

# آموزش پایتون

تهیه و تنظیم: محسن یعقوبی

این فایل برای دوره کشوری و پروژه محور SciSmart تهیه شده است

## پایتون چیست؟

پایتون یک زبان برنامه‌نویسی سطح بالا، تفسیری، همه‌منظوره و شی‌گرا است که در سال ۱۹۹۱ توسط خیدو فان روسوم ایجاد شد.

فلسفه طراحی آن بر خوانایی کد تأکید دارد و برنامه‌نویسان را تشویق می‌کند که کدهای تمیز و قابل فهم بنویسند.

## ویژگی‌های کلیدی پایتون

۱. ساده و خوانا: نحو (Syntax) ساده و شبیه به زبان انگلیسی دارد که یادگیری آن را برای مبتدیان آسان می‌کند.
۲. مفسری: (Interpreted) کدها خط به خط اجرا می‌شوند که باعث می‌شود عیب‌یابی و تست سریع‌تر انجام شود.
۳. سطح بالا: جزئیات پیچیده سطح پایین مانند مدیریت حافظه را از برنامه‌نویس پنهان می‌کند.
۴. شی‌گرا (Object-Oriented): از برنامه‌نویسی شی‌گرا پشتیبانی می‌کند، اما پارادایم‌های دیگر مانند برنامه‌نویسی تابعی و رویه‌ای نیز در آن قابل استفاده هستند.
۵. پویا (Dynamic Typing): نوع متغیرها در حین اجرا مشخص می‌شود و نیازی به تعریف صریح نوع متغیر نیست.
۶. کتابخانه استاندارد غنی: دارای مجموعه عظیمی از کتابخانه‌های استاندارد است که برای کارهای مختلفی مانند توسعه وب، علم داده، هوش مصنوعی و خودکارسازی در دسترس هستند.
۷. جامعه بزرگ و فعال: جامعه‌ی گسترده‌ای از توسعه‌دهندگان دارد که به پشتیبانی قوی، مستندات فراوان و کتابخانه‌های بسیار دارد.

## کاربردهای اصلی پایتون

- توسعه وب (Backend) : با فریم‌ورک‌هایی مانند **Flask** و **Django**
- علم داده و تحلیل داده: با کتابخانه‌هایی مانند **Pandas, NumPy, Matplotlib**
- یادگیری ماشین و هوش مصنوعی: با کتابخانه‌های محبوبی مانند **Scikit-learn, TensorFlow, PyTorch**
- اسکریپت‌نویسی و اتوماسیون: برای خودکارسازی کارهای تکراری.
- توسعه اپلیکیشن‌های دسکتاپ: با ابزارهایی مانند **Tkinter, PyQt**
- برنامه‌نویسی آموزشی: به دلیل سادگی، یکی از محبوب‌ترین زبان‌ها برای تدریس مفاهیم برنامه‌نویسی است

پایتون می‌تواند برای توسعه انواع مختلف نرم‌افزارها و ابزارها استفاده شود. در اینجا دسته‌بندی کاملی از نرم‌افزارهای قابل توسعه در بستر پایتون ارائه می‌دهیم:

توسعه وب (Web Development)	مثال‌های واقعی
فریم‌ورک‌های Backend:	
- Django - فریم‌ورک کامل و سطح بالا	
- Flask - میکروفریم‌ورک سبک‌وزن	
- FastAPI - برای ساخت API های مدرن و سریع	
- Pyramid - فریم‌ورک انعطاف‌پذیر	
	<b>مثال‌های واقعی</b>
	- Instagram (با Django ساخته شده)
	- Pinterest - پلتفرم اشتراک تصویر
	- Dropbox - سرویس ذخیره‌سازی ابری
	- Spotify - برای سرویس‌های بک‌اند

علم داده و تحلیل (Data Science & Analytics)
نرم افزارهای تحلیلی:
- Jupyter Notebook - محیط تعاملی برای تحلیل داده
- Apache Spark (با PySpark) - پردازش داده های بزرگ
- Pandas GUI - ابزارهای تحلیل داده با رابط گرافیکی

