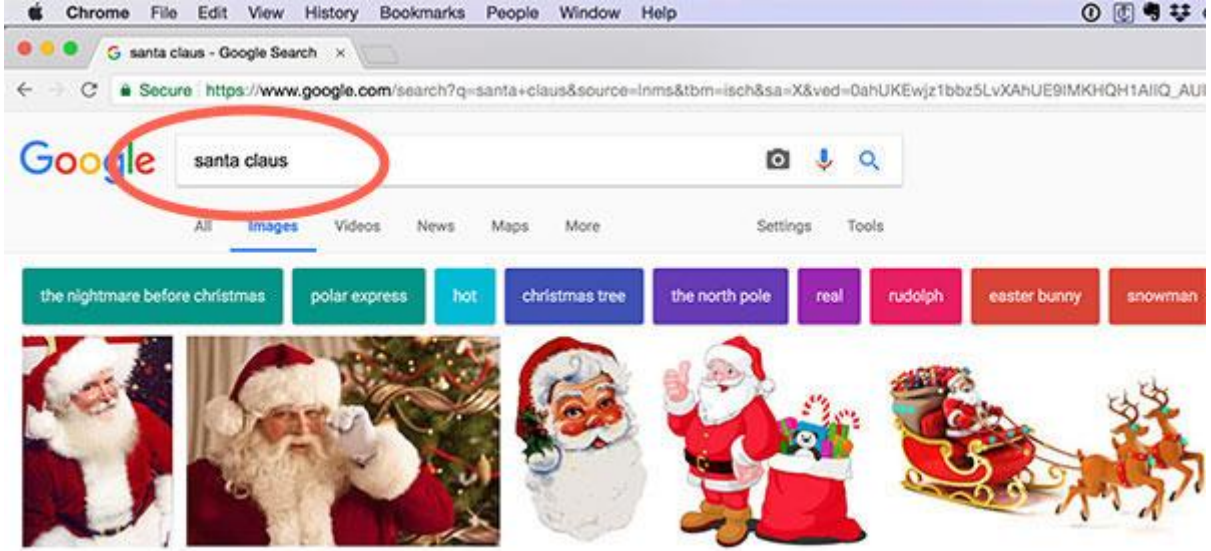
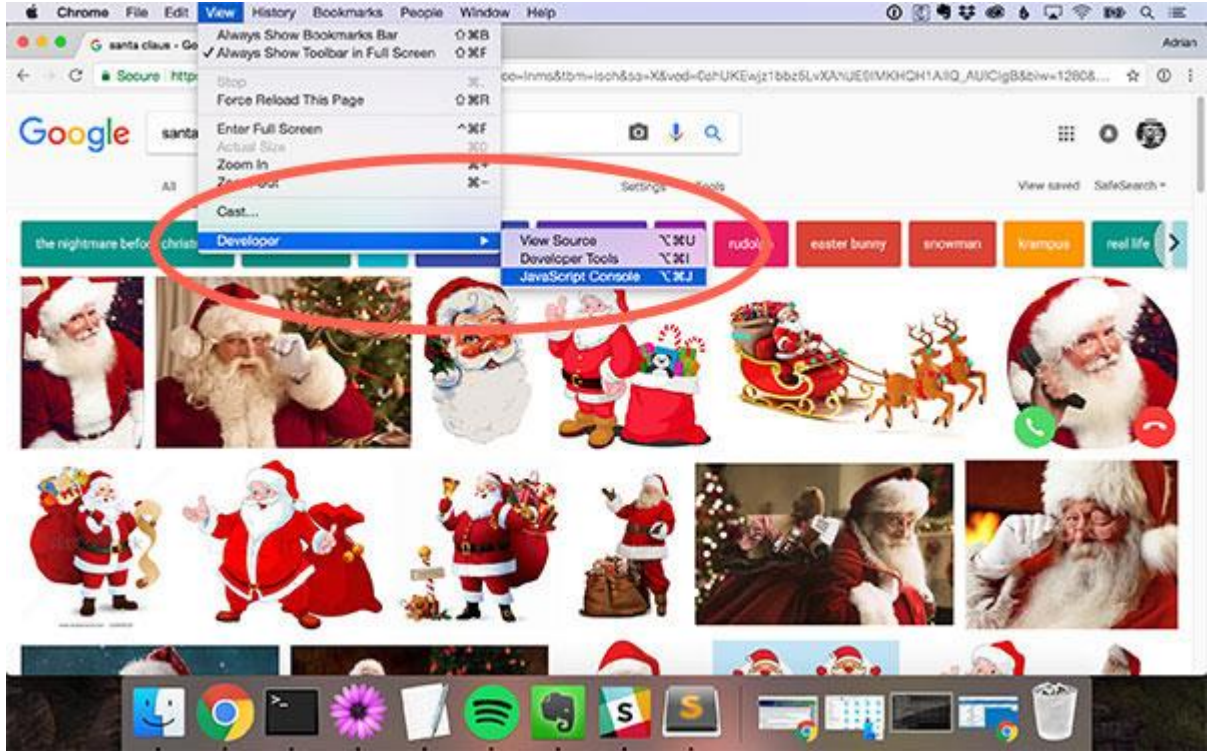


