Python - install - uti wiki - dibuix gràfic

Autor:
Data de publicació: 12-04-2026

Instal·lació i passos (molt fàcils)

01

Comprovar si ja tens Python instal·lat
Pas inicial

Molts ordinadors moderns ja el porten, així que val la pena mirar-ho primer.

Prem Tecla Windows

Escriu cmd i obre la finestra negra (Símbol del sistema)

Escriu: python --version

Si et surt un número (per exemple Python 3.10), ja el tens instal·lat

02

Instal·lar Python (si no hi és)
Important

Python és gratuït i segur, i la instal·lació és molt senzilla.

https://www.python.org/downloads/

Entra a la pàgina i prem Download Python 3.x

Quan s'obri l'instal·lador, marca la casella Add Python to PATH (molt important)

Prem Install Now i espera que acabi

03

Crear un fitxer amb el codi

El codi Python es guarda en un fitxer de text amb extensió .py.

Obre el Bloc de notes

Copia-hi el codi Python que et vaig donar

Desa el fitxer com: comptar_categories.py (o el nom que vulguis)

Assegura’t que el tipus de fitxer és Tots els fitxers i no .txt

04

Executar el codi
Ja gairebé ho tens

Ara ja pots fer-lo funcionar i veure el resultat.

Torna a obrir cmd (Símbol del sistema)

Ves a la carpeta on tens el fitxer, per exemple: cd C:UsersEL_TEU_NOMDesktop

Escriu: python comptar_categories.py

Prem Enter i veuràs el resultat a la pantalla

05

Interpretar el resultat

El programa et mostrarà quants articles hi ha a cada categoria.

Veureu línies com: Articles a 'Cavalls': 41 Articles a 'Hipologia': 96

I el total sumat

Si vols afegir més categories, t'ho puc adaptar

_____

programa Python complet que:

Llegeixi una categoria de la Viquipèdia (en català)

Recorri totes les subcategories recursivament

Compti tots els articles que pengen de cadascuna

I sumi el total de les dues categories: Cavalls i Hipologia

La manera correcta i robusta és usar l'API de MediaWiki, no pas fer scraping.

Programa Python per comptar articles d'una categoria (amb subcategories)
Funciona per a qualsevol categoria de la Viquipèdia en català.

python

import requests

API = "https://ca.wikipedia.org/w/api.php"

```
def get_category_members(category, cmtype="page|subcat", continue_token=None):
    params = {
        "action": "query",
        "list": "categorymembers",
        "cmtitle": f"Categoria:{category}",
        "cmlimit": "500",
        "cmtype": cmtype,
        "format": "json"
    }
    if continue_token:
        params["cmcontinue"] = continue_token

    r = requests.get(API, params=params).json()
    members = r["query"]["categorymembers"]
    cont = r.get("continue", {}).get("cmcontinue")
    return members, cont

def count_articles_recursive(category, visited=None):
    if visited is None:
        visited = set()

    if category in visited:
        return 0
    visited.add(category)

    total = 0
    continue_token = None

    while True:
        members, continue_token = get_category_members(category, continue_token=continue_token)

        for m in members:
            if m["title"].startswith("Categoria:"):
                subcat = m["title"].replace("Categoria:", "")
                total += count_articles_recursive(subcat, visited)
            else:
                total += 1

        if not continue_token:
            break

    return total

if __name__ == "__main__":
    categories = ["Cavalls", "Hipologia"]
    grand_total = 0

    for cat in categories:
        print(f"Comptant articles de la categoria {cat}...")
        count = count_articles_recursive(cat)
        print(f" ? {count} articles")
        grand_total += count

    print("nTOTAL SUMAT:", grand_total)
```

Què fa exactament aquest script

Usa action=query&list=categorymembers

Recorre totes les subcategories amb una crida recursiva

Evita bucles infinits amb un conjunt visited

Suma tots els articles (tipus page)

Funciona amb categories de qualsevol profunditat

_____

versió ampliada i professional del programa, amb totes les funcionalitats que has demanat:

Entrada interactiva de categories

Recorregut recursiu complet

Filtre per comptar només articles principals (namespace 0)

Generació d'un llistat complet d'articles

Exportació a CSV (tant del recompte com del llistat)

