



# Basic Tools for Scraping

WORKSHOP 1 | CREATOR: CHARLOTTE LLOYD

# Outline

- I. Introduction
- II. Scraping Overview
- III. Workshop Example
- IV. Chrome Inspect & Console
- V. Terminal
- VI. Verify Data
- VII. Celebration, Back-slapping

# Introduction

PART I

# Programming Philosophy

- ▶ Concepts are key.
- ▶ Syntax is secondary.
- ▶ Stackoverflow is your friend.

# Scraping Overview

PART II

# What is scraping?



# Scraping Process // Battle Plan

- ▶ 1. Surveillance
  - ▶ Evaluate the page, learn the terrain.
- ▶ 2. Plan of Attack
  - ▶ Brainstorm ways to approach the enemy.
- ▶ 3. Write code
  - ▶ Be willing to change your strategy if you encounter obstacles or see another “weakness” to exploit.
- ▶ 4. Emerge bloodied, yet victorious.
  - ▶ Verify the data before all that syntax evaporates from your short term memory.

# Step 1: Evaluate

- ▶ Navigate to the web page.
  - ▶ What data would you like to scrape?
  - ▶ Where is that data located visually on the page?
- ▶ Take a deep breath and right click to open the page source.
  - ▶ Where is the data you need located?
  - ▶ How is the data formatted? (HTML table? external file?)
  - ▶ Can you identify any HTML patterns for that data?



# Step 2: Brainstorm

- ▶ What is the best (easiest) way to interact with the structure of the web page?
- ▶ What format would be best (easiest) for saving/exporting the data?
- ▶ Think at a conceptual level first:
  - ▶ “If I can find a way to grab all those links, I’ll have all the files I need.”
  - ▶ “Hm, it looks like I need data from two different HTML tables.”
  - ▶ “All the paragraphs I need have the same id!”
- ▶ Reach into your technical toolkit second:
  - ▶ jQuery, shell commands, basic Python, Python packages, LOTS of other stuff
  - ▶ Your toolkit is small now, but it will grow!
  - ▶ Don’t forget, it’s not illegal to do some things manually if it’s faster.

# Step 3: Code

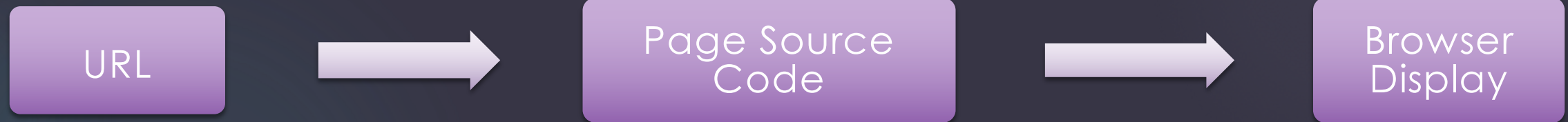
- ▶ Pursue a strategy. Wrestle with syntax. Change your strategy.
- ▶ Curse syntax. Curse the mother of all syntax.
- ▶ Copy examples from stackoverflow as much as you possibly can.

## 4. WIN

- ▶ Verify the data before you wrap-up.
- ▶ Make sure you take some notes and save things logically.

# What are web pages?

- ▶ Web pages are fundamentally files of code then displayed by browsers such as Chrome, Firefox, and Safari.



www.google.com

```
1 <!doctype html><html itemscope=""  
  itemtype="http://schema.org/WebPage" lang="en"><head><meta  
  content="Search the world's information, including  
  webpages, images, videos and more. Google has many special  
  features to help you find exactly what you're looking for."  
  name="description"><meta content="nooodp" name="robots">  
  <meta  
  content="/images/branding/google/ix/google_standard_color  
  _128dp.png" itemprop="image"><link  
  href="/images/branding/product/ico/google_ldp.ico"  
  rel="shortcut icon"><meta content="origin" id="mref"  
  name="referrer"><title>Google</title> <script>(function()  
{window.google=  
{kEEl: 'NuGUNWIG_NuKijwTm47iwCQ', kEKPI: '750721,1351903,1352240  
,1352382,3300103,3300131,3300164,3312811,3313275,3313321,37  
00303,3700347,4029815,4032678,4038012,4043492,4045841,40483  
47,4062664,4065786,4065918,4066195,4067859,4069839,4069841,  
4071842,4072774,4073405,4073726,4073959,4076096,4076931,407  
6999,4077777,4079105,4079894,4081038,4081485,4082194,408220  
1,4082446,4082618,4082700,4083476,4083863,4084343,4084858,4  
084976,4086011,4086498,4088154,4088429,4088525,4089003,4089  
183,4089538,4090352,4090414,4090442,4090547,4090549,4090657  
,4090730,4090806,4090851,4090894,4091068,4091966,4092218,40
```



# What are web pages?

- ▶ The main code on webpages is HTML.
  - ▶ Other code may be called on webpages to do fancy things. This commonly includes CSS, php, json, and javascript.
  - ▶ In addition to the “code” files, websites may also link to or display files, such as pdfs, images, and videos. These files *all* have their very own URLs (kind of like mini-webpages themselves)!

