر وجود دارد: **الف-سطرهای فیزیکی** سطرهایی هستند که توسط ویرایشگرهای متن شماره گذاری می شوند و به سادگی توسط کاربر قابل تشخیص می باشند. **ب- سطرهای منطقی** ، که برداشت مفسر از اجرای برنامه است. هر سطر بیان گر یک دستور پایتون است. به عنوان مثال ، دستورات زیر را در نظر بگیرید:

```
>>> name = "Fanavarienovin.net"
>>> print(name)
```

دستور اول رشته fanavarienovin.net را به متغیر name نسبت می دهد و دستور دوم، عبارت fanavarienovin.net را نمایش می دهد. در این دستورات، هر سطر منطقی یک سطر فیزیکی در نظر گرفته شده است. با اجرای این دستورات خروجی زیر نمایش داده میشود:

Fanavarienovin.net

گاهی اوقات هر سطر فیزیکی می تواند شامل چند سطر منطقی باشد. در این حالت ، باید بین سطرها، کاراکتر ”؛” قرار داد. به عنوان مثال، دستورات زیر را ببینید:

```
>>> name = "Fanavarienovin.net"; print(name)
```

با اجرای این دستورات نیز خروجی زیر نمایش داده میشود:

Fanavarienovin.net

گاهی اوقات برای خوانایی بیشتر بهتر است دستورات یک سطر منطقی در چند سطر فیزیکی تایپ شود؛ به عنوان مثال، دستورات زیر را مشاهده کنید:

```
>>> message = "Python is a \
good programing language"
>>> print(message)
```

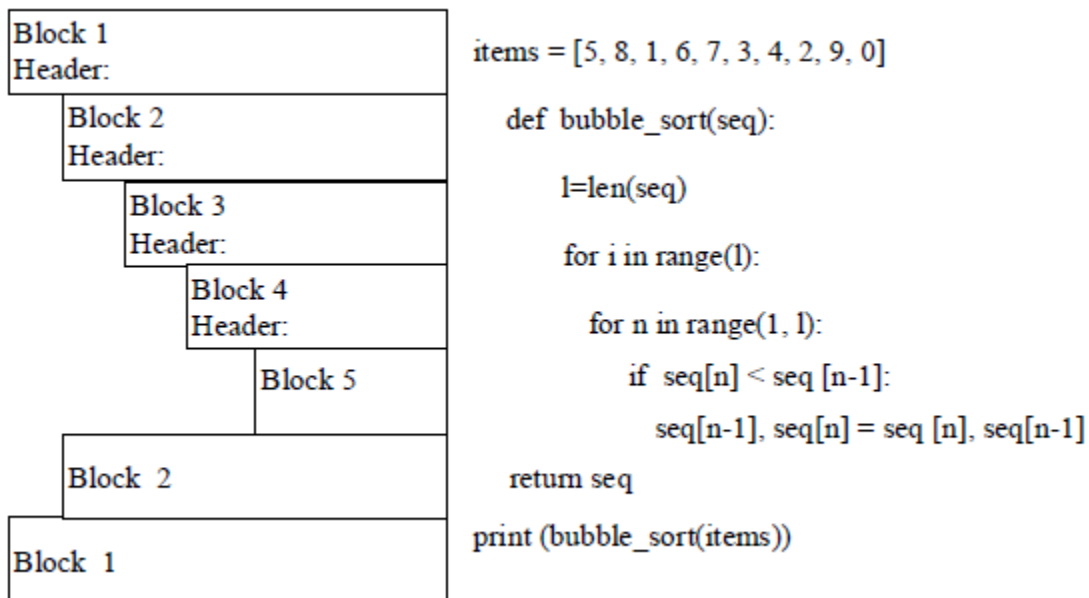
در این مثال، خطوط اول و دوم یک دستور منطقی هستند که در دو سطر آمده اند. برای توسعه یک دستور در چند سطر فیزیکی از کاراکتر “\” استفاده می شود.
با اجرای این دستورات خروجی زیرنمایش داده میشود:

Python is a good programming language

تعداد کاراکترهای هر سطر فیزیکی نباید از ۷۳ کاراکتر بیش تر شود.
سطرهای خالی، برای افزایش خوانایی برنامه به کار می روند که شامل فضای خالی (Tab یا Space) هستند و توسط مفسر نادیده گرفته می شوند و به بایت کد ترجمه نمی گردند.

