## Visualizar repositorios con Plotly

Vamos a hacer una visualización utilizando los datos que hemos recopilado para mostrar la popularidad relativa de los proyectos Python en GitHub. Haremos un gráfico de barras interactivo: la altura de cada barra representará el número de estrellas que ha adquirido el proyecto, y podrás hacer clic en la etiqueta de la barra para ir a la página principal de ese proyecto en GitHub.

Guarda una copia del programa en el que hemos estado trabajando como *python\_repos\_visual.py*, y luego modifícalo para que quede como sigue:

**python\_repos\_visual.py**

import requests  
import plotly.express as px  
  
# Make an API call and check the response.  
url = "https://api.github.com/search/repositories"  
url += "?q=language:python+sort:stars+stars:>10000"  
  
headers = {"Accept": "application/vnd.github.v3+json"}  
r = requests.get(url, headers=headers)  
❶ print(f"Status code: {r.status\_code}")  
  
# Process overall results.  
response\_dict = r.json()  
❷ print(f"Complete results: {not response\_dict['incomplete\_results']}")  
  
# Process repository information.  
repo\_dicts = response\_dict['items']  
❸ repo\_names, stars = [], []  
for repo\_dict in repo\_dicts:  
 repo\_names.append(repo\_dict['name'])  
 stars.append(repo\_dict['stargazers\_count'])  
  
# Make visualization.  
❹ fig = px.bar(x=repo\_names, y=stars)  
fig.show()

Importamos Plotly Express y realizamos la llamada a la API como hemos venido haciendo. Seguimos imprimiendo el estado de la respuesta de la llamada a la API para saber si hay algún problema ❶. Cuando procesemos los resultados globales, seguimos imprimiendo el mensaje que confirma que hemos obtenido un conjunto completo de resultados ❷. Eliminamos el resto de las llamadas a print() porque ya no estamos en la fase exploratoria; sabemos que tenemos los datos que queremos.

A continuación, creamos dos listas vacías ❸ para almacenar los datos que incluiremos en el gráfico inicial. Necesitaremos el nombre de cada proyecto para etiquetar las barras (repo\_names) y el número de estrellas para determinar la altura de las barras (stars). En el bucle, añadimos a estas listas el nombre de cada proyecto y el número de estrellas que tiene.

Hacemos la visualización inicial con sólo dos líneas de código ❹. Esto es coherente con la filosofía de Plotly Express de que debes poder ver tu visualización lo más rápidamente posible antes de refinar su aspecto. Aquí utilizamos la función px.bar() para crear un gráfico de barras. Pasamos la lista repo\_names como argumento x y stars como argumento y.

[La Figura 17-1](#figure17-1) muestra el gráfico resultante. Podemos ver que los primeros proyectos son significativamente más populares que el resto, pero todos ellos son proyectos importantes en el ecosistema Python.

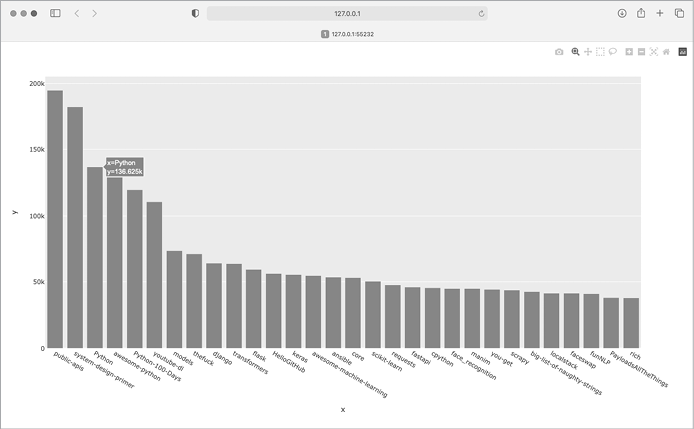


Figura 17-1: Los proyectos Python con más estrellas en GitHub

[anterior](c17_10.html)[Subtema 11 de 20: (Ver todo)](c17.html)[siguiente](c17_12.html)