

آموزش نصب پایتون در ویندوز

برای نصب پایتون روی ویندوز، مراحل زیر را دنبال کنید :

مرحله ۱: دانلود پایتون

- به وبسایت رسمی پایتون بروید www.python.org.
- از منوی اصلی، گزینه Downloads را انتخاب کنید
- نسخه ویندوز مناسب سیستم خود (۳۲ بیتی یا ۶۴ بیتی) را دانلود کنید. معمولاً برای سیستمهای جدیدتر، نسخه ۶۴ بیتی مناسب است .

مرحله ۲: اجرای فایل نصبی

- فایل نصبی دانلودشده (معمولاً با نامی شبیه `python.exe.x.x-3`) را باز کنید
- در پنجره نصب، گزینه `Add python to PATH` را حتماً تیک بزنید (این کار بسیار مهم است و دسترسی به پایتون از خط فرمان را آسان میکند).
- روی دکمه `Customize Installation` یا `Install Now` را کلیک کنید
- `Install Now` : پایتون با تنظیمات پیشفرض نصب میشود
- `Customize Installation` .: امکان انتخاب ویژگیها و مسیر نصب به شما داده میشود.

مرحله ۳: اتمام نصب

- پس از اتمام فرآیند نصب، پنجرهای نمایش داده میشود که گزینه `Disable path length limit` را ارائه میدهد.
- روی آن کلیک کنید (اختیاری، اما پیشنهاد میشود).
- روی دکمه `Close` کلیک کنید

مرحله ۴: بررسی نصب

- اجرای `Command Prompt (CMD)`
- کلیدهای `R + Windows` را فشار دهید، سپس `cmd` را تایپ کرده و `Enter` بزنید
- دستور زیر را اجرا کنید :

```
python --version
```

اگر نسخه پایتون مثل Python 3.x (x.x3) نمایش داده شد، نصب به درستی انجام شده است.

آموزش نصب VSCode

- **دانلود VSCode** : به وب سایت رسمی VSCode بروید ، نسخه ویندوز مناسب سیستم خود (۶۴ بیتی یا ۳۲ بیتی) را دانلود کنید.
 - **اجرای فایل نصبی** : فایل دانلود شده (VSCode Setup-x.x.x.exe) را اجرا کنید . با شرایط و ضوابط موافقت کنید (تیک گزینه I accept the agreement).
 - **انتخاب مسیر نصب** : مسیر پیش فرض یا مسیر دلخواه خود را انتخاب کنید.
 - **تنظیمات اضافی** : گزینه های زیر را تیک بزنید:
 - Add to patch (recommended)**
 - Register Code as an editor for supported file types**
 - Add a shortcut to the desktop**
- نصب نرم افزار : روی Install کلیک کنید و منتظر بمانید تا فرایند نصب کامل شود.
- **اجرای VSCode** : گزینه Launch Visual Studio Code را تیک بزنید و روی Finish کلیک کنید.
 - **نصب افزونه پایتون** : از منوی Command palette (کلید ctrl + shift + p) گزینه python: select Interpreter را انتخاب کنید . مسیر مفسر پایتون نصب شده روی سیستم خود را انتخاب کنید.

آشنایی با ساختار پایه پایتون در کدنویسی

خطوط کد (Statements) :

- کدها به صورت خطی نوشته میشوند و هر خط یک دستور یا بیان (Statement) است.
- نیازی به علامت نقطه ویرگول در انتهای خطوط نیست (مگر اینکه چند دستور در یک خط نوشته شود).

تورفتگی (Indentation) :

- پایتون از تورفتگی برای تعریف بلاک های کد استفاده میکند (به جای براکت {} در زبان هایی مثل C یا Java)
- تمام خطوط مربوط به یک بلاک باید دارای تورفتگی یکسان باشند .
- استفاده نادرست از تورفتگی منجر به خطا میشود.

کامنت ها (Comments) :

- برای مستندسازی و توضیح کد استفاده میشوند و توسط مفسر اجرا نمیشوند .
- کامنت تک خطی : با # شروع میشود
- کامنت چند خطی : از سه تا نقل قول (' ') یا (" ") استفاده میشود .