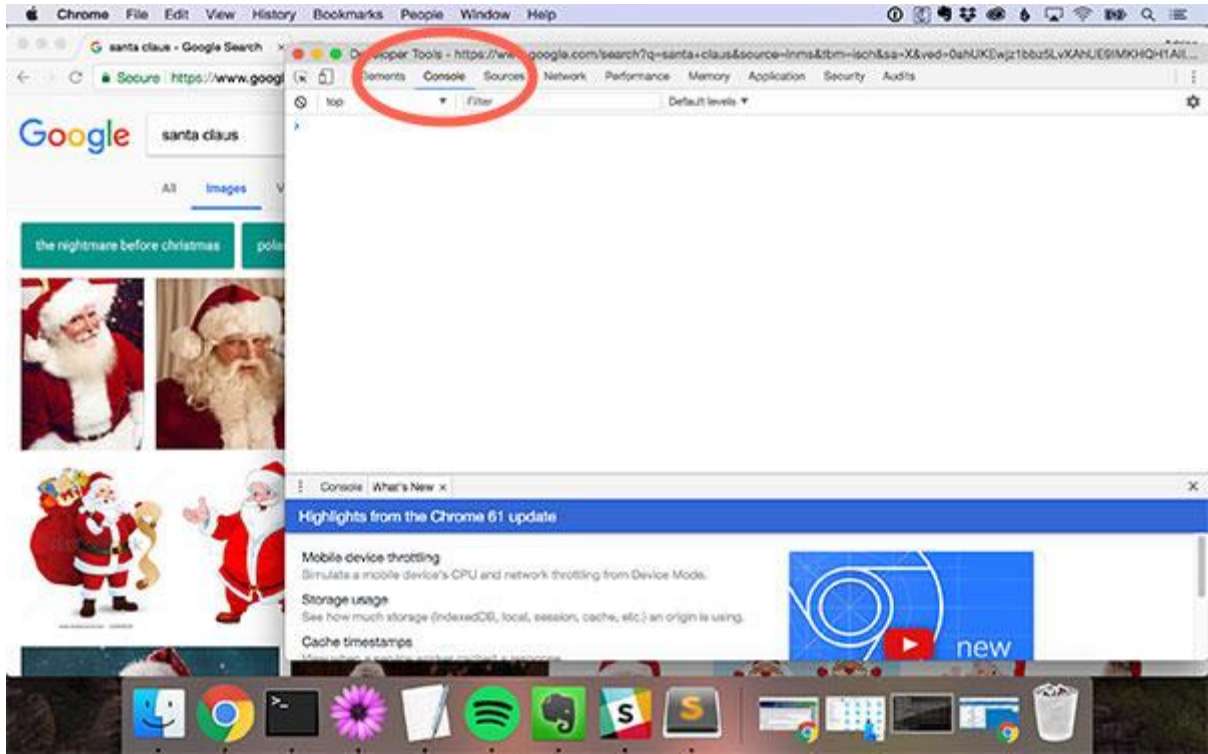
[Michael Sallomi](#) tarafından oluşturulmuş bu metotta makine öğrenmesi, derin öğrenme vb. için Google görsellerden nasıl resim toplanacağı anlatılacaktır.

