



چرا پایتون؟

چرا برنامه نویسی؟

رایانه یا کامپیوتر

Computer = compute + er → پسوند فاعل ساز

↓
محاسبه به واسطه ماشین



واژه کامپیوتر در سال ۱۶۴۶ میلادی به فرهنگ انگلیسی آکسفورد وارد گردید که به معنی شخصی که محاسبه می‌کند بوده است



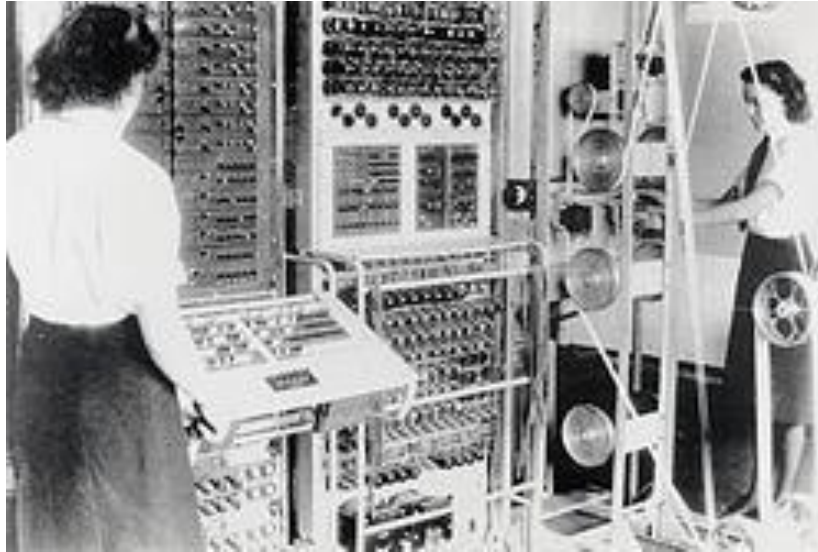
یک رایانه انسانی با تلسکوپ و ماشین حساب در سال ۱۹۵۲. در گذشته زن‌ها به علت عدم توازن درآمدی بین جنسیت‌ها به‌عنوان رایانه انسانی به‌طور گسترده استخدام می‌شدند.

معنای واژه فارسی رایانه

واژه رایانه از مصدر **رایانیدن** ساخته شده



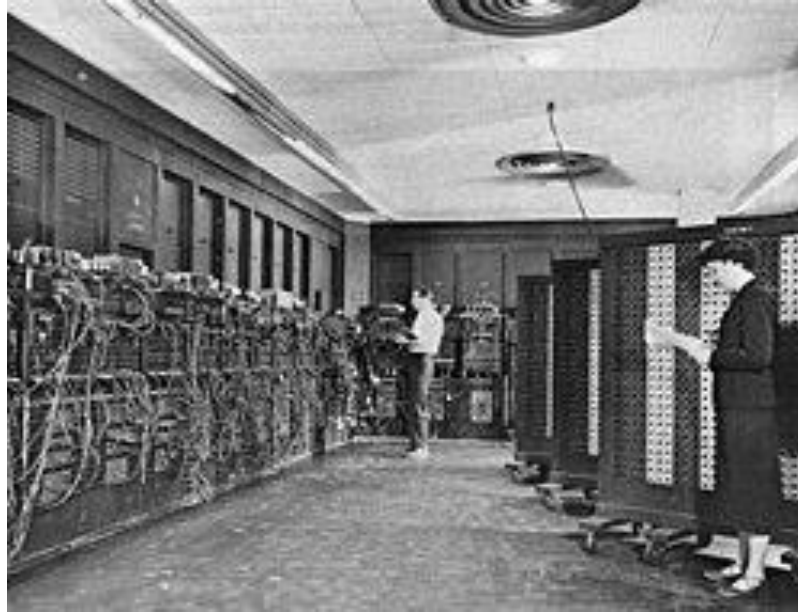
مرتب کردن، نظم بخشیدن و سامان دادن چیزی



Colossus

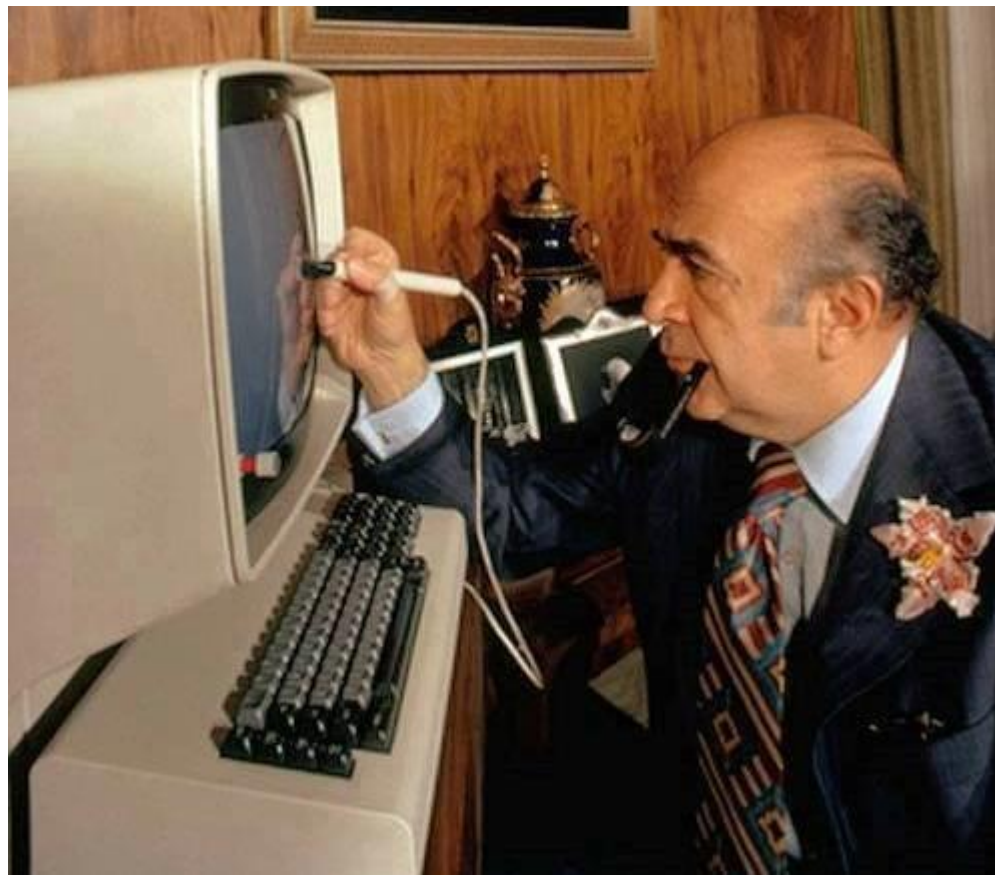
کلوسوس، اولین دستگاه محاسباتی دیجیتال الکترونیکی قابل برنامه‌ریزی، برای شکستن رمزهای آلمانی در طول جنگ جهانی دوم استفاده می‌شد. این دستگاه در سال ۱۹۴۳ در بلچلی پارک در حال استفاده دیده می‌شود.

ENIAC

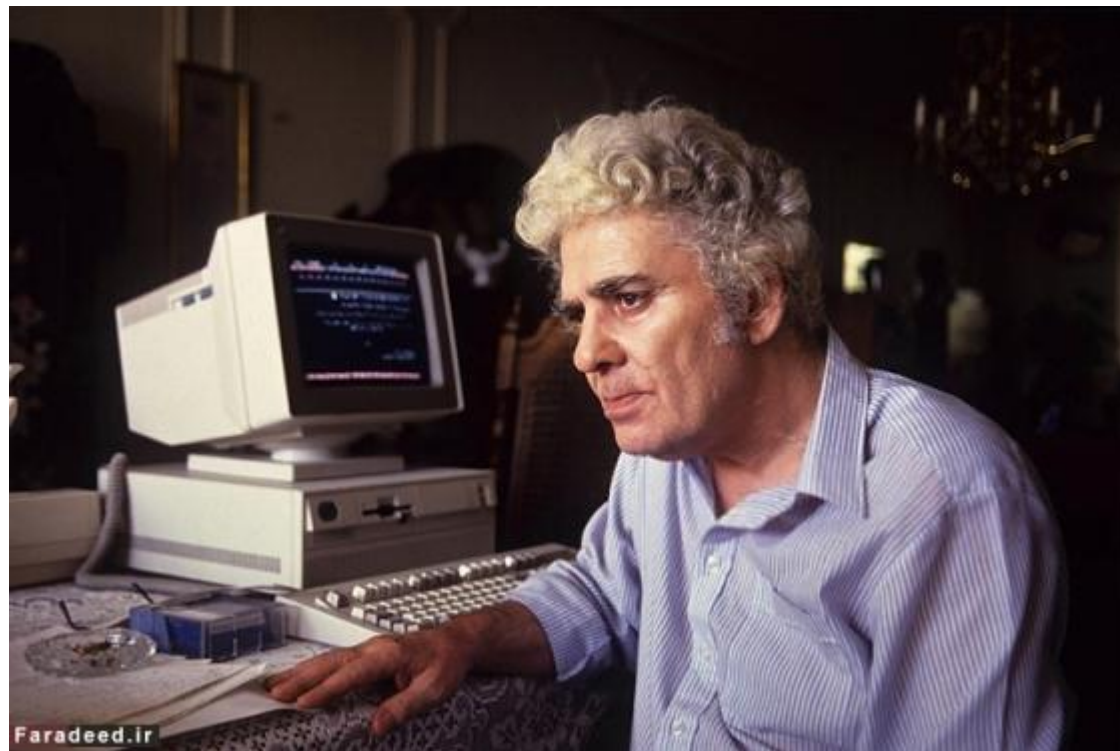


انیاک اولین دستگاه الکترونیکی و کامل از نظر تورینگ بود که محاسبات مسیر
بالستیک را برای ارتش ایالات متحده انجام می داد.