متغیرها و انواع داده ها :

- نیازی به تعریف نوع متغیر نیست ؛ پایتون به صورتت پویا نوع داده را تشخیص میدهد .
- از نام های معنادار برای متغیرها استفاده کنید .

ورودی و خروجی (Input / Output) :

- از تابع input برای دریافت داده از کاربر و از print برای نمایش خروجی استفاده میشود .

کنترل جریان برنامه :

شامل شرط های (if , else if) و حلقه های (while , for)

توابع (Functions) :

از توابع برای دسته بندی کدهای تکراری استفاده میشود .

کتابخانه ها و ماژول ها :

میتوانید از ماژول های داخلی یا کتابخانه های خارجی برای گسترش قابلیت ها استفاده کنید .

مفهوم مفسر (Interpreter) و کامپایلر (Compiler) :

در دنیای برنامه نویسی، مفسر و کامپایلر ابزارهایی هستند که کدهای نوشته شده توسط برنامه نویس را به زبانی تبدیل میکنند که کامپیوتر میتواند آن را بفهمد و اجرا کند. این دو ابزار عملکرد مشابهی دارند، اما روش کارشان متفاوت است. اجازه بدهید به صورت ساده و گام به گام توضیح بدهم :

مفسر (Interpreter) :

- مفسر یک ابزار نرم افزاری است که کدهای نوشته شده در یک زبان برنامه نویسی را خط به خط میخواند ، تجربه میکند و بلافاصله اجرا میکند. در واقع مفسر کد را خط به خط اجرا میکند و اگر در حین اجرای برنامه خطایی رخ دهد اجرای کد متوقف میشود و خطا را گزارش میدهد .
- زبان های پایتون ، PHP جاوا اسکریپت ، روبي ، پرل و شل اسکریپت مفسری هستند.

ویژگی های مفسر :

- اجرای خط به خط : کد را خط به خط میخواند و اجرا میکند .
- خطایابی ساده تر : به محض بروز خطا ، اجرای کد متوقف میشود و خطا را نمایش میدهد .
- کندی اجرا : از آنجا که ک خط به خط اجرا میشود ، اجرای برنامه ها ممکن است کمی کندتر باشد.

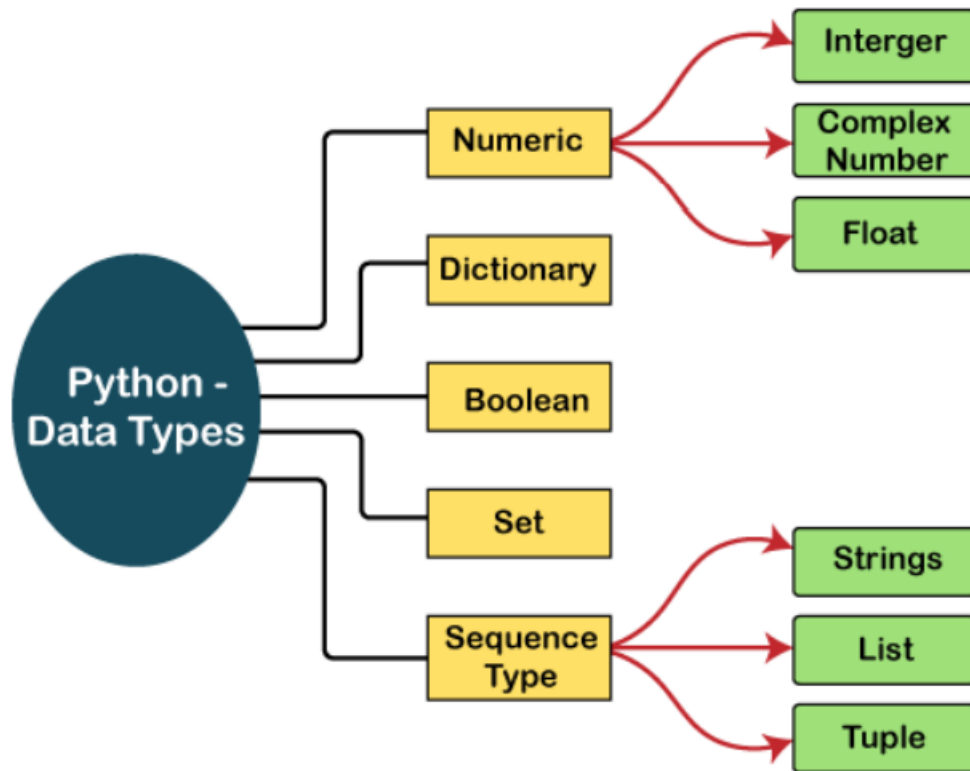
کامپایلر (Compiler) :