1. İlk olarak örnek olarak toplayacağımız veri setinin adını yani "santa clause"ı tarayıcıdan girdiğimiz Google görseller kısmına yazıyoruz.

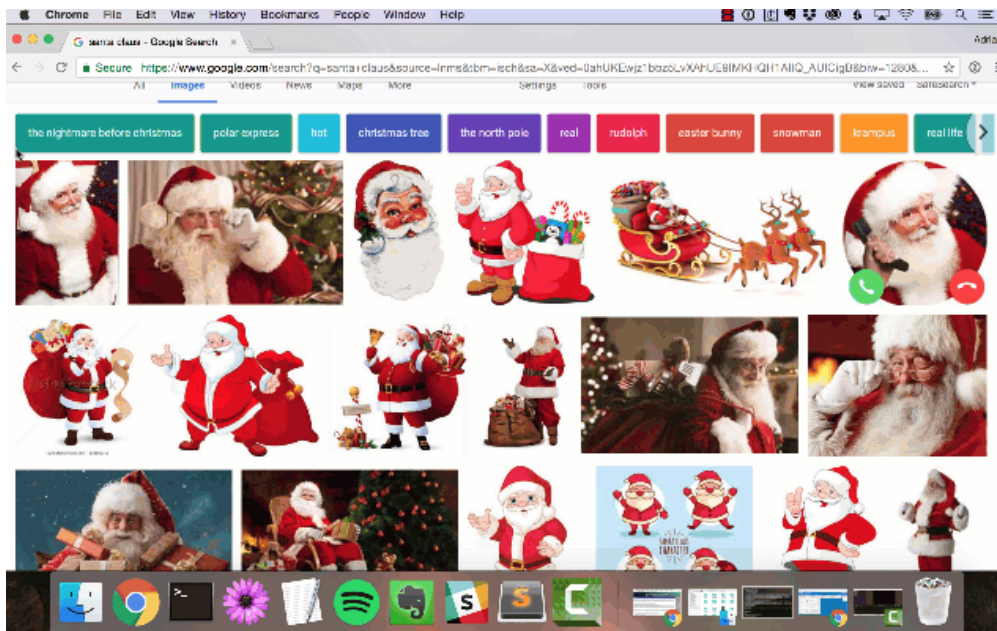


2. Devamında ise bulunduğumuz sekmede mouse sağ tuşu -> öğeyi incele-denetle -> konsol diyerek konsolu açıyoruz.





3. Ve mouse tekerleği ile sayfayı aşağı doğru kaydırıyoruz.



4. Resime sağ tuş ile tıklamayı simüle eden başka bir sayfaya gitme ihtiyacını azaltıp sağ tık menüsünden resmi otomatik alan scripti konsola kopyala-yapıştır yapıyoruz.

```
function simulateRightClick( element ) {
  var event1 = new MouseEvent( 'mousedown', {
    bubbles: true,
    cancelable: false,
    view: window,
    button: 2,
    buttons: 2,
    clientX: element.getBoundingClientRect().x,
    clientY: element.getBoundingClientRect().y
  } );
  element.dispatchEvent( event1 );
  var event2 = new MouseEvent( 'mouseup', {
    bubbles: true,
    cancelable: false,
    view: window,
    button: 2,
    buttons: 0,
    clientX: element.getBoundingClientRect().x,
    clientY: element.getBoundingClientRect().y
  } );
  element.dispatchEvent( event2 );
  var event3 = new MouseEvent( 'contextmenu', {
    bubbles: true,
    cancelable: false,
    view: window,
    button: 2,
    buttons: 0,
    clientX: element.getBoundingClientRect().x,
    clientY: element.getBoundingClientRect().y
  } );
  element.dispatchEvent( event3 );
}
```