۲- بلاک بندی:

بلاک بندی، یکی از امکاناتی است که برای افزایش خوانایی کد پایتون به کار می رود. در زبان پایتون برای ایجاد بلاک از **تورفتگی** سطرها استفاده می شود. در واقع، تورفتگی میزان فضای خالی (Tab یا Space) است که در ابتدای هر سطر فیزیکی قرار می گیرد. تمام دستورات موجود در یک بلاک باید به یک میزان نسبت به سرآیند خود تورفتگی داشته باشند. یعنی، تعداد فضای خالی تمام دستورات آن بلاکها نسبت به سرآیند یکی باشد. طبق شکل زیر:



نکته: برای ایجاد هر تورفتگی از چهار جای خالی (کلید Space) استفاده کنید.
هرگز برای تورفتگی از کلیدهای space و Tab باهم استفاده نکنید.

۳- شناسه ها:

شناسه یا **identifier** یک اسم است که به منظور شناسایی متغیر، تابع، کلاس، ماژول یا دیگر اشیا به آن ها تخصیص داده می شود. شناسه با یک حرف از **A** تا **Z** یا **a** تا **z** یا زیرخط _ آغاز شده و به دنبال آن یک یا چند صفر، حرف، زیرخط و عدد قرار می گیرد.

استفاده از علائم نگارشی همچون **@** ، **\$** و **%** در شناسه مجاز نمی باشد. پایتون یک زبان حساس به کوچک و بزرگی حروف است. از این رو، دو واژه ی **Manpower** و **manpower** دو شناسه ی کاملا متفاوت از یکدیگر هستند.

قوانین نام گذاری توابع، متغیرها و اشیا در زیر فهرست شده:

۱. اسم کلاس ها با حرف بزرگ آغاز می شود. دیگر شناسه ها با حرف کوچک شروع می شود.
۲. آغاز کردن یک شناسه با زیرخط _ ، بیانگر این است که آن شناسه **private** می باشد.
۳. آغاز کردن یک شناسه با دو زیرخط نشانگر **strongly private** بودن آن شناسه است.
۴. اگر شناسه ای با دو زیرخط پشت سرهم پایان یابد، در آن صورت شناسه ی مورد نظر اسم خاص **language defined** می باشد.

لیست زیر کلمات رزرو شده ی زبان پایتون را نمایش می دهد. این کلمات را نمی توان به عنوان اسم متغیر، ثابت (**constant**) یا هر چیز دیگر استفاده کرد. لازم به ذکر است که تمامی کلیدواژه های پایتون تماما با حروف کوچک نوشته می شوند.

And	exec	Not
Assert	finally	or
Break	for	pass
Class	from	print
Continue	global	raise
def	if	return
del	import	try
elif	in	while
else	is	with
except	lambda	yield

نکته : اگر از اسامی غیرمجاز برای متغیرها و ثابتها و ... استفاده کنید با خطای نحوی زیر مواجه می شوید:

SyntaxError : invalid syntax

۴- علامت نقل قول یا کوتیشن در پایتون:

می توان در پایتون از تک کوتیشن ('') ، دابل کوتیشن (") و سه کوتیشن (""") (یا """) استفاده کرد . مقداری که درون این علامت ها قرار می گیرد، یک رشته ی نوشتاری (**string literal**) را تشکیل می دهد .بایستی دقت داشت که یک رشته با یک نوع کوتیشن آغاز شده و با همان نوع نیز پایان می یابد . از "" یا "" "" برای محصور کردن یک رشته که در چندین خط ادامه دارد استفاده می شود .تمامی نمونه های زیر مجاز و صحیح هستند:

```
word = 'word'
sentence = "This is a sentence."
paragraph = """This is a paragraph. It is
made up of multiple lines and sentences."""
```

۵- توضیحات در پایتون (Comment)

علامت # اگر داخل کوتیشن محصور نباشد، نشانگر شروع **comment** خواهد بود .تمام کاراکترهایی که پس از # قرار می گیرند تا پایان آن خط بخشی از توضیح محسوب می شوند و مفسر پایتون آن ها را نادیده گرفته و اجرا نمی کند.

```
#!/usr/bin/python
# First comment
print ("Hello, Python!") # second comment
```

خروجی کد بالا:

```
Hello, Python!
```

۶- متغیرها: (Variable)

متغیر صرفا فضاهای رزرو شده در حافظه هستند که مقادیری را در آن ها ذخیره می کنیم، بدین معنا که در زمان ایجاد یک متغیر، بخشی از حافظه اشغال شده و به آن متغیر تخصیص داده میشود.

بسته به نوع داده ای متغیر، مفسر بخشی از حافظه را رزرو کرده و تصمیم می گیرد چه مقداری در حافظه ی تخصیص داده شده، ذخیره گردد. بنابراین، با تخصیص نوع داده های مختلف به متغیرها، می توانید اعداد صحیح **integer** ، اعداد اعشاری **decimal** یا **character** در این متغیرها ذخیره کنید.