اولین کامپیوتر در ایران



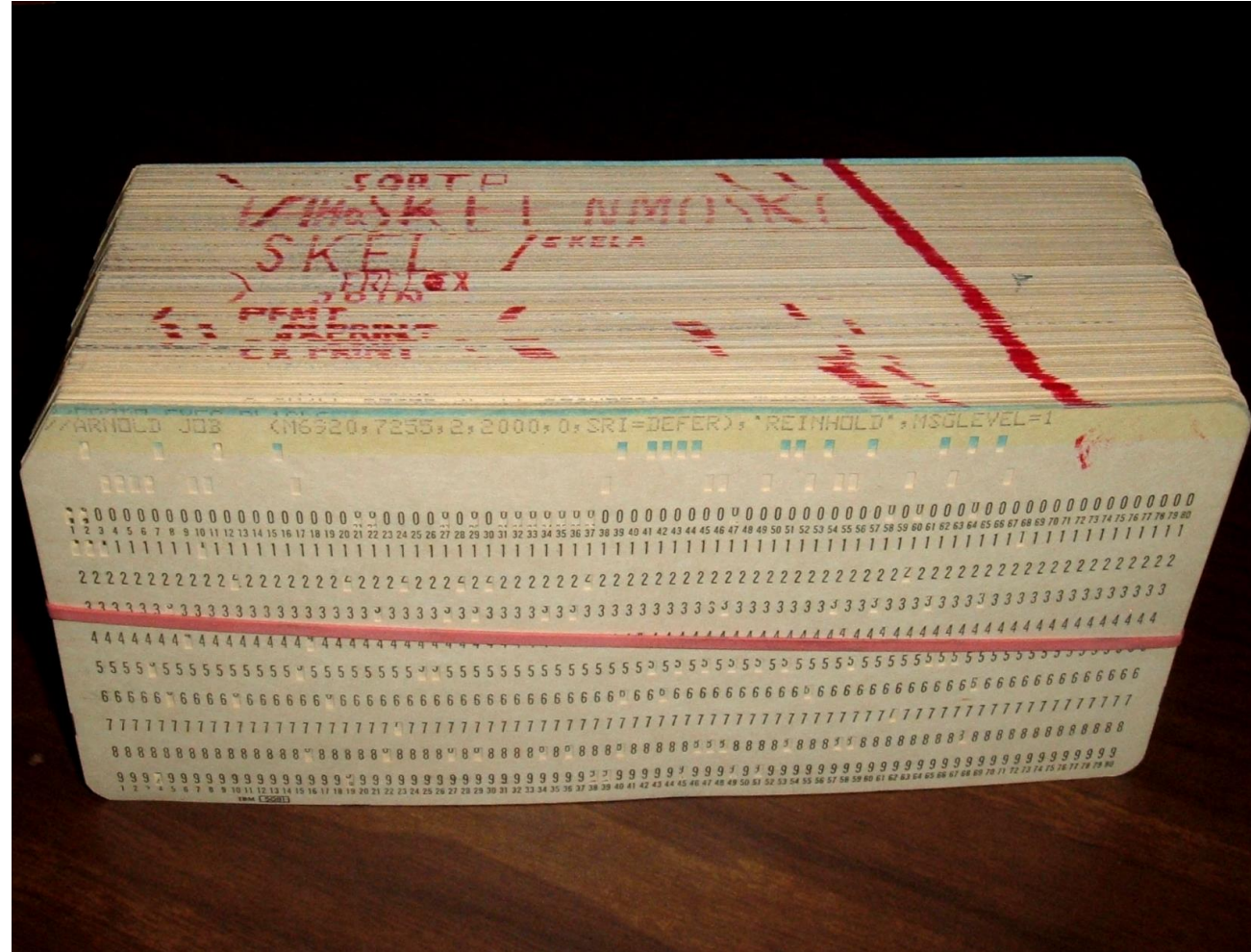
1354



نیمه دوم دهه ۶۰



برنامه‌نویسان دانشجو در دانشگاه فنی آخن در آلمان غربی در سال ۱۹۷۰ در حال استفاده از دستگاه‌های پانچ کلید IBM 026 بودند.



A card deck containing one program.



Cards would sometimes jam in the reader, requiring one or more to be repunched.

Evolution of the Computer



تکامل زبان‌ها

محاسبه مجموع اعداد ۱ تا ۱۰

زبان ماشین (خیلی پیچیده است، فقط نمایش کوتاه کد):

```
MOV AX, 0
MOV BX, 1
ADD_LOOP:
ADD AX, BX
INC BX
CMP BX, 11
JL ADD_LOOP
```

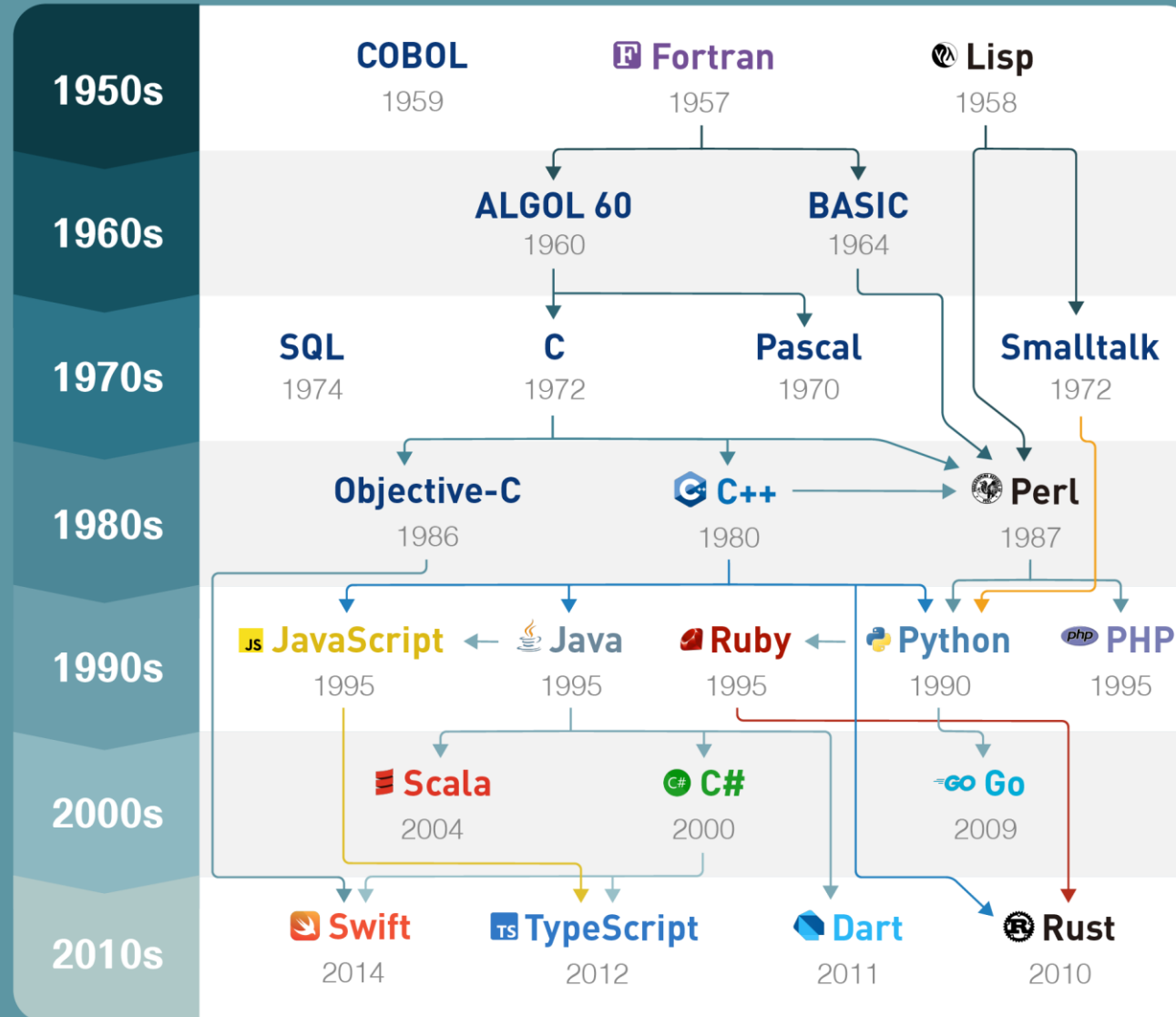
زبان C:

```
#include <stdio.h>
int main() {
    int sum = 0;
    for (int i = 1; i <= 10; i++) {
        sum += i;
    }
    printf("Sum: %d\n", sum);
    return 0;
}
```


Python:

```
sum = sum(range(1, 11))  
print(f"Sum: {sum}")
```

Timeline of Programming Languages



روند تکامل تعامل انسان با کامپیوتر

- وارد کردن اعداد صفر و یکی
- زبان های برنامه نویسی سطح پایین
- زبان های برنامه نویسی سطح بالا
- ؟

Chat GPT

Generative Ai

Will the skill of programming or coding become irrelevant due to LLM like Chat GPT or AI?

Will LLMs make programming languages obsolete?

In This Era of AI, Will Everyone Be a Programmer?

Do developers still need to learn programming languages in the age of AI?

Answer from quora

It is unlikely that the skill of programming or coding will become irrelevant due to LLMs like ChatGPT or AI. While these technologies can automate some programming tasks, they will not be able to replace the need for human programmers for the foreseeable future.

Here are some reasons why:

- LLMs are still under development and **have limitations**. They can only generate code that is similar to code that they have been trained on. They cannot come up with new and innovative solutions to programming problems.
- LLMs are not able to understand the context of a programming problem. They need to be given explicit instructions on what to do. They cannot understand the nuances of a programming problem and come up with creative solutions.
- LLMs are **not able to debug code**. They can only generate code. They cannot find and fix errors in code.

As a result, human programmers will still be needed to:

- **Design and architect software systems.** LLMs can generate code, but they cannot design and architect software systems. This requires human understanding of the problem domain and the ability to come up with creative solutions.
- **Debug code.** LLMs can only generate code. They cannot find and fix errors in code. This requires human understanding of the code and the ability to think logically.
- **Innovate.** LLMs can only generate code that is similar to code that they have been trained on. They cannot come up with new and innovative solutions to programming problems. This requires human creativity and ingenuity.