5. Google görseller tam resmin URL sini sorgu parametresinde sakladığından saklanan URL parametresinin sorgudan çekmek gerekir. Onun için de aşağıdaki fonksiyonu kopyala-yapıştır yapmak yeterlidir.

```
function getURLParam( queryString, key ) {
  var vars = queryString.replace( /\^?/, '' ).split( '&' );
  for ( let i = 0; i < vars.length; i++ ) {
    let pair = vars[ i ].split( '=' );
    if ( pair[0] == key ) {
      return pair[1];
    }
  }
  return false;
}
```

6. Her URL contents parametresiyle createDownload fonksiyonundan geçer. Ve ilk olarak hiddenElement oluşturuyoruz. Devamında urls.txt oluşturarak içeriğini contents ile dolduruyoruz ve tıklamayı simüle ediyoruz.

```
function createDownload( contents ) {
    var hiddenElement = document.createElement( 'a' );
    hiddenElement.href = 'data:attachment/text,' + encodeURIComponent( contents );
    hiddenElement.target = '_blank';
    hiddenElement.download = 'urls.txt';
    hiddenElement.click();
}
```

7. Bu adımdaki grabUrls() fonksiyonu ise tüm resim URL'lerinin sağ tıklama içerik menüsünün simülasyonu ile elde edileceğine dair bir fonksiyondur. İçerik menüsündeki tam boyutlu resmi görüntüle seçeneğini otomatik olarak kullanır. Not: Internet Explorer'da çalışmaz

```
function grabUrls() {
    var urls = [];
    return new Promise( function( resolve, reject ) {
        var count = document.querySelectorAll(
            '.isv-r a:first-of-type' ).length,
            index = 0;
        Array.prototype.forEach.call( document.querySelectorAll(
            '.isv-r a:first-of-type' ), function( element ) {
            simulateRightClick( element.querySelector( ':scope img' ) );
            var interval = setInterval( function() {
                if ( element.href.trim() !== '' ) {
                    clearInterval( interval );
                    //resimlerin tam boyutlu hallerini çıkarma
                    let googleUrl = element.href.replace( /\.*(\?)/, '$1' ),
                        fullImageUrl = decodeURIComponent(
                            getURLParam( googleUrl, 'imgurl' ) );
                    if ( fullImageUrl !== 'false' ) {
                        urls.push( fullImageUrl );
                    }
                    // bazen URL "false" string olarak dönebiliyor ve
                    // biz yine de onları saymasını istiyoruz
                    index++;
                    if ( index == ( count - 1 ) ) {
                        resolve( urls );
                    }
                }
            }, 10 );
        } );
    } );
}
```

8. Uygulamayı başlatmak için temel noktamız grabUrls() fonksiyonudur. Dikkat etmemiz gereken de her URL'nin metin dosyasında kendi satırında olması için her URL'nin yeni satır karakteriyle nasıl birleştiğidir. Görebileceğimiz gibi createDownload fonksiyonu da son adımda çağırıldı.

```
grabUrls().then( function( urls ) {
    urls = urls.join( '\n' );
    createDownload( urls );
} );
```

9. Şimdi ise python kodlarına geçiyoruz. Bunun için resim_indirme.py adında bir dosya oluşturup adım adım kodları kopyala-yapıştır yapalım. Aşağıdaki kodda gerekli kütüphaneleri ekliyoruz. Not: Gerekli kütüphaneler kurulu değil ise sırasıyla cmd ekranına pip install imutils, pip install argparse, pip install requests, pip install os-win yazarak enter'layın.

```
from imutils import paths
import argparse
import requests
import os
```

10. Sonra urls.txt dosyasındaki verileri bölerek belleğe ekliyoruz. Komut satırında da iki argümanı elle ayırmamız gerekiyor.