# What is HTML?

- ▶ For scraping purposes, you don't need to know much HTML.
- ▶ The key is to look for patterns in the HTML structure of the page source... and then exploit!
  - ▶ Fortunately HTML is full of great patterns.

# HTML Facts

- ▶ HTML features “tags”, usually in convenient little pairs.
  - ▶ `<html></html>` (web page)
  - ▶ `<head></head><body></body>` (parts of a webpage)
  - ▶ `<p></p>` (paragraph)
  - ▶ `<b></b>` (bold text)
  - ▶ `<i></i>` (italics)
  - ▶ `<table></table>` (table formatting)
  - ▶ `<tr></tr>` (table row)
  - ▶ `<td></td>` (table cell)
  - ▶ `<a></a>` (hyperlink)

# HTML Facts

- ▶ Text that's not in a tag is displayed (i.e. it's visible in the browser).

```
<html><body><p>Hello World.</p></body></html>
```

=

Hello World.

```
<html><body><p><b><i>Hello</i> World.</b></p></body></html>
```

=

***Hello World.***



# HTML Facts

- ▶ Sometimes tags have ids, classes, and styles. Don't worry about why.
  - ▶ `<h1 id="myHeader">Hello World!</h1>`
  - ▶ `<a id="photoalbum">cute puppy photo</a>`
  - ▶ `<a href="www.google.com" class="myClass">click me</a>`

<html>Let's look go at some page source!</html>

- ▶ If you think the title of this slide is mildly funny, you're grasping the concepts very well so far!
- ▶ [www.example.com](http://www.example.com)

# Workshop Example

PART III



<http://www.goes-r.gov/users/2016-OCONUS.html>

# Browser Display

The screenshot displays the GOES-R website interface. At the top, there is a navigation bar with icons for Home, Mission, User Info, Outreach, Multimedia, Resources, and Organization. Below the navigation bar is a search bar. The main content area features a header for a meeting: "2016 GOES-R JPSS OCONUS SATELLITE PROVING GROUND TECHNICAL INTERCHANGE MEETING" held in Honolulu, Hawaii, from June 27-30, 2016. A "USER INFO." link is visible in the top left of the content area. The meeting agenda is organized by date, with a section for "June 28, 2016" containing 13 items, each with a title, presenter name, and a link (pdf or pptx). A second section for "June 29, 2016" contains two items.

June 28, 2016		
GOES-R Welcome and Program Update	Greg Island	<a href="#">pptx</a>
GOES-R Proving Ground Update	Steve Goodman	<a href="#">pptx</a>
GOES-R Post Launch Activities	Wayne MacKenzie	<a href="#">pptx</a>
JPSS Program Welcome and Proving Ground Update	Mitch Goldberg	<a href="#">pptx</a>
JPSS Post Launch Activities	Lihang Zhou	<a href="#">pptx</a>
NOAT Priorities for OCONUS	Bill Ward	<a href="#">pptx</a>
SPoRT Support to OCONUS and Future Plans (include action item status)	Geoffrey Stano	<a href="#">pptx</a>
GMA Support to OCONUS and Future Plans	Tom Heinrichs and Eric Stevens	<a href="#">pptx</a>
CIRA/RAMMB Support to OCONUS and Future Plans	C. Seaman	<a href="#">pptx</a>
CMSS/ASPS Support to OCONUS and Future Plans	Wayne Feltz	<a href="#">pptx</a>
GOES-R Products from Himawari-8	Walter Wolf	<a href="#">pptx</a>
Himawari-8 satellite image utilization and user readiness for the new data - the Australian VLab Centre of Excellence experience	Bodo Zeschke	<a href="#">pdf</a>
June 29, 2016		
JPSS Training Plan	Joni Torres	<a href="#">pptx</a>
Himawari-8 Training Program	Jordan Gerth	<a href="#">pptx</a>