- کامپایلر نیز مانند مفسر ، کدهای برنامه نویس را به کدی تبدیل میکند که کامپیوتر میتواند آن را اجرا کند ، اما کامپایلر تمام کد را در یک مرحله تجزیه و تحلیل کرده و سپس تمام آن را به یک فایل اجرایی تبدیل میکند. این به این معنی است که قبل از اجرا ، کامپایلر کل کد را بررسی میکند و در صورتی که خطایی وجود داشته باشد ، تمام خطاها را گزارش میدهد و کد اجرا نمیشود تا زمانی که خطاها اصلاح شوند.
- زبان های C و ++C ، جاوا ، سی شارپ (C#) ، GO ، Swift ، Rust کامپایلری هستند.

ویژگی های کامپایلر :

- تجزیه کد به صورت یک پارچه : تمام کد را به یک فایل اجرایی تبدیل میکند
- اجرای سریع تر : پس از کامپایل کردن کد ، میتوانید آن را به سرعت اجرا کنید .
- خطایابی دشوارتر : ممکن است چندین خطا در کد وجود داشته باشد که باید پس از کامپایل گزارش شوند.

انواع داده در پایتون :



عملگرها در پایتون (operators) :

در پایتون ، عملگرها ابزارهایی هستند که برای انجام عملیات های مختلف روی داده ها (مثل اعداد ، رشته ها ، و غیره) استفاده میشوند . این عملگرها به دسته های مختلفی تقسیم میشوند که در اینجا آنها را به تفصیل توضیح خواهیم داد .

۱- عملگرهای ریاضی (Arithmetic Operators) :

عملگرهای ریاضی برای انجام عملیات پایه ای مانند جمع ، تفریق ، ضرب و تقسیم استفاده میشوند .

عملگر	توضیح	مثال	خروجی
+	جمع	5 + 3	8
-	تفریق	5 - 3	2
*	ضرب	5 * 3	15
/	تقسیم (اعشاری)	5 / 2	2.5
//	تقسیم صحیح (بدون اعشار)	5 // 2	2
%	باقی مانده تقسیم (مدول)	5 % 2	1
**	توان	5 ** 2	25

۲- عملگرهای مقایسه ای (Comparison Operators) :

عملگرهای مقایسه ای برای مقایسه دو مقدار استفاده میشوند و نتیجه آنها یک مقدار بولی (True یا False) است .

عملگر	توضیح	مثال	خروجی
==	برابر با	5 == 3	False
!=	نابرابر با	5 != 3	True
>	بزرگتر از	5 > 3	True
<	کوچکتر از	5 < 3	False
>=	بزرگتر یا مساوی با	5 >= 3	True
<=	کوچکتر یا مساوی با	5 <= 3	False

۳- عملگرهای منطقی (Logical Operators) :

عملگرهای منطقی برای انجام عملیات های منطقی مانند AND و OR و NOT بر روی مقادیر بولی استفاده میشوند .

عملگر	توضیح	مثال	خروجی
and	اگر هر دو شرط True باشند، نتیجه True است	True and False	False
or	اگر یکی از شرایط True باشد، نتیجه True است	True or False	True
not	معکوس کردن مقدار بولی	not True	False

۴- عملگرهای انتساب (Assignment Operators) :

عملگرهای انتساب برای انتساب مقدار به متغیرها استفاده میشوند . این عملگرها میتوانند به شکلی کوتاه تر انجام شوند.

عملگر	توضیح	مثال	خروجی
=	انتساب مقدار به متغیر	x = 5	x=5
+=	افزودن به مقدار متغیر	x = x + 3 (معادل x += 3)	x=8
-=	کم کردن از مقدار متغیر	x = x - 3 (معادل x -= 3)	x=2
*=	ضرب کردن مقدار متغیر با یک عدد	x = x * 3 (معادل x *= 3)	x=15
/=	تقسیم کردن مقدار متغیر با یک عدد	x = x / 3 (معادل x /= 3)	x=1.66
//=	تقسیم صحیح (بدون اعشار)	x = x // 3 (معادل x //= 3)	x=1
%=	محاسبه باقی مانده تقسیم	x = x % 3 (معادل x %= 3)	x=2
**=	محاسبه توان	x = x ** 2 (معادل x ** 2)	x=25

۵- عملگرهای عضویت (Membership Operators) :

این عملگرها برای بررسی عضویت یک عنصر در یک رشته یا لیست استفاده میشوند .