هوش مصنوعی و یادگیری ماشین
ابزارهای ML/AI:
- TensorFlow - کتابخانه جامع یادگیری ماشین
- PyTorch - برای تحقیقات هوش مصنوعی
- scikit-learn - کتابخانه کلاسیک ML
- Keras - رابط سطح بالا برای شبکه های عصبی

توسعه بازی (Game Development)	بازی های معروف:
فریم ورک های بازی سازی:	
- Pygame - برای بازی های دو بعدی	- EVE Online (بخش هایی با پایتون)
- Arcade - کتابخانه مدرن بازی سازی	- Civilization IV - بازی استراتژیک
- D3Panda - برای بازی های سه بعدی	- Battlefield ۲ - برای اسکریپت نویسی

اپلیکیشن‌های دسکتاپ	<b>مثال‌ها:</b> - Dropbox Client - کلاینت دسکتاپ - MusicBee - پخش کننده موسیقی - Calibre - مدیریت کتاب‌های الکترونیکی
ابزارهای GUI	
- کتابخانه استاندارد پایتون Tkinter -	
- رابط‌های حرفه‌ای PyQt/PySide -	
- برای اپلیکیشن‌های چندپلتفرمی Kivy -	
- ابزار قدیمی اما قدرتمند wxPython -	

ابزارهای سیستم و اتوماسیون	اپلیکیشن‌های موبایل
نرم‌افزارهای سیستمی:	فریم‌ورک‌های موبایل:
Ansible - ابزار اتوماسیون DevOps	- Kivy - برای اپ‌های چندپلتفرمی
SaltStack مدیریت پیکربندی	- BeeWare - ساخت اپلیکیشن‌های native
- OpenStack - پلتفرم رایانش ابری	
- Blender - نرم‌افزار مدل‌سازی سه بعدی (اسکرپت‌نویسی)	

ابزارهای شبکه و امنیت	نرم‌افزارهای چندرسانه‌ای
نرم‌افزارهای امنیتی:	ابزارهای صوتی/تصویری:
- آنالیزگر بسته‌های شبکه Scapy -	- دانلود ویدئو از YouTube-DL - پلتفرم‌ها
- ابزار تست نفوذ Sqlmap - SQL injection	- ابزارهای Spotify-Downloader - دانلود موسیقی
- (ماژول‌های پایتون) Metasploit - فریم‌ورک تست نفوذ	- ویرایش ویدئو MoviePy -

نرم افزارهای علمی و آموزشی	نرم افزارهای تجاری و سازمانی
ابزارهای آموزشی:	سیستم های سازمانی:
Anaconda - توزیع پایتون برای علم داده	ERP - نرم افزار جامع Odoo -
SageMath - نرم افزار ریاضی	- پلتفرم تجاری Tryton -
Minecraft: Pi Edition - نسخه آموزشی	- سیستم برنامه ریزی منابع ERPNext - سازمانی

## ویژگی های مشترک این نرم افزارها

✓ چند پلتفرمی: اجرا روی ویندوز، لینوکس، مک

✓ متن باز: بسیاری رایگان و قابل توسعه هستند

✓ مقیاس پذیر: از پروژه های کوچک تا سیستم های enterprise

✓ جامعه فعال: پشتیبانی قوی و مستندات کامل

- پایتون به دلیل سادگی، کتابخانه های گسترده و جامعه فعال، بستر ایده آلی برای توسعه انواع نرم افزارها محسوب می شود.
- پایتون می تواند برای توسعه انواع مختلف نرم افزارها و ابزارها استفاده شود. در اینجا دسته بندی کاملی از نرم افزارهای قابل توسعه در بستر پایتون ارائه می دهیم:
- پایتون می تواند برای توسعه انواع مختلف نرم افزارها و ابزارها استفاده شود. در اینجا دسته بندی کاملی از نرم افزارهای قابل توسعه در بستر پایتون