Tot plegat mantenint l'ús correcte de l'API de MediaWiki.

```
import requests
import csv

API = "https://ca.wikipedia.org/w/api.php"
```

```
def get_category_members(category, cmtype="page|subcat", continue_token=None):
    params = {
        "action": "query",
        "list": "categorymembers",
        "cmtitle": f"Categoria:{category}",
        "cmlimit": "500",
        "cmtype": cmtype,
        "format": "json"
    }
    if continue_token:
        params["cmcontinue"] = continue_token

    r = requests.get(API, params=params).json()
    members = r["query"]["categorymembers"]
    cont = r.get("continue", {}).get("cmcontinue")
    return members, cont


def count_and_list_articles(category, visited=None, articles=None):
    if visited is None:
        visited = set()
    if articles is None:
        articles = []

    if category in visited:
        return articles
    visited.add(category)

    continue_token = None

    while True:
        members, continue_token = get_category_members(category, continue_token=continue_token)

        for m in members:
            title = m["title"]

            if title.startswith("Categoria:"):
                subcat = title.replace("Categoria:", "")
                count_and_list_articles(subcat, visited, articles)

            else:
                # Només namespace 0 (articles principals)
                if m["ns"] == 0:
                    articles.append(title)

        if not continue_token:
            break

    return articles


def save_csv(filename, articles):
    with open(filename, "w", newline="", encoding="utf-8") as f:
        writer = csv.writer(f)
        writer.writerow(["Títol"])
        for a in articles:
            writer.writerow([a])


if __name__ == "__main__":
```

```
    print("Introdueix categories separades per comes (ex: Cavalls,Hipologia):")
    user_input = input("Categories: ").strip()

    categories = [c.strip() for c in user_input.split(",")]

    all_articles = []

    for cat in categories:
        print(f"nProcessant categoria: {cat}")
        articles = count_and_list_articles(cat)
        print(f" ? {len(articles)} articles principals trobats")
        all_articles.extend(articles)

        save_csv(f"{cat}_articles.csv", articles)
        print(f"   CSV generat: {cat}_articles.csv")

    # Eliminar duplicats si dues categories comparteixen articles
    all_articles = sorted(set(all_articles))

    print("n=== RESULTATS FINALS ===")
    print(f"Total d'articles únics trobats: {len(all_articles)}")

    save_csv("totes_les_categories_articles.csv", all_articles)
    print("CSV global generat: totes_les_categories_articles.csv")
```

DeepSeek trampa

programa en Python que, donada una URL d'una categoria de la Viquipèdia en català, recorri totes les subcategories de manera recursiva i compti tots els articles (pàgines que no són categories) que pengen d'ella.

A continuació et deixo el codi complet. Hauràs d'instal·lar la llibreria requests i beautifulsoup4 si no les tens:

Codi
Bash

```
pip install requests beautifulsoup4
```

Codi

```
import requests
from bs4 import BeautifulSoup
from urllib.parse import urljoin, urlparse
import time
import sys

def get_soup(url):
 """Obté el contingut d'una pàgina i el parseja amb BeautifulSoup"""
 headers = {
 'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36'
 }
 try:
 response = requests.get(url, headers=headers)
 response.raise_for_status()
 return BeautifulSoup(response.text, 'html.parser')
 except requests.RequestException as e:
 print(f"Error carregant {url}: {e}")
 return None

def extract_category_articles_and_subcats(soup, base_url):
```

```
    """Extreu els articles i subcategories d'una pàgina de categoria"""
    articles = []
    subcategories = []

    # Trobar la secció de pàgines (articles) - div amb id 'mw-pages'
    pages_div = soup.find('div', id='mw-pages')
    if pages_div:
    # Dins d'aquest div, els articles estan en llistes ()
    for li in pages_div.find_all('li'):
    link = li.find('a')
    if link and not 'Categoria:' in link.get('href', ''):
    articles.append(link.text)

    # Trobar la secció de subcategories - div amb id 'mw-subcategories'
    subcats_div = soup.find('div', id='mw-subcategories')
    if subcats_div:
    for li in subcats_div.find_all('li'):
    link = li.find('a')
    if link and 'Categoria:' in link.get('href', ''):
    subcat_url = urljoin(base_url, link.get('href'))
    subcategories.append((link.text, subcat_url))

    return articles, subcategories

def count_articles_in_category(category_url, visited=None, depth=0, verbose=True):
    """
    Compta recursivament tots els articles d'una categoria i les seves subcategories.
    Retorna (total_articles, llista_detall_articles)
    """
    if visited is None:
    visited = set()

    # Evitar bucles infinits
    if category_url in visited:
    if verbose:
    print("  " * depth + f"??  Ja visitat: {category_url}")
    return 0, []

    visited.add(category_url)

    if verbose:
    print("  " * depth + f"
```