خروجی	مثال	توضیح	عملگر
True	'a' in 'apple'	بررسی اینکه آیا یک عنصر در داده‌ها وجود دارد یا خیر	in
True	'b' not in 'apple'	بررسی اینکه آیا یک عنصر در داده‌ها وجود ندارد یا خیر	not in

۶- عملگرهای هویتی (Identity Operators) :

عملگرهای هویت برای بررسی اینکه آیا دو متغییر به یک شی اشاره میکنند یا نه، استفاده میشوند.

خروجی	مثال	توضیح	عملگر
True/False	x is y	بررسی اینکه آیا دو متغیر به یک شی اشاره می‌کنند یا خیر	is
True/False	x is not y	بررسی اینکه آیا دو متغیر به یک شی اشاره نمی‌کنند	is not

اولویت اجرای عملگرها (Operator Precedence)

اولویت عملگرها مشخص میکند که در صورت وجود چندین عملگر در یک عبارت ، کدام یک ابتدا اجرا میشود. اولویت هادر

پایتون از بالا به پایین به شرح زیر هستند :

۱- پرانتزها : بالاترین اولویت را دارند .

۲- عملگرهای ریاضی : به ترتیب (** و * و / و // و %)

۳- عملگرهای مقایسه ای : (<= و < و > و >= و != و ==)

۴- عملگرهای منطقی : or , and , not

۵- عملگرهای انتساب : **= و /= و //= و /= و *= و -= و += و =

مقدار دهی متغییرها و دستور نویسی پایتون :

```
x = "Rahkar"
y = "is"
z = "your way"
print(x, y, z)
print(x + y + z)
x = 5
y = "Rahkar"
print(x)
print(y)
x = 1
y = 2.8
z = 1j
print(type(x))
print(type(y))
print(type(z))
```



```
Rahkar is your way
Rahkarisyour way
5
Rahkar
<class 'int'>
<class 'float'>
<class 'complex'>
```

```
35000.0
120000.0
-8.77e+101
----
<class 'float'>
<class 'float'>
<class 'float'>
```



```
x = 35e3
y = 12E4
z = -87.7e100
print((x))
print((y))
print((z))
print('----')
print(type(x))
print(type(y))
print(type(z))
```



```
(3+5j)
5j
(-0-5j)
<class 'complex'>
<class 'complex'>
<class 'complex'>
```

```
x = 3+5j
y = 5j
z = -5j
print(x)
print(y)
print(z)
print(type(x))
print(type(y))
print(type(z))
```



```
x = int(1)
y = int(2.8)
z = int("3")
print(x)
print(y)
print(z)
```



```
1
2
3
```

عملکرد تابع `:int()`

```
x = float(1)
y = float(2.8)
z = float("3")
w = float("4.2")
print(x)
print(y)
print(z)
print(w)
```



```
1.0
2.8
3.0
4.2
```

عملکرد تابع `:float()`

```
x = str("s1")
y = str(2)
z = str(3.0)
print(x)
print(y)
print(z)
```



```
s1
2
3.0
```

عملکرد تابع `:str()`

کار با رشته‌ها در پایتون

فرمت‌بندی رشته‌ها با استفاده از f-strings در پایتون

در پایتون، **f-strings** یا **formatted string literals** یکی از بهترین و ساده‌ترین روش‌ها برای فرمت‌بندی رشته‌ها (strings) است. این قابلیت از پایتون 3.6 به بعد معرفی شد و به شما این امکان را می‌دهد که به راحتی مقادیر متغیرها را در داخل رشته‌ها قرار دهید بدون اینکه نیاز به استفاده از عملگرهای پیچیده یا توابع اضافی داشته باشید.