# Page Source

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5
6 <!-- PAGE EDITOR NOTE: search for "****" - alter only those parts and not the surrounding structural code.
7
8 <!-- STANDARD/SHARED <HEAD> TAGS FOR PROJECT: meta tags, std css links etc -->
9 <!-- htmlHeadSET.html: STANDARD/SHARED <HEAD> TAGS FOR PROJECT: meta tags, std css links etc -->
10
11 <!-- MOBILE SITE SCALING: for a non resp home design, set width to 1300-1400, and after testing found either no scale, orscale 1.0 behaves best
12 overall behavior on mobiles and ipad
13 <meta name="viewport" content="width=1300, initial-scale=1.0">
14
15 -->
16 <!-- for a RESP design, -->
17 <meta name="viewport" content="width=device-width">
18
19 <!-- SHARED META TAGS ETC -->
20 <meta http-equiv="Content-Type" content="text/html;">
21 <meta name="description" content="GOSS-K Program NOAA NADA">
22 <meta name="orgname" content="Code 400">
23 <meta name="rno" content="Lauren Gaches">
24 <meta name="content-owner" content="Michelle Smith">
25 <meta name="webmaster" content="Steve Sabia">
26 <meta name="keywords" content="">
27
28 <!-- SHARED CSS FILES -->
29
30 <!-- Bootstrap -->
31 <!-- SEM rev1.3 on 1.1.16: using bootstrap.css instead of bootstrap.min.css, was edited to use 99px break points -->
32 <link href="/toplevel/includes/bootstrap/css/bootstrap.css" rel="stylesheet">
33 <link href="/toplevel/includes/bootstrap/overrides.css" rel="stylesheet">
34
35 <!-- icons -->
36 <link href="/toplevel/includes/fonts/fontello&#215;dist/css/fontello-embedded.css" rel="stylesheet"> <!-- must use this version of fontello.css for
37 FFox to work, or alter server config, see readme.txt - was <link href="fontello.css" rel="stylesheet"> -->
38
39 <!-- fonts -->
40 <link rel="stylesheet" id="tp-oswald-css" href="http://fonts.googleapis.com/css?family=Oswald&#215;v=4.1" type="text/css" media="all" />
41
42
43 <!-- OLD WEBSITE CSS - legacy styles -INCLUDE THIS FIRST SO new css OVERRIDES legacy content uses styles in this to for content -->
44
45 <!-- LEGACY CSS - the OLD/legacy site css styles compiled into one file here in order to override them in later css definitions and includes.
46 Will eventually be eliminated. -->
47 <!-- commenting out all legacy styles for testing
48 <link href="/include/css/gossLegacyContent.css" rel="stylesheet" type="text/css" media="screen" />
49 -->
50
51 <!-- LEGACY CSS OVERRIDES - Override selectively styles from gossLegacyContent.css (doing this in separate file as it will collect all styles
```

# Surveillance: Table Data

```
<!-- constrain to 80% on big screens, scroll horo on overflow small screens -->
<div class="tedRow#0 table-responsive">
  <!-- width modified to % flex -->
  <table align="center" width="100%"

