



# Solución Desafío - Integración con respuesta

## Solución 1

Primero dividiremos el problema en 2 grandes partes.

- Hacer el request a la API de CoinDesk y obtener los resultados
- Procesar los resultados para obtener una lista con las fechas

```
import json
import requests

def request(requested_url):
    headers = {
        "cache-control": "no-cache",
        "Postman-Token": "2defb9f3-9b11-499b-be4f-82505a2f8e1a",
    }
    response = requests.request("GET", requested_url, headers=headers)
    return json.loads(response.text)

prices = request("https://api.coindesk.com/v1/bpi/historical/close.json")["bpi"]
```

El segundo paso consiste en obtener la lista con las fechas, para lo que hay distintas posibilidades. La primera pregunta que deberíamos hacernos es si podemos utilizar una sola línea para seleccionar elementos en base a un valor.

```
selected_data = [k for k, v in prices.items() if v < 5000]
selected_data
```

```
['2018-11-19',
 '2018-11-20',
 '2018-11-21',
 '2018-11-22',
```

```
'2018-11-23',  
'2018-11-24',  
'2018-11-25',  
'2018-11-26',  
'2018-11-27',  
'2018-11-28',  
'2018-11-29',  
'2018-11-30',  
'2018-12-01',  
'2018-12-02',  
'2018-12-03',  
'2018-12-04',  
'2018-12-05',  
'2018-12-06',  
'2018-12-07',  
'2018-12-08',  
'2018-12-09',  
'2018-12-10',  
'2018-12-11']
```

## Solución 2

Sin embargo hay muchas otras formas de resolver el problema, por ejemplo pudimos haberlo hecho utilizando un `for` y guardando en un nuevo diccionario todos los elementos que cumplan el criterio.

```
selected_data = []
for date, value in prices.items():
    if value < 5000:
        selected_data.append(date)
selected_data
```

```
['2018-11-19',
 '2018-11-20',
 '2018-11-21',
 '2018-11-22',
 '2018-11-23',
 '2018-11-24',
 '2018-11-25',
 '2018-11-26',
 '2018-11-27',
 '2018-11-28',
 '2018-11-29',
 '2018-11-30',
 '2018-12-01',
 '2018-12-02',
 '2018-12-03',
 '2018-12-04',
 '2018-12-05',
 '2018-12-06',
 '2018-12-07',
 '2018-12-08',
 '2018-12-09',
 '2018-12-10',
 '2018-12-11']
```

## Solución 3

Otra opción, bastante distinta, sería obtener una lista con únicamente los precios, filtrarla por los valores bajo 5000 y luego utilizar un diccionario invertido para buscarlos.

```
under_5000 = [v for v in prices.values() if v < 5000]
prices_inv = {v: k for k, v in prices.items()}
selected_data = [prices_inv[k] for k in under_5000]
selected_data
```

```
['2018-11-19',
 '2018-11-20',
 '2018-11-21',
 '2018-11-22',
 '2018-11-23',
 '2018-11-24',
 '2018-11-25',
 '2018-11-26',
 '2018-11-27',
 '2018-11-28',
 '2018-11-29',
 '2018-11-30',
 '2018-12-01',
 '2018-12-02',
 '2018-12-03',
 '2018-12-04',
 '2018-12-05',
 '2018-12-06',
 '2018-12-07',
 '2018-12-08',
 '2018-12-09',
 '2018-12-10',
 '2018-12-11']
```

Existen muchas soluciones a un problema siempre deberíamos preferir las mas sencillas.