نحوه استفاده از f-strings

برای استفاده از **f-strings**، کافی است که رشته را با حرف **f** قبل از علامت نقل قول (single یا double quotes) شروع کنید. سپس می‌توانید داخل رشته، از **آکولادها** { } برای قرار دادن متغیرها یا عبارات استفاده کنید.

فرمت کلی:

```
"متن ثابت {متغیر یا عبارت}"
```

مثال‌های ساده از f-strings

1. استفاده از متغیر در: f-string

فرض کنید دو متغیر داریم. برای وارد کردن این متغیرها داخل یک رشته، می‌توانیم از **f-string** به صورت زیر استفاده کنیم:

```
name = "Ali"  
age = 25  
text = f"شما {age} ساله هستید، {name} سلام"  
print(text)
```

پاسخ:

```
شما 25 ساله هستید، Ali سلام
```

فرمت‌بندی اعداد در f-strings

یکی از ویژگی‌های جالب f-strings این است که می‌توانید برای فرمت‌بندی اعداد، تاریخ‌ها و مقادیر دیگر از قالب‌های مختلف استفاده کنید.

مثال 1: فرمت‌بندی اعداد اعشاری

```
pi = 3.14159265358979
formatted = f"عدد پی برابر است با {pi:.2f}"
print(formatted)
```

مثال 2: استفاده از فرمت برای اعداد بزرگ

```
large_number = 1000000
formatted = f"عدد به‌صورت فرمت هزارگان: {large_number:,}"
print(formatted)
```

مثال 3: فرمت تاریخ‌ها

```
from datetime import datetime
current_date = datetime.now()
formatted = f"تاریخ امروز: {current_date:%Y-%m-%d %H:%M:%S}"
print(formatted)
```

متدهای نوع داده رشته ای (String)

متد	توضیح
upper()	تمام حروف رشته را به حروف بزرگ تبدیل می کند.
lower()	تمام حروف رشته را به حروف کوچک تبدیل می کند.
capitalize()	اولین حرف رشته را بزرگ کرده و بقیه را کوچک می کند.
title()	اولین حرف هر کلمه را بزرگ می کند.
strip()	فاصله های اضافی در ابتدا و انتهای رشته را حذف می کند.
rstrip()	فاصله های اضافی از ابتدای رشته را حذف می کند.
lstrip()	فاصله های اضافی از انتهای رشته را حذف می کند.
find()	موقعیت اولین وقوع یک زیررشته را می دهد. اگر پیدا نشد -1 باز می گرداند.
replace()	یک زیررشته را با زیررشته ای دیگر جایگزین می کند.
split()	رشته را بر اساس یک جداکننده مشخص تقسیم می کند و یک لیست برمی گرداند.
join()	یک لیست از رشته ها را با جداکننده ای خاص به هم می چسباند.
isalpha()	بررسی می کند که آیا رشته تنها شامل حروف است.
isdigit()	بررسی می کند که آیا رشته تنها شامل اعداد است.
isspace()	بررسی می کند که آیا رشته تنها شامل فضای خالی است.
startswith()	بررسی می کند که آیا رشته با یک زیررشته خاص شروع می شود.

متد	توضیح
endswith()	بررسی می کند که آیا رشته با یک زیررشته خاص تمام می شود.
count()	تعداد وقوع یک زیررشته در رشته را می شمارد.
zfill()	صفرهای اضافی به ابتدای رشته اضافه می کند تا طول رشته به اندازه مشخصی برسد.
expandtabs()	تمام کاراکترهای تب (\t) را به فضای سفید تبدیل می کند.
strip()	فضای خالی (whitespace) را از ابتدای و انتهای رشته حذف می کند.
isupper()	بررسی می کند که آیا تمام حروف رشته به حروف بزرگ است.
islower()	بررسی می کند که آیا تمام حروف رشته به حروف کوچک است.
isalnum()	بررسی می کند که آیا رشته فقط شامل حروف و اعداد است.
isnumeric()	بررسی می کند که آیا رشته تنها شامل اعداد است.
istitle()	بررسی می کند که آیا رشته مانند یک عنوان (با حروف بزرگ شروع شده و سایر حروف کوچک) است.
isidentifier()	بررسی می کند که آیا رشته یک شناسه معتبر پایتون است (مثل نام متغیر).
format()	فرمت بندی رشته ها با استفاده از جایگذاری مقادیر.