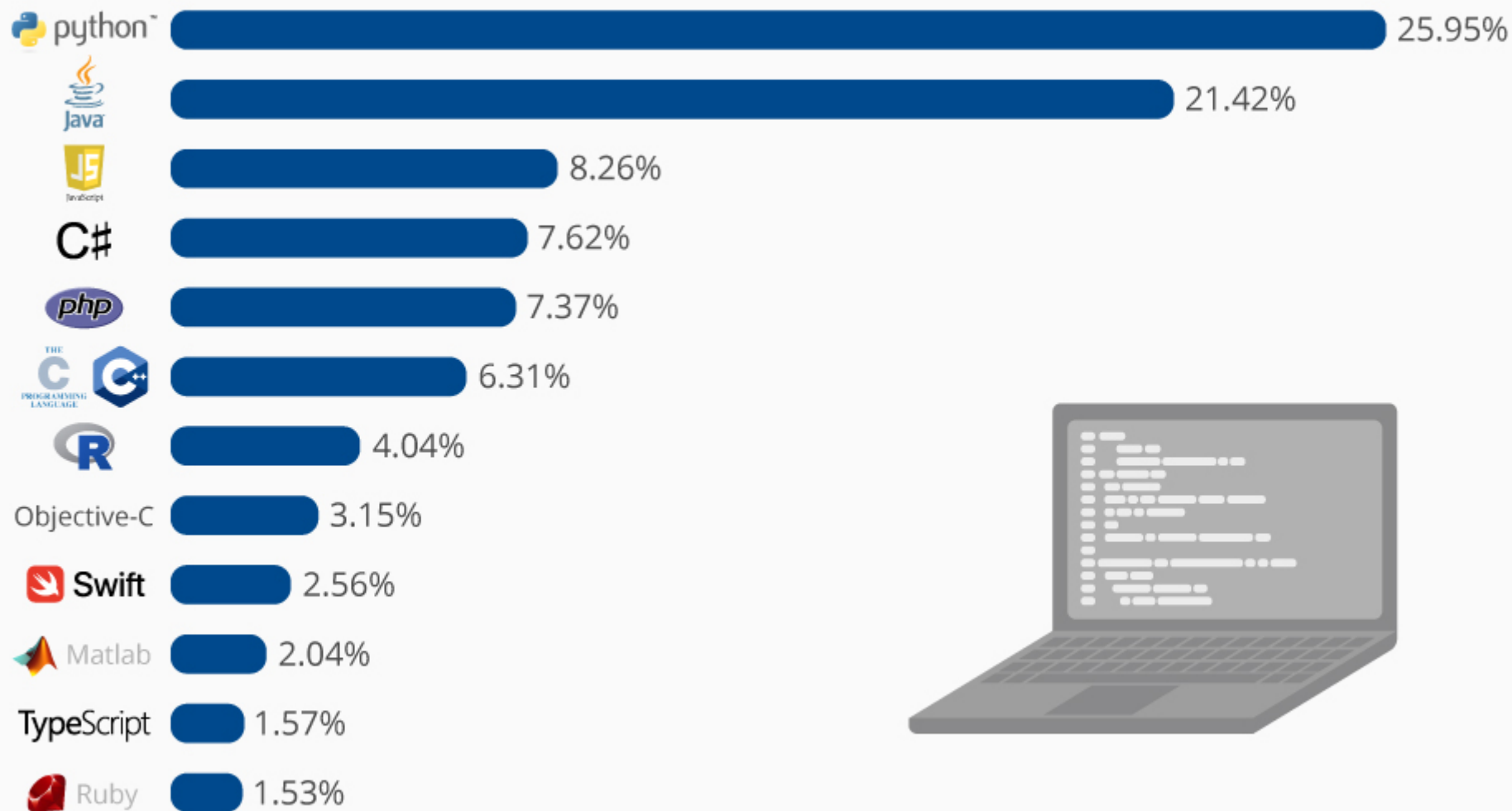
بنابراین، حتی در دورانی که هوش مصنوعی نقش مهمی ایفا می کند، **یادگیری**
برنامه نویسی همچنان یک مهارت ارزشمند و ضروری است که به شما امکان
می دهد در دنیای فناوری پیشرو باشید.

چرا پایتون ؟



The Most Popular Programming Languages

Share of the most popular programming languages in the world*

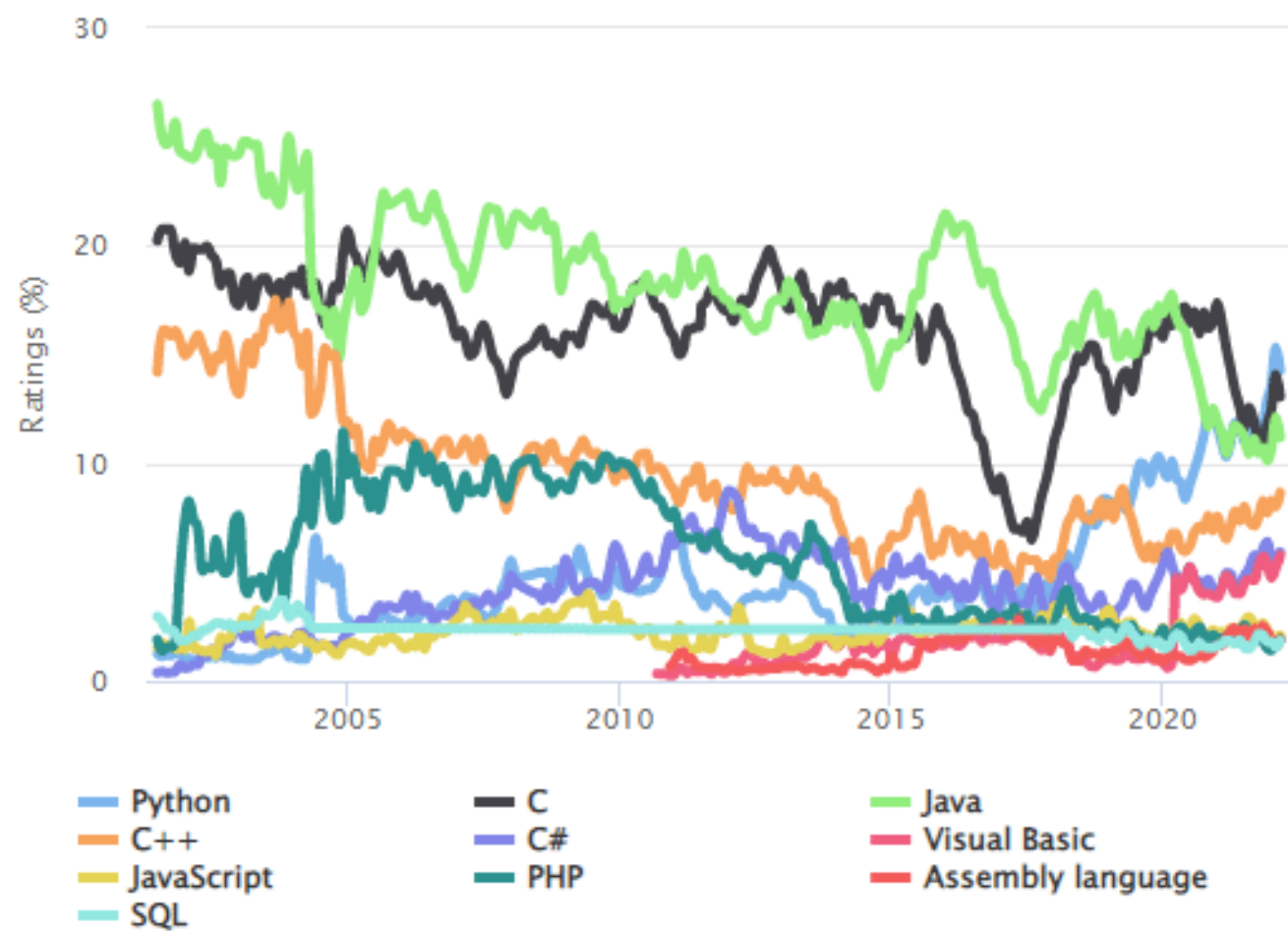


@StatistaCharts

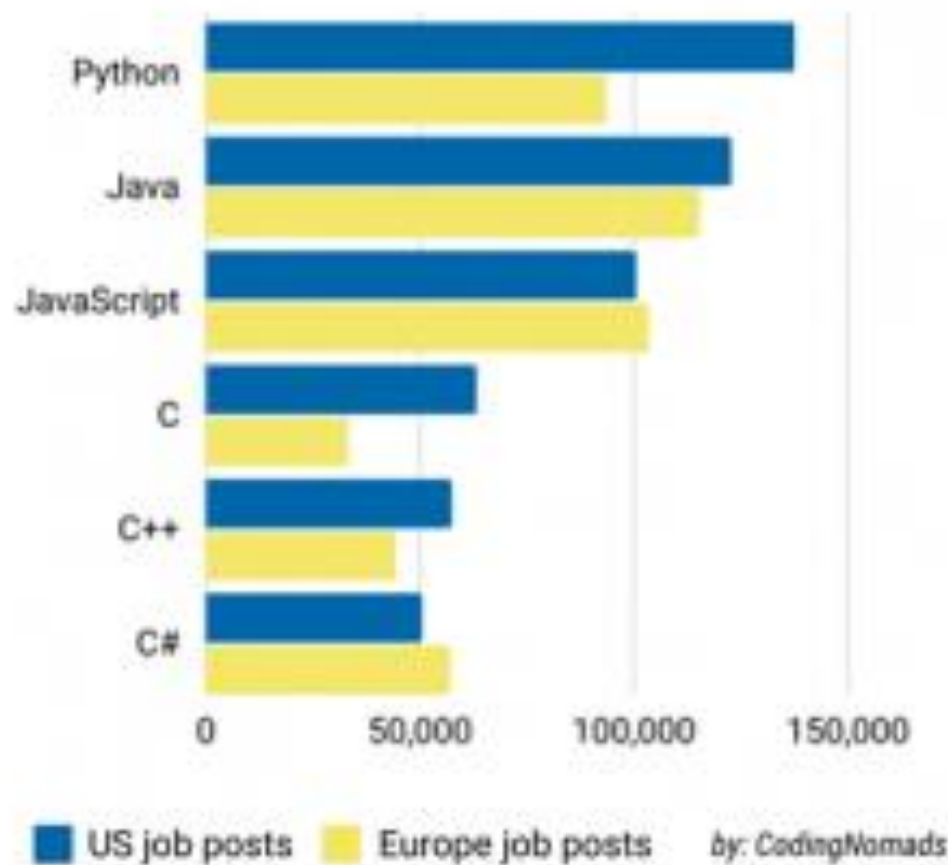
* Based on the PYPL-Index, an analysis of Google search trends for programming language tutorials.