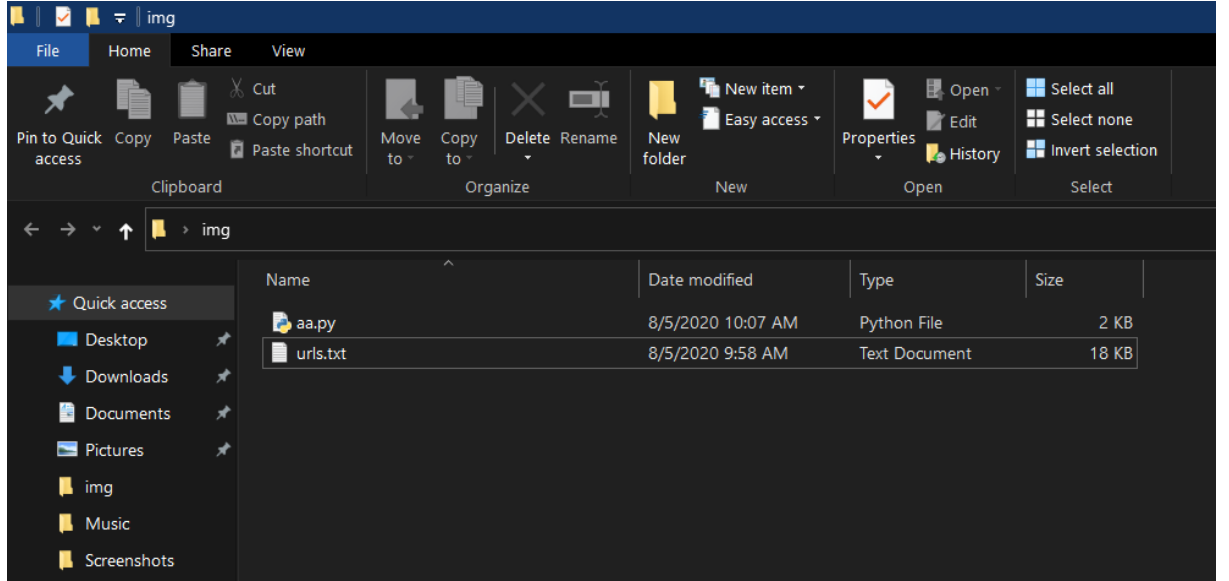
- a. -urls: JavaScript koduyla ürettiğimiz URL'leri içeren txt dosyasının yolu.
- b. -output: Google görsellerden indirilecek olan resimlerin indirileceği klasörün yolu.

```
# argümanları yani url'leri bölecek yapıyı oluşturur ve argümanları
# ayrıştırır
ap = argparse.ArgumentParser()
ap.add_argument("-u", "--urls", required=True,
                help="JavaScript koduyla ürettiğimiz URL'leri içeren txt dosyasının
yolu.")
ap.add_argument("-o", "--output", required=True,
                help="Google görsellerden indirilecek olan resimlerin indirileceği
klasörün yolu.")
args = vars(ap.parse_args())
# girdi dosyasındaki URL'leri alır ve toplam indirilen resim sayısını
# sıfıra eşitleyerek çalışmaya başlar
rows = open(args["urls"]).read().strip().split("\n")
total = 0
```

