Space X Falcon 9 First Stage Landing Prediction

Web scraping Falcon 9 and Falcon Heavy Launches