Source: PYPL

statista

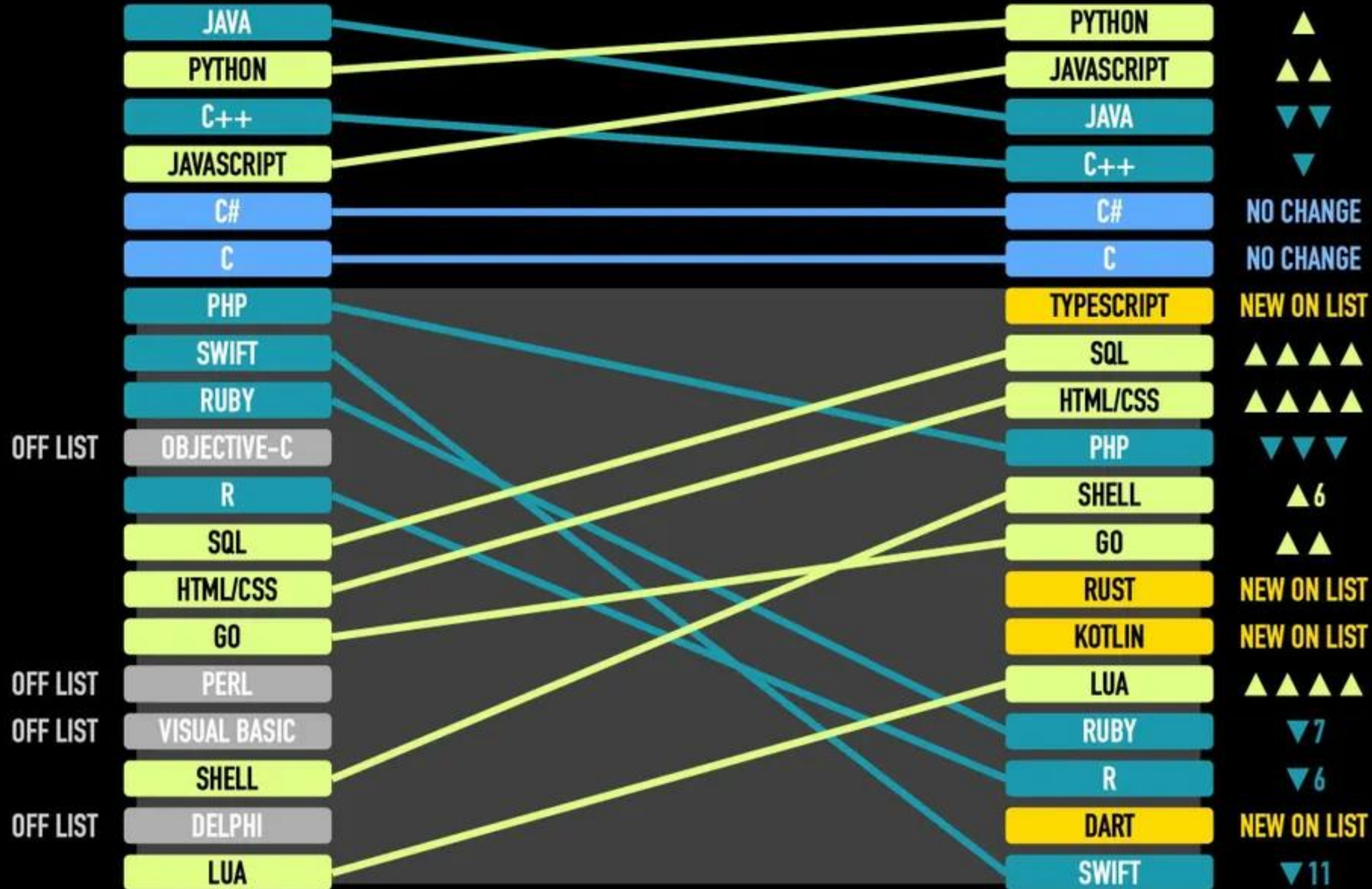


Most in-demand programming languages 2021-2022



PROGRAMMING LANGUAGE POPULARITY – 2016 vs 2024

2016



2024

میانگین درآمد سالانه‌ی یک برنامه نویس پایتون

ایالات متحده آمریکا:

میانگین سالانه: حدود ۱۱۲،۳۸۲ دلار در سال ۲۰۲۴.

انگلستان:

میانگین سالانه: حدود ۵۰،۰۰۰ پوند.

آلمان:

میانگین سالانه: بین ۵۰،۰۰۰ تا ۷۰،۰۰۰ یورو.

چرا برای برنامه نویسان تازه کار یادگیری پایتون بهتر از دیگر
زبان های برنامه نویسی است ؟

۱. سادگی و خوانایی کد

طراحی ساده و سینتکس نزدیک به زبان طبیعی (انگلیسی)

```
# پایتون  
print("Hello, World!")
```

```
// جاوا  
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
}
```

۲. پشتیبانی گسترده از منابع آموزشی

۳. کتابخانه‌ها و ابزارهای متنوع

۴. کاربرد گسترده در دنیای واقعی

۵. توسعه سریع

۶. مدیریت ساده‌تر خطاها

منکه با کامپیوتر کار می کنم
چه نیازی است برنامه نویسی یاد بگیرم ؟

مرتب‌سازی و دسته‌بندی فایل‌ها

یک کارمند ممکن است در پایان هر روز نیاز داشته باشد فایل‌های مختلف را بر اساس نوعشان (PDF، Excel، عکس و غیره) در پوشه‌های جداگانه ذخیره کند.

```
import os
import shutil

# مسیر فایل‌ها
source_dir = "C:/Users/Files"
destination_dirs = {
    ".pdf": "C:/Users/Sorted_Files/PDFs",
    ".xlsx": "C:/Users/Sorted_Files/Excels",
    ".jpg": "C:/Users/Sorted_Files/Images"
}

# مرتب‌سازی
for file in os.listdir(source_dir):
    ext = os.path.splitext(file)[1]
    if ext in destination_dirs:
        shutil.move(os.path.join(source_dir, file), destination_dirs[ext])
print("فایل‌ها مرتب شدند")
```

جمع‌آوری اطلاعات از وبسایت‌ها

کارمندان ممکن است نیاز داشته باشند اطلاعاتی مانند قیمت‌ها، اخبار، یا داده‌های خاص را از وبسایت‌ها جمع‌آوری کنند.

```
import requests
from bs4 import BeautifulSoup

# دانلود داده‌ها
url = "https://example.com/news"
response = requests.get(url)
soup = BeautifulSoup(response.text, 'html.parser')

# استخراج اخبار
news = soup.find_all("h2", class_="headline")
for item in news:
    print(item.text)
```

پردازش خودکار فایل‌های Excel

اگر کارمند روزانه نیاز داشته باشد داده‌ها را در فایل‌های Excel ویرایش یا پردازش کند، این کار به راحتی با پایتون انجام می‌شود.

```
import pandas as pd

# خواندن فایل
data = pd.read_excel("sales_data.xlsx")

# پردازش: محاسبه مجموع فروش هر دسته
summary = data.groupby("Category")["Sales"].sum()

# ذخیره خروجی
summary.to_excel("sales_summary.xlsx")
print("خلاصه فروش ذخیره شد")
```

ایجاد اسناد Word از قالب‌ها

کارمندان ممکن است نیاز داشته باشند نامه‌ها یا اسناد مشابهی ایجاد کنند که تنها بخش‌هایی از آنها تغییر می‌کند. این فرآیند می‌تواند با قالب‌های از پیش تعریف‌شده اتوماتیک شود.

```
from docx import Document

# ایجاد سند از قالب
template = Document("template.docx")
template.paragraphs[0].text = "نام: علی"
template.paragraphs[1].text = "تاریخ: ۱۴۰۳/۱۰/۲۱"

# ذخیره فایل جدید
template.save("new_document.docx")
print("اسناد ایجاد شد")
```

پشتیبان‌گیری از فایل‌ها

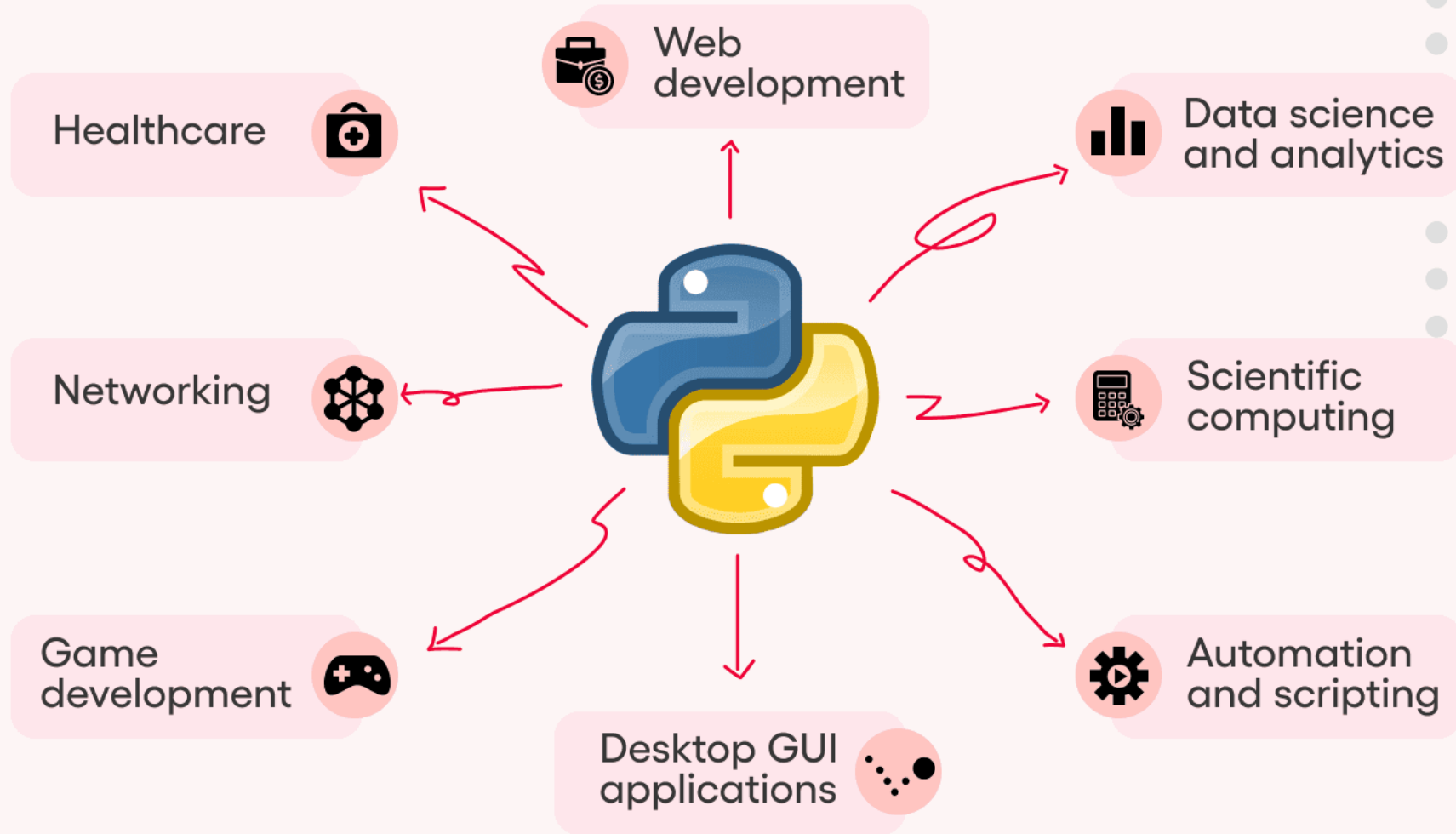
کارمندان ممکن است بخواهند از فایل‌های مهم خود به طور مرتب پشتیبان بگیرند.

```
import shutil
from datetime import datetime

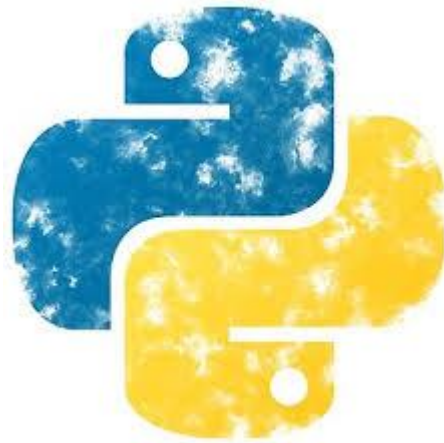
# مسیرهای فایل
source = "C:/Users/Documents"
backup = f"C:/Backup/Backup_{datetime.now().strftime('%Y%m%d_%H%M%S')}"

# ایجاد پشتیبان
shutil.copytree(source, backup)
print("پشتیبان‌گیری انجام شد")
```