## بسترهای پیاده‌سازی پایتون

پایتون در بسترهای مختلفی قابل پیاده‌سازی و اجراست که هر کدام کاربردهای خاص خود را دارند:

### محیط‌های اجرا (Runtime Environments)

#### ۱. CPython (رایج‌ترین)

- پیاده‌سازی اصلی و استاندارد پایتون
- نوشته شده با C
- شامل مفسر و کتابخانه استاندارد

#### ۲. PyPy

- پیاده‌سازی با مفسر JIT (Just-In-Time)
- سرعت اجرای بالاتر نسبت به CPython
- سازگار با CPython

#### ۳. Jython

- پیاده‌سازی پایتون روی پلتفرم JVM
- قابلیت استفاده از کتابخانه‌های جاوا
- مناسب برای محیط‌های سازمانی

#### ۴. IronPython

- پیاده‌سازی برای پلتفرم .NET
- یکپارچه‌سازی با زبان‌های داتنت
- مناسب برای توسعه ویندوز

## ۵. MicroPython

- برای سیستم‌های تعبیه‌شده و میکروکنترلرها
- سبک‌وزن و بهینه‌شده
- مناسب برای IoT و رزبری پای

## محیط‌های توسعه (Development Environments)

### • IDE های تخصصی

- PyCharm (حرفه‌ای - JetBrains)
- VS Code (سبک و قابل گسترش)
- Spyder (مخصوص علم داده)
- Thonny (مبتدی‌پسند)

### ویرایشگرهای کد

- Sublime Text
- Atom
- Vim/Neovim
- Emacs

## پلتفرم‌های ابری و دیپلوی

### • پلتفرم‌های ابری

### سرویس‌های ابری محبوب

- AWS\*\* (Lambda, EC2, Beanstalk)
- Google Cloud\*\* (App Engine, Cloud Functions)
- Microsoft Azure\*\* (App Service, Functions)
- \*\*Heroku\*\* - ساده برای استارت‌آپ‌ها
- \*\*PythonAnywhere\*\* - میزبانی تخصصی پایتون



## پلتفرم‌های توزیع و بسته‌بندی

### ابزارهای بسته‌بندی

- **PyInstaller** اجرای کردن برنامه
- **cx\_Freeze** ایجاد فایل اجرایی
- **Py2exe** مخصوص ویندوز
- **Py2app** مخصوص مک

## پلتفرم‌های خاص کاربردی

### وب سرورها

- WSGI: Gunicorn, uWSGI, mod\_wsgi
- ASGI: Uvicorn, Hypercorn, Daphne

### پلتفرم‌های خاص

- Raspberry Pi - پروژه‌های سخت‌افزاری
- AWS IoT - اینترنت اشیاء
- Google Colab - محاسبات ابری رایگان
- Jupyter Hub - سرویس نوت‌بوک سازمانی

## پلتفرم‌های استریم و پردازش واقعی

### پردازش جریان داده

- Apache Spark (PySpark)
- Apache Flink (PyFlink)
- Kafka (kafka-python)

## پلتفرم‌های موبایل و embedded

### موبایل

- BeeWare (اپلیکیشن‌های native)
- Kivy (اپلیکیشن‌های چندپلتفرمی)
- React Native (با پشتیبانی پایتون)

### انتخاب بستر مناسب

نیاز	بستر پیشنهادی
یادگیری و توسعه عمومی	CPython + VS Code
کاربردهای سازمانی	Jython + IDE حرفه‌ای
دیتا ساینس	Anaconda + Jupyter
توسعه وب	CPython + Django/Flask
سیستم‌های embedded	MicroPython
اپلیکیشن‌های دسکتاپ	CPython + PyQt
سرورهای پرتراфик	PyPy + Async Frameworks

- هر بستر مزایا و معایب خاص خود را دارد و انتخاب آن بستگی به نیاز پروژه، مقیاس‌پذیری مورد نظر و منابع در دسترس دارد.