summary="table contains links to conference presentations, papers, pdfs" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td align="left" valign="top"><table width="100%" border="0" cellpadding="5" cellspacing="5">
      <tr>
        <td width="12"></td>
      </tr>
      <tr>
        <td align="center" valign="top" class="bodytext12table-LH-17"><table width="90%" border="0" align="center" cellpadding="0" cellspacing="1">
          <tr>
            <td bgcolor="#001959"><table width="100%" border="0" cellspacing="1" cellpadding="1">
              <tr>
                <td align="top" bgcolor="#FFFFFF"><table width="100%" border="0" cellspacing="4" cellpadding="4">
                  <tr bgcolor="#95B0C0">
                    <td colspan="4" align="left" valign="top" bgcolor="#95B0C0" class="bodytext14blue"><table width="100%" border="0" cellspacing="0" cellpadding="0">
                      <tr>
                        <td width="55%" align="left" valign="bottom" class="lrg-white">June 28, 2016 </td>
                        <td width="45%" align="left" valign="bottom" class="">&nbsp;&nbsp;&nbsp;</td>
                      </tr>
                    </table></td>
                  </tr>
                <tr>
                  <td width="61%" align="left" valign="top" class="bodytext12table-LH-17"><strong>OOES-R Welcome and Program Update</strong></td>
                  <td width="4%" align="left" valign="top" class="bodytext12">&nbsp;&nbsp;&</td>
                  <td width="26%" align="left" valign="top" class="bodytext12">Greg Mandt</td>
                  <td width="9%" align="center" valign="top" class="bodytext12table-LH-17"><a href="/users/docs/2016/OCONUS 2016/June 28/OCONUS_Greg_Mandt_GOES-R_StatusUpdate_FINAL2.pptx" title="Greg Mandt" target="_blank">pptx</a></td>
                </tr>
                <tr>
                  <td align="left" valign="top" class="bodytext12table-LH-17"><strong>OOES-R Proving Ground Update</strong></td>
                  <td align="left" valign="top" class="bodytext12">&nbsp;&nbsp;&</td>
                  <td align="left" valign="top" class="bodytext12">Steve Goodman</td>
                  <td width="9%" align="center" valign="top" class="bodytext12table-LH-17"><a href="/users/docs/2016/OCONUS 2016/June 28/OCONUS 6-28 Goodman_GOES-R Proving Ground Update.pptx" title="Goodman" target="_blank">pptx</a></td>
                </tr>
                <tr>
                  <td align="left" valign="top" class="bodytext12table-LH-17"><strong>OOES-R Post Launch Activities</strong></td>
                  <td align="left" valign="top" class="bodytext12">&nbsp;&nbsp;&</td>
                  <td align="left" valign="top" class="bodytext12">Wayne MacKenzie</td>
                  <td width="9%" align="center" valign="top" class="bodytext12table-LH-17"><a href="/users/docs/2016/OCONUS 2016/June 28/MacKenzie OCONUS 2016 Final.pptx" title="MacKenzie" target="_blank">pptx</a></td>
                </tr>
              </tr>
            </table>
          </tr>
        </table>
      </tr>
    </td>
  </tr>
</table>
```



# Surveillance: Table Data

- ▶ Let's find out how many of the following elements there are on the page (using a simple search):
  - ▶ tables
    - ▶ "`<table`"
    - ▶ "`</table>`"
  - ▶ rows
    - ▶ "`<tr`"
    - ▶ "`</tr>`"
  - ▶ cells
    - ▶ "`<tr>`"
    - ▶ "`</tr>`"
  - ▶ external links
    - ▶ "`<a href`"



# Plan of Attack

- ▶ Problem Statement: How can we save all the files on this page?
- ▶ Make a list of all links (`<a>`) found in cells (`<tr>`)
  - ▶ Download pptx and pdf files from this list of links

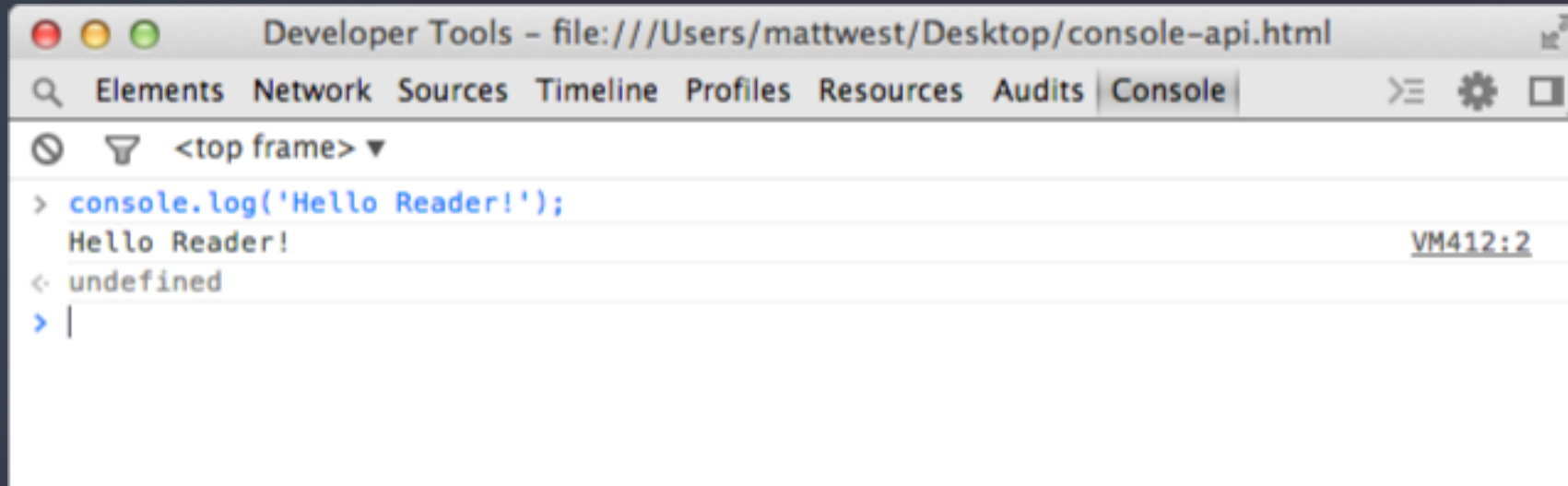


# Chrome Console: jQuery (JavaScript)

PART IV

# Introducing Chrome's console

- ▶ A JavaScript programming interface built into Chrome
  - ▶ Accessible through the “Inspect” option on right-click



The screenshot shows the Chrome Developer Tools interface with the Console tab selected. The title bar reads "Developer Tools - file:///Users/mattwest/Desktop/console-api.html". The console shows a log message: `> console.log('Hello Reader!');` followed by the output `Hello Reader!` and the source `VM412:2`. Below the log message, the prompt `< undefined` is visible, and a new line with a prompt `> |` is ready for input.

# JavaScript & jQuery

- ▶ JavaScript a coding language that's now an integral part of the web's linguistic ecosystem because it plays nice with HTML
  - ▶ JavaScript can often found within webpages (achieving fancy things HTML can't do) and taking long walks on the beach
  - ▶ JavaScript is also great for interacting with webpages (such as through Chrome's console)
- ▶ jQuery is a specialized part of the Javascript language
  - ▶ Involves the \$ sign, not entirely sure how to tell them apart
  - ▶ <https://blog.udemy.com/jquery-vs-javascript/>

# WARNING

- ▶ jQuery will not work on all webpages.
- ▶ If you're not sure, my brother says to run this command first
  - ▶ 