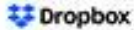

Applications of Python




8 Best Apps Built With Python



















Dropbox

[Why Dropbox?](#)[Features](#)[Plans & pricing](#)

[Support](#)[Contact sales](#)[Sign in](#)[Get started](#)







Store and sync

Share


Stay secure

Sign up

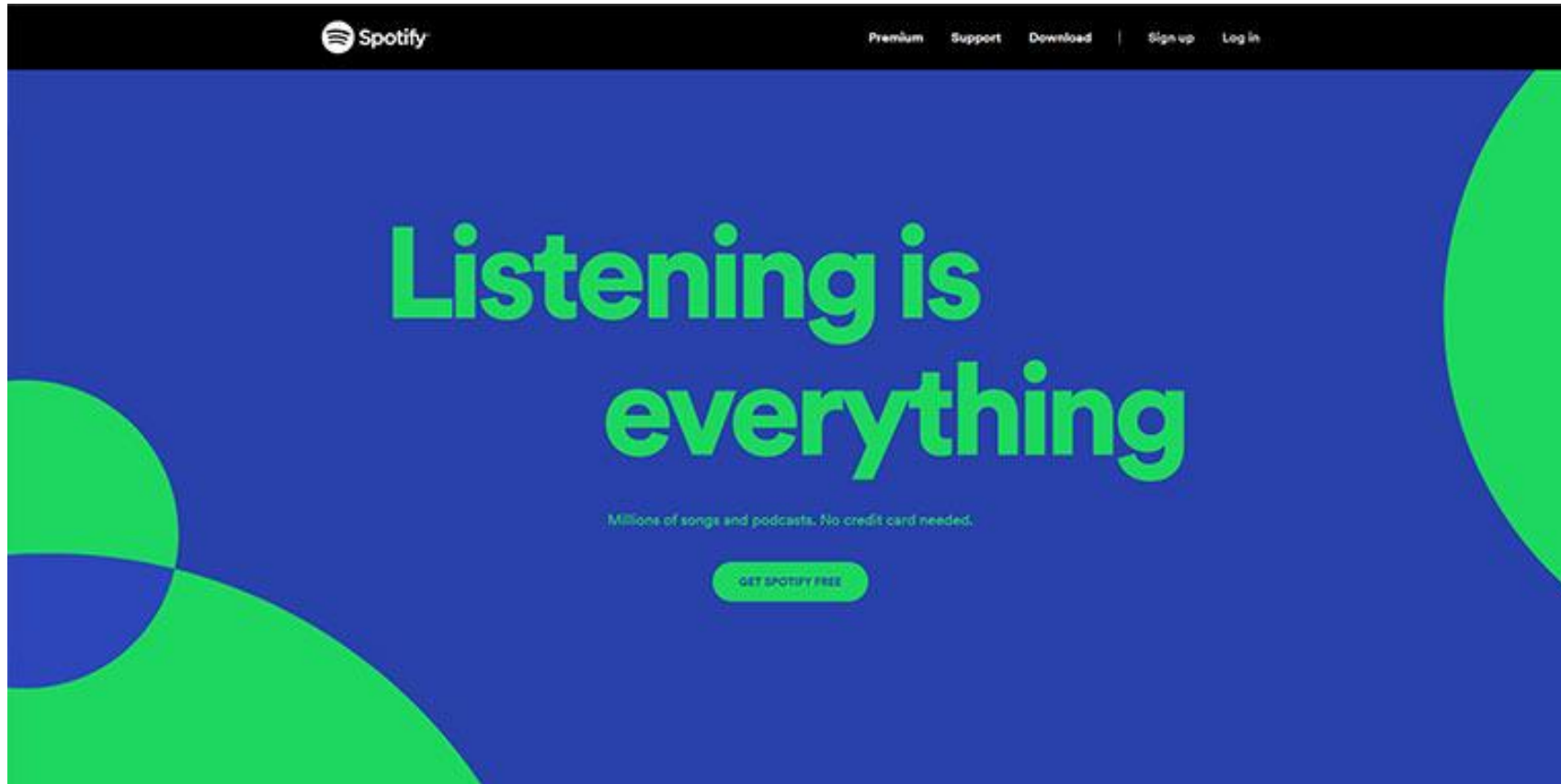
or sign in to your account

☐ I agree to the [Dropbox Terms](#)