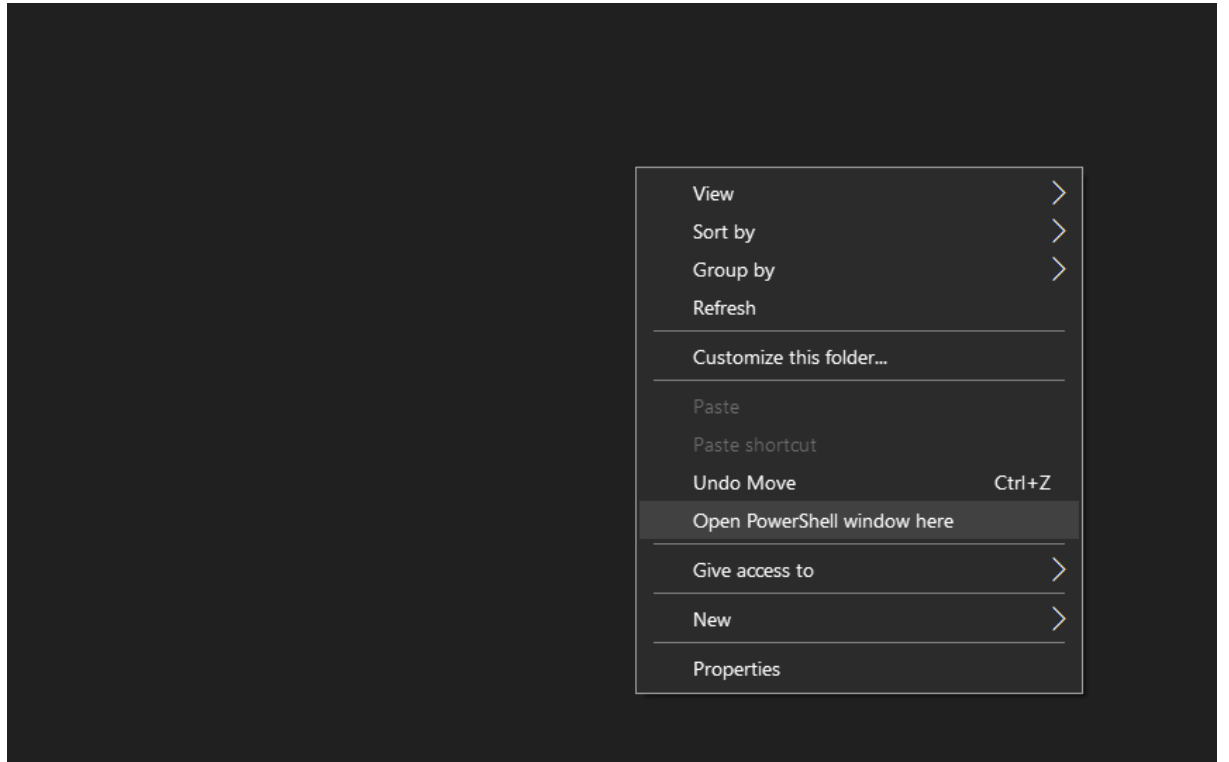
11. URL'lerin üzerinden döngü kurup bütün resimleri indiren kod.

```
# URL'lerin döngüsü
for url in rows:
    try:
        # resim indirmeyi dener
        r = requests.get(url, timeout=60)
        # resmi 00000000.jpg olarak adlandırıp diske kaydeder
        p = os.path.sep.join([args["output"], "{}.jpg".format(
            str(total).zfill(8))])
        f = open(p, "wb")
        f.write(r.content)
        f.close()
        # sayacı günceller
        print("[BİLGİ] indirildi: {}".format(p))
        total += 1
    # herhangi bir hatayla karşılaşıldığında alttaki hatayı basar
    except:
        print("[BİLGİ] indirme hatası {}...es geçildi".format(p))
```


12. Son olarak resimleri indireceğimiz bir klasör oluşturuyoruz. Bu klasörün içine bir önceki adımlarda oluşturduğumuz python ve JavaScript koduyla oluşan urls.txt dosyasını atıyoruz.



13. Oluşturduğumuz klasör içerisinde shift+sağ tuşa tıklayarak gelen pencereden “Powershell penceresini burada aç”a tıklayarak bulunduğumuz klasör yolunda Powershell açıyoruz.



14. Açılan Powershell ekranına

python "9.adımda oluşturduğumuz python dosyasının adı" -urls urls.txt -output .

yazarak enterliyoruz ve resimler inmeye başlıyor.