```
var script = document.createElement('script');script.src = "https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/jquery.min.js";document.getElementsByTagName('head')[0].appendChild(script);
```

# Simple jQuery commands

- ▶ Let's repeat our previous counting exercise with jQuery commands
  - ▶ tables
    - ▶ `$("table").length`
  - ▶ rows
    - ▶ `$("tr").length`
  - ▶ cells
    - ▶ `$("td").length`
  - ▶ links
    - ▶ `$("a").length`

# Find all links within cells

- ▶ `$("#tr a").length`
  - ▶ Tells us the number of links
- ▶ `$("#tr a")`
  - ▶ Creates a temporary object full of links
- ▶ `var a_list = $("#tr a");`
  - ▶ Saves an object called "a\_list" full of links that we can do stuff with later
- ▶ `a_list`
  - ▶ Displays the variable (check out the link properties!)

# Make a list of hrefs

- ▶ `a_list.each(function(){console.log(this.href);});`
  - ▶ Prints *console.log* all *.each(function)* of the links *this.href* in the variable *a\_list*
- ▶ `var href_list = [];`
  - ▶ Makes a new empty "array" (list)
- ▶ `a_list.each(function(){href_list.push(this.href);});`
  - ▶ Adds *push* all *.each(function)* of the links *this.href* in the variable *a\_list* to the array *href\_list*



# Copy list of URLs

- ▶ `copy(href_list);`
- ▶ open a TextEdit file
  - ▶ Paste the href\_list copied from the Chrome console
  - ▶ do a little bit of manual clean up so that each URL is on one line (no quotes!)
  - ▶ save as a plain text file (you may have to go to Format -> Make Plain Text) with a .txt ending

# Terminal

PART V

# Terminal / PowerShell

- ▶ Very powerful. Interacts with files, and the Internet. Uses Unix. I'm pretty sure...
  - ▶ might also be called "bash"? and shell?
- ▶ pwd
  - ▶ gets the "present working directory"
- ▶ cd /Users/your/path/here
  - ▶ changes working directory

# Save files

- ▶ `for i in `cat urls.txt`; do curl -O $i; done`
  - ▶ for each URL *i* in the *urls.txt* file, grab *curl* the file and save it `-O`

# Verify Data

PART VI

# Make notes, be organized

- ▶ This kind of piecemeal code is hard to come back to later, so if you'll need it again, organize it and write yourself notes.
  - ▶ SYNTAX WILL LEAK OUT OF YOUR BRAIN FASTER THAN ALL THE OTHER IMPORTANT THINGS THAT YOU HAVE ALREADY FORGOTTEN YOU WERE SUPPOSED TO REMEMBER. PLEASE BELIEVE ME THAT JUST A FEW SHORT WEEKS FROM NOW YOU WILL NOT KNOW WHY YOU DID THAT THING YOU DID OR WHAT THAT VARIABLE MEANS OR WHAT THAT FUNCTION DOES AND WHY YOU APPARENTLY PUT THAT THERE. AND WHY ISN'T THIS PACKAGE UPDATE WORKING WITH MY OLD CODE AND DO I EVEN HAVE THE CORRECT THINGS INSTALLED TO MAKE THIS ALL WORK AGAIN?