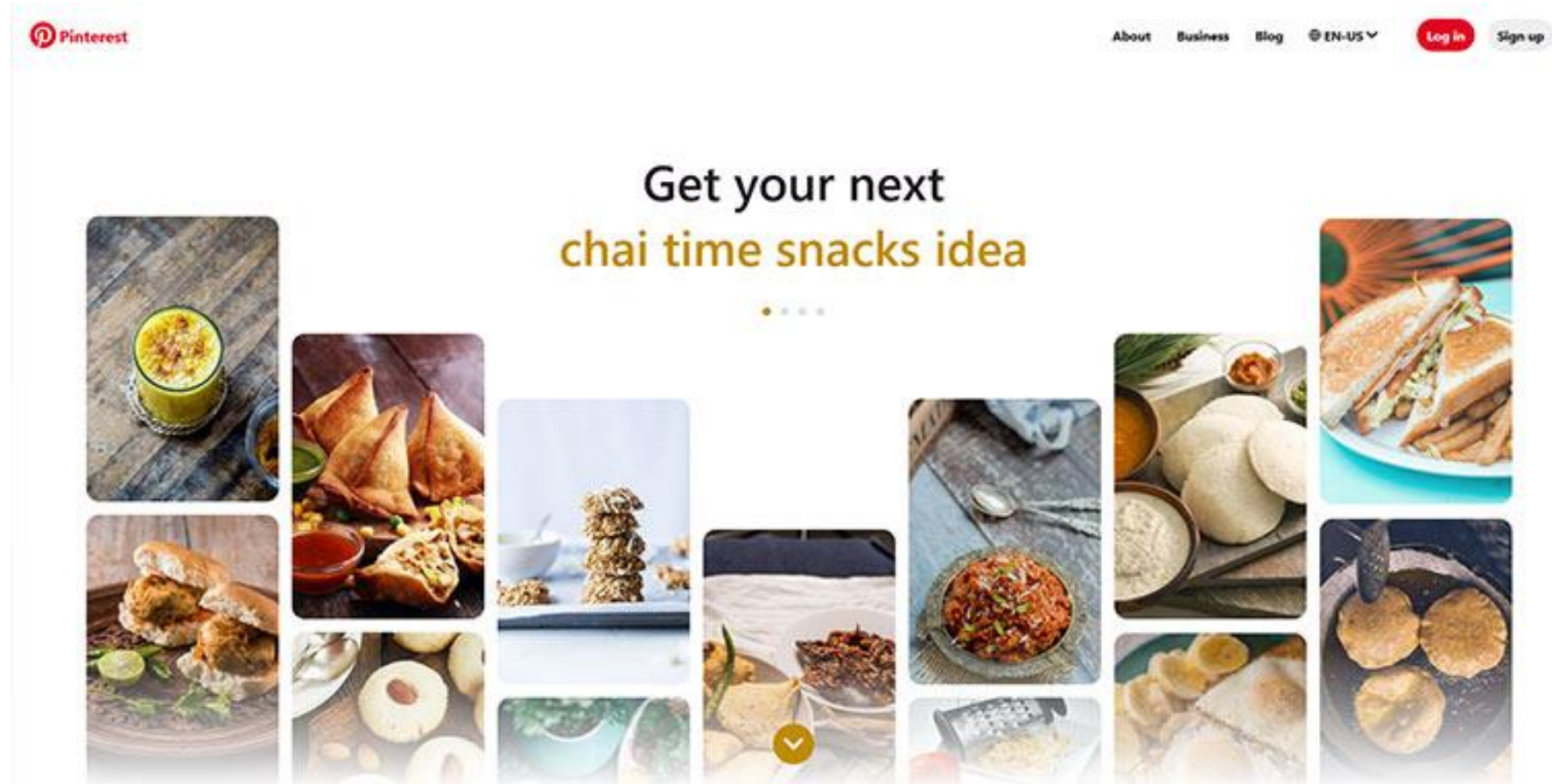
Sign up

 Sign up with Google

Spotify



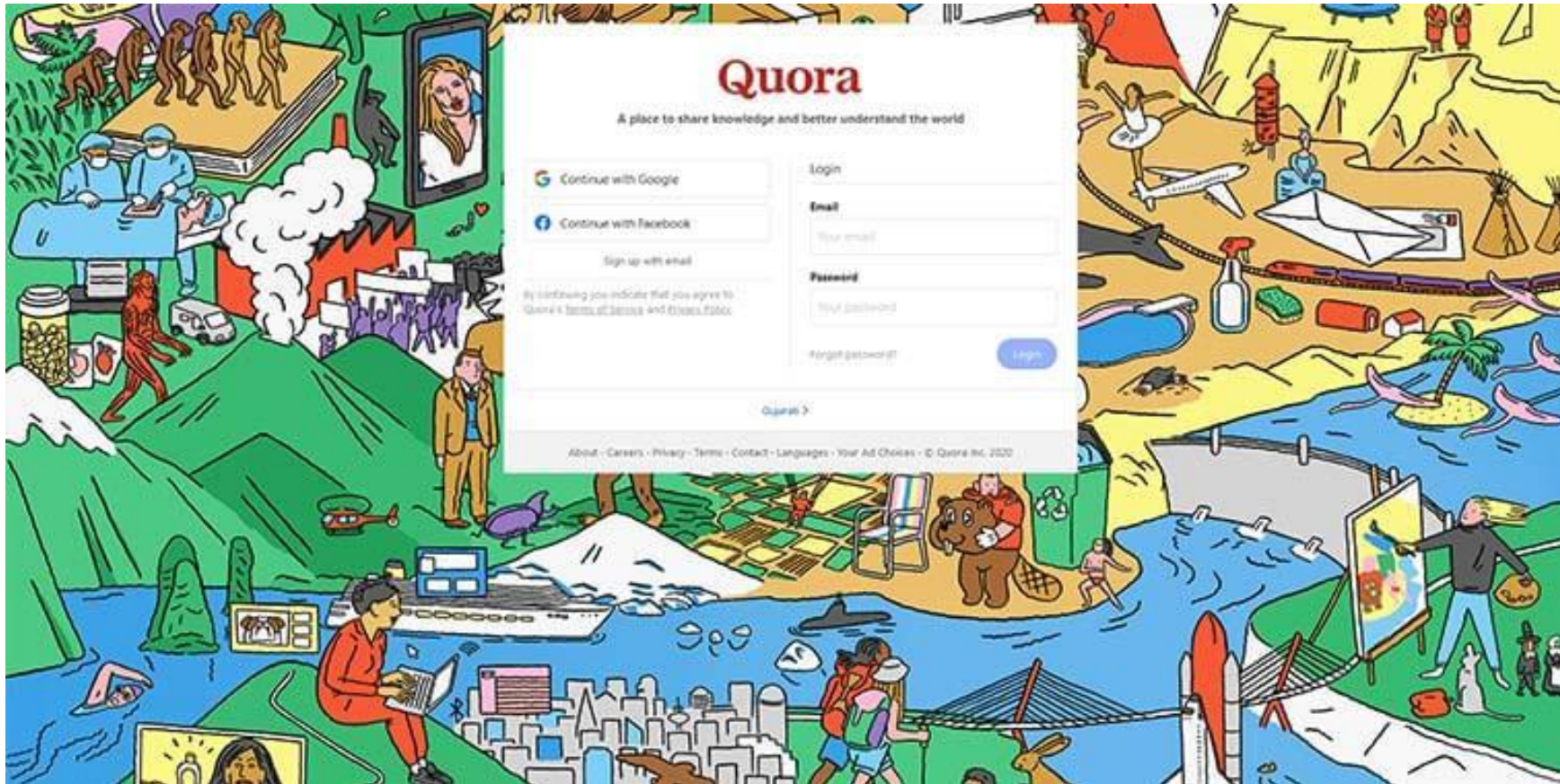
Pinterest



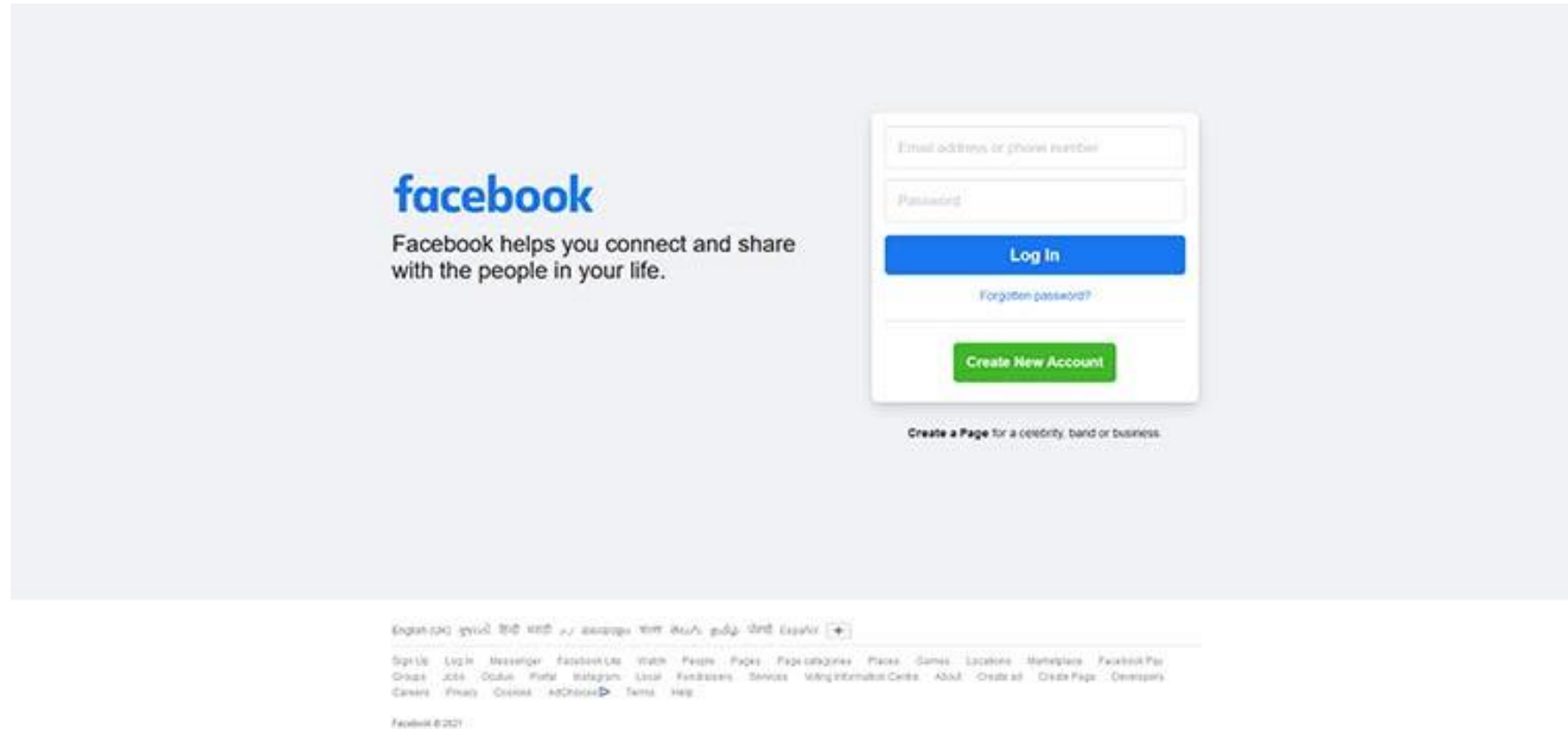
Uber



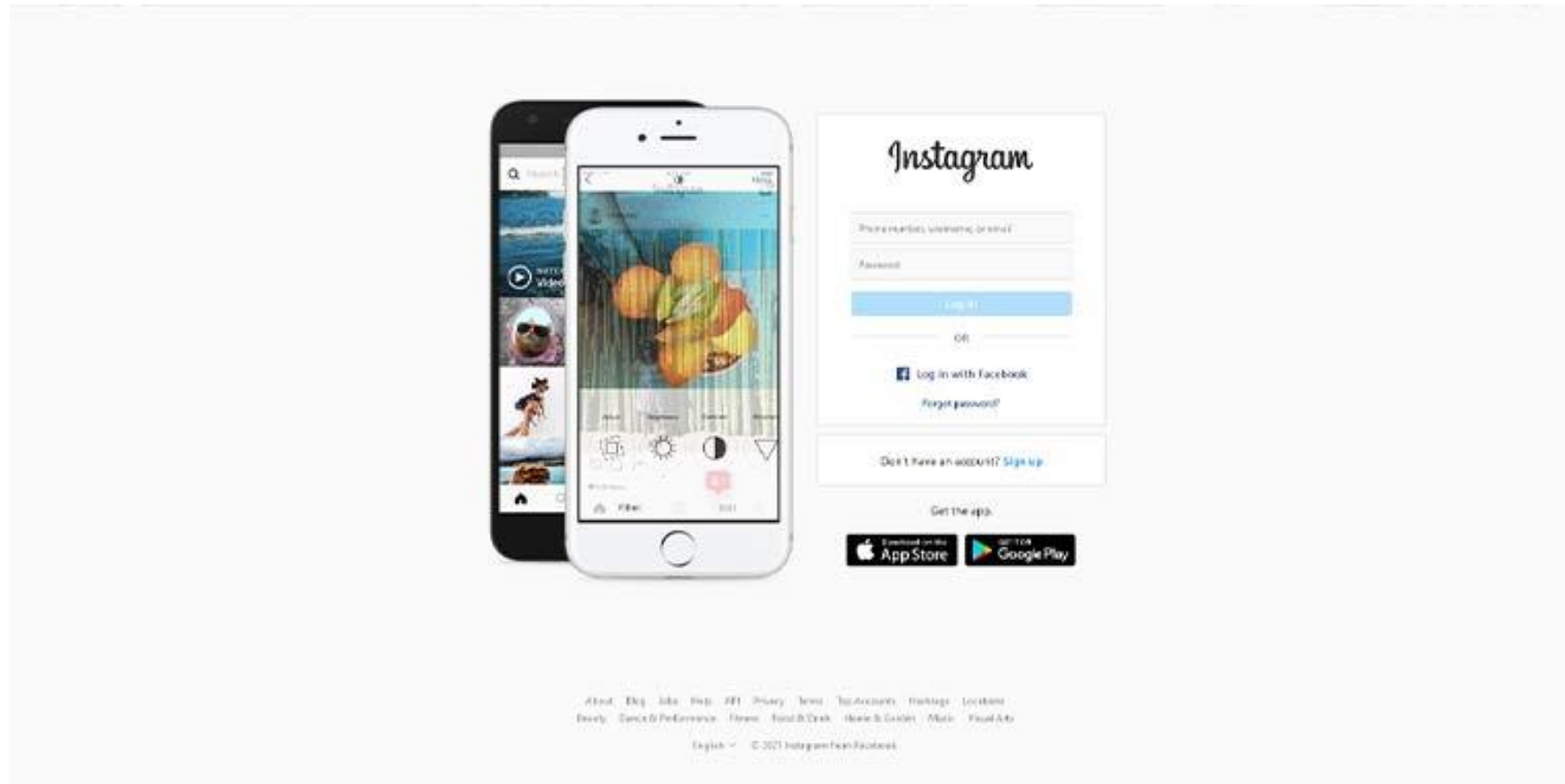
Quora



Facebook



Instagram



Guido van Rossum



2006



2024

British comedy series

Monty Python's Flying Circus



1969 to 1974

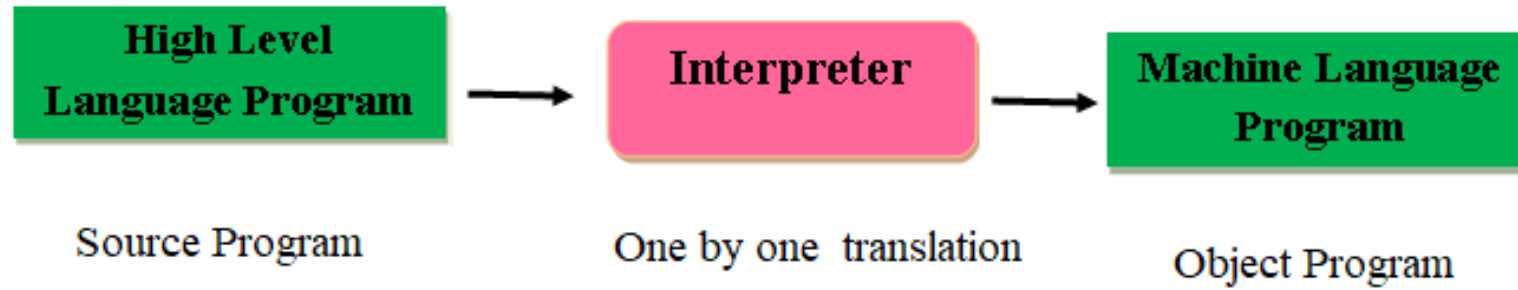
compiler



کامپایلر: برنامه‌ای که **کل** **کد منبع** را به یک زبان سطح پایین‌تر (مانند زبان ماشین) تبدیل می‌کند تا برنامه مستقیماً توسط کامپیوتر اجرا شود. این فرآیند معمولاً یک‌بار انجام می‌شود و سپس برنامه کامپایل شده اجرا می‌شود.