۱-۶: تخصیص مقادیر به متغیرها

برای تخصیص حافظه، در پایتون نیازی به اعلان صریح متغیر نیست. زمانی که مقداری را به متغیر انتساب می دهد، اعلان به صورت خودکار رخ می دهد. مانند زبان های برنامه نویسی دیگر، انتساب مقدار به متغیر توسط علامت مساوی " = " صورت می پذیرد.

عملوندی (**operand**) که در سمت چپ عملگر " = " (**operator**) قرار می گیرد، اسم متغیر و عملوندی که در سمت راست عملگر " = " قرار می گیرد، مقداری است که در متغیر ذخیره میشود. مثال:

```
counter = 100 # An integer assignment
miles = 1000.0 # A floating point
name = "John" # A string
print (counter)
print (miles)
print (name)
```

۷- نوع داده های رایج

داده های ذخیره شده در حافظه می توانند از هر نوعی باشند. به عنوان مثال، سن یک شخص به صورت یک مقدار عددی ذخیره شده و آدرس وی در قالب حروف الفبا داخل حافظه ذخیره می شود. پایتون دارای نوع داده های متعددی است که عملیات ممکن بر روی آن ها و روش های ذخیره ویژه ی هر یک را تعریف می کند.

در کل زبان پایتون از نوع داده های زیر پشتیبانی می کند:

۱. نوع عددی
۲. نوع رشته ای
۳. لیست
۴. نوع داده ی چندتایی یا tuple
۵. نوع Dictionary

۷-۱ اعداد یا نوع عددی در پایتون

نوع داده های عددی قادرند مقادیر عددی را در خود نگه دارند. اشیا **number** زمانی ایجاد می شوند که مقداری را به آن تخصیص دهید. برای مثال:

```
var1 = 1
var2 = 10
```

پایتون نوع های عددی زیر را پشتیبانی می کند:

۱. **int** اعداد صحیح علامت دار
۲. **Long** اعداد صحیح بسیار بزرگ یا **long integers**، آن ها را می توان به صورت شانزده شانزدهی و هشت هشتی نمایش داد
۳. **Float** مقادیر حقیقی ممیز شناور
۴. **Complex** اعداد مختلط

مثال:

int	long	float	complex
10	51924361L	0.0	3.14j
100	-0x19323L	15.20	45.j
-786	0122L	-21.9	9.322e-36j
080	0xDEFABCECBDAECBFBAE	32.3+e18	.876j

- پایتون به شما اجازه می دهد برای مشخص کردن نوع عددی **Long** از **l** کوچک استفاده کنید، اما برای اینکه آن **l** با عدد **1** اشتباه گرفته نشود، توصیه می کنیم از **L** بزرگ استفاده کنید. بنابراین پایتون اعداد صحیح بسیار بزرگ را با **L** بزرگ نمایش می دهد.
- یک عدد مختلط متشکل است از دو عدد ممیز شناور حقیقی و یک بخش که یکه ی موهومی نام دارد. برای مثال در **x + yj**، **x** و **y** اعداد حقیقی هستند و **j** نشانگر واحد یا یکه ی موهومی (**imaginary**) می باشد.

۷-۲ رشته ها در پایتون

رشته ها در پایتون عبارت است از مجموعه کاراکترهای همجوار که در علامت نقل قول نمایش داده می شوند. پایتون از هر دو شکل تک کوتیشن و دابل کوتیشن پشتیبانی می کند. می توان بابه گیری از عملگر برش یا

([] [:]) **slice operator** که اندیس آن در آغاز رشته با اندیس 0 شروع شده و تا - 1 در انتها ادامه می یابد، بخش هایی از یک رشته را استخراج کرد. علامت (+) یک عملگر اتصال است که دو رشته را به هم پیوند می دهد . علامت (*) درواقع یک **repetition operator** است که دستوری را تکرار می کند (برای مثال یک رشته را دوبار چاپ مینماید).

مثال:

```
str = 'Hello World!'
print (str)      # Prints complete string
print (str[0])   # Prints first character of the string
print (str[2:5]) # Prints characters starting from 3rd to 5th
print (str[2:])  # Prints string starting from 3rd character
print (str * 2)  # Prints string two times
print (str + "TEST") # Prints concatenated string
```

نتیجه ی زیر حاصل می گردد:

```
Hello World!
H
llo
llo World!
Hello World!Hello World!
Hello World!TEST
```