interpreter

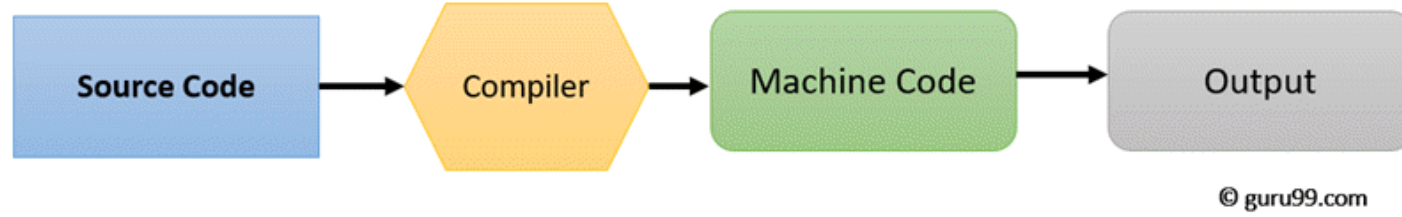
How Interpreter Works



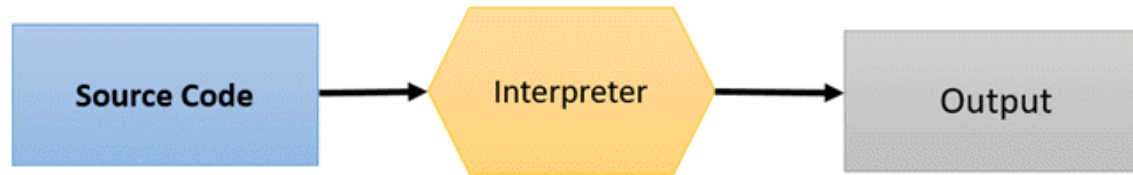
VidyaGyaan.com

مفسر: برنامه‌ای که کد منبع را **خط به خط** (یا دستور به دستور) اجرا می‌کند و هر خط را همان لحظه تفسیر و اجرا می‌کند.

How Compiler Works



How Interpreter Works

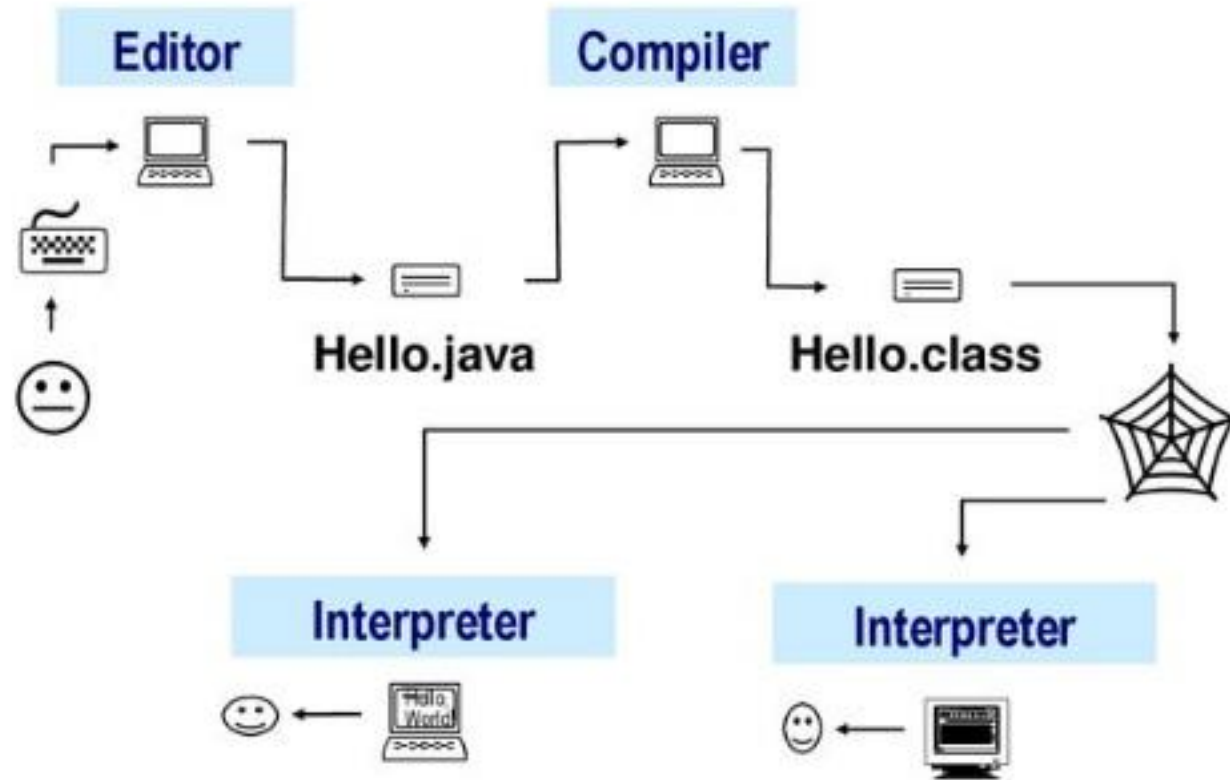


یک مثال روزمره

کامپایلر: مثل اینکه کل کتاب را یکبار ترجمه کنید و سپس به خواننده بدهید.
مفسر: مثل اینکه هر صفحه از کتاب را هنگام مطالعه ترجمه کنید.

ویژگی	کامپایلر	مفسر
نحوه اجرا	تبدیل کل برنامه به یک فایل اجرایی	اجرا خط به خط
سرعت اجرا	سریع‌تر (بعد از کامپایل)	کندتر
خطاها	همه خطاها در زمان کامپایل شناسایی می‌شوند	خطاها هنگام اجرا مشخص می‌شوند
مثال‌ها	C، C++، Go	Python، JavaScript

زبان‌های ترکیبی



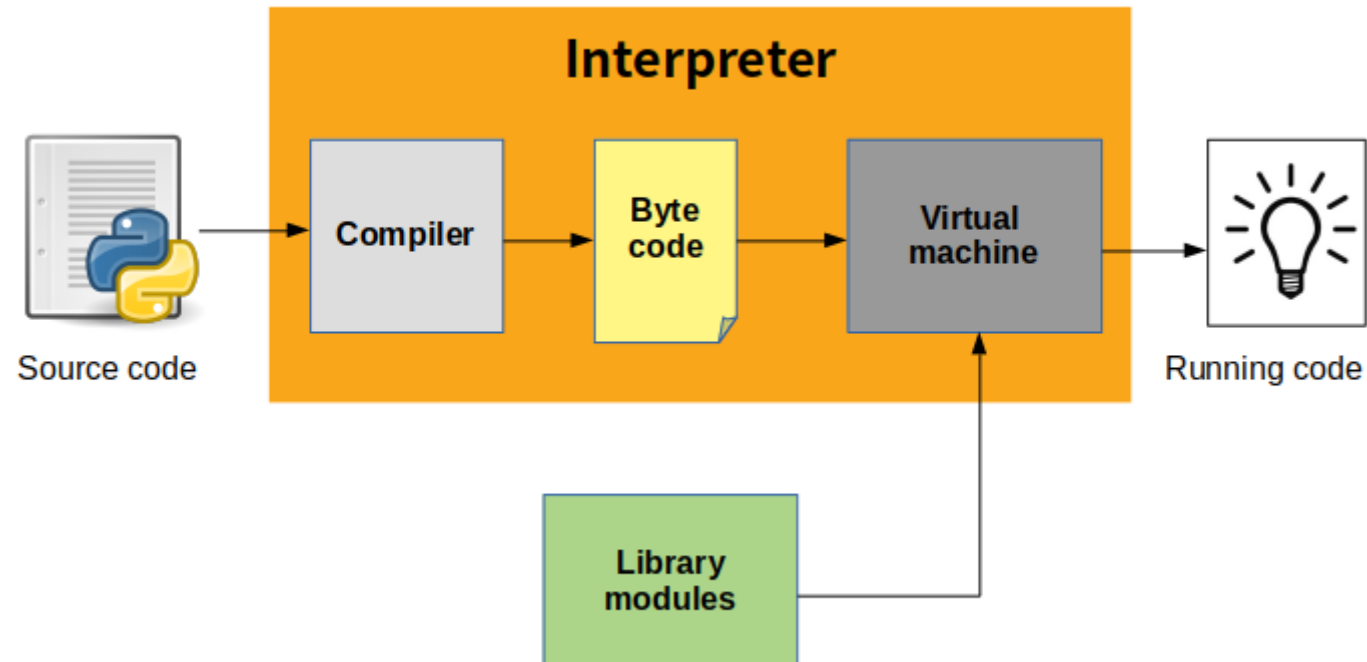
مفہوم زبان ترکیبی

Upfront it is compiled to become a file called Intermediate language, that close to a machine code that can be portable **cross-platform**. The user machine will take the Intermediate language file and do the last step to compile which usually called Just in Time compilation (JIT compilation). The Intermediate language is also called byte-code.

Java, C#

آیا پایتون یک زبان ترکیبی است ؟

کامپایل اولیه: وقتی یک برنامه پایتون اجرا می‌شود، کد ابتدا به بایت کد تبدیل می‌شود. فایل‌هایی با پسوند `.pyc` در پوشه `__pycache__` تفسیر نهایی: بایت کد توسط Python Virtual Machine (PVM) تفسیر و اجرا می‌شود.





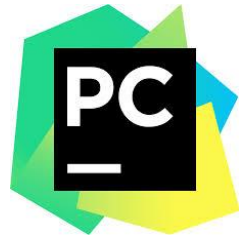
Thonny



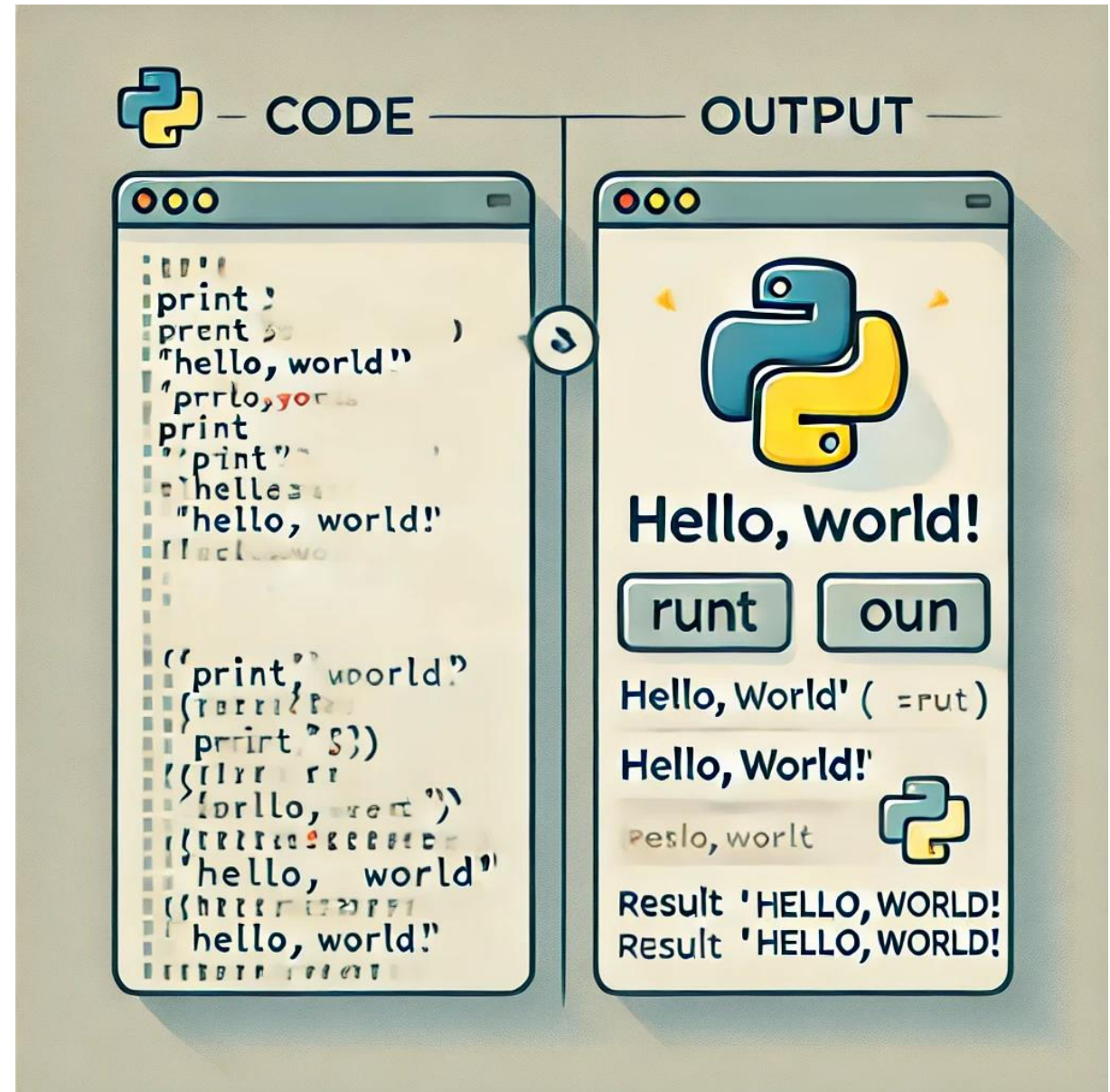
IDLE



PyCharm



Visual Studio Code





About

Downloads

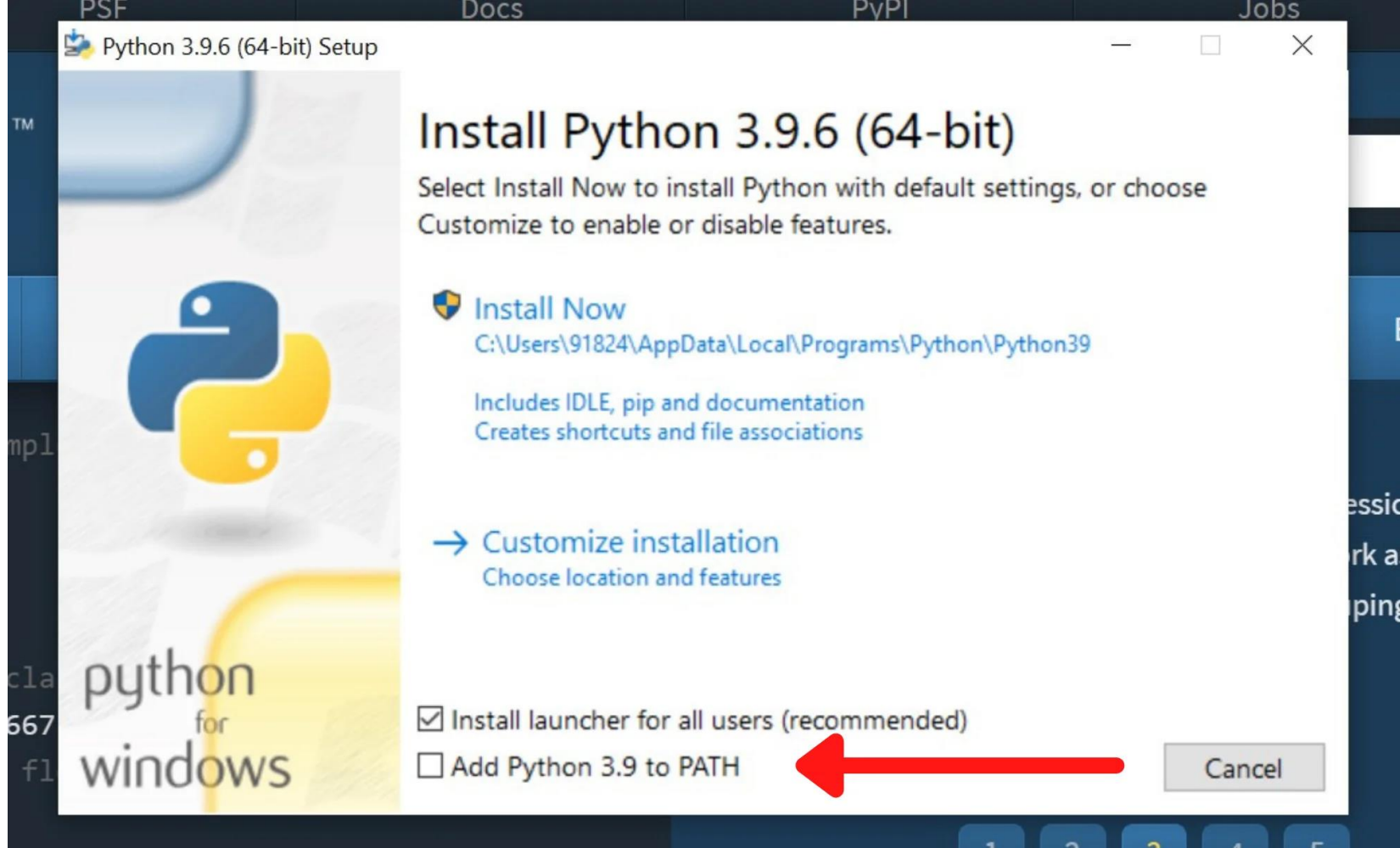
Documentation

Commu

Download the latest version for Windows

Download Python 3.13.1





حتما این گزینه را فعال کنید

cmd

```
Microsoft Windows [Version 10.0.19045.3448]
```

```
khodaverdian@KHODAVERDIAN2 C:\Users\khodaverdian
```

```
$ python
```

```
Python 3.12.5 (tags/v3.12.5:ff3bc82, Aug 6 2024, 20:45:27) [MSC v.1940 64 bit (AMD64)] on win32
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> print('Hi!')
```

```
Hi!
```

result

```
>>> 3 + 2
```

```
5
```

result

```
>>> |
```

command



main.py



```
1 print("I am an employee in rcii company")
```

```
2
```



Run



Share



Command Line Argument



I am an employee in rcii company



** Process exited - Return Code: 0 **

```
>>> print("Hello World!")  
Hello World!
```

Double quotation

“ ” „ ” „ ” „ ”
” ” ” ” ” ” // ”
” ” ” ” ” ” ” ”
” // ” ” ” ” ” ”
” ” ” ” ” ” ” ”
” ” ” ” ” ” ” ”
” ” ” ” ” ” ” ”


```
>>> print('Hello World!')  
Hello World!
```

```
>>> print("Hello World!");  
Hello World!
```

```
x = 10  
y = 20  
print(x + y)
```



```
x = 10; y = 20; print(x + y)
```

استفاده از ; در پایتون معمول نیست

```
print "Hello, World!"
```

```
>>> print "Hello, World!"
```

```
File "<stdin>", line 1
```

```
    print "Hello, World!"
```

```
      ^
```

```
SyntaxError: Missing parentheses in call to 'print'. Did you mean print("Hello, World!")?
```

```
print(Hello, World!)
```

```
>>> print(Hello, World!)
```

```
File "<stdin>", line 1
```

```
    print(Hello, World!)
```

```
        ^
```

```
SyntaxError: invalid syntax
```

```
print("Hello, World!")
```

```
>>> print("Hello, World!')
```

```
File "<stdin>", line 1
```

```
    print("Hello, World!')
```

```
        ^
```

```
SyntaxError: EOL while scanning string literal
```



```
print("He said \"Hello\" to everyone")
```

```
>>> print("He said "Hello" to everyone")
File "<stdin>", line 1
    print("He said "Hello" to everyone")
                        ^
SyntaxError: invalid syntax
```

```
print("My name is Ali and I'm learning Python.", "Nice to meet you!")
```

```
>>> print("My name is Ali and I'm learning Python.", "Nice to meet you!")
```

```
My name is Ali and I'm learning Python. Nice to meet you!
```

```
print(3+4)
```

```
>>> print(3+4)  
7
```

```
>>> print("3+4")  
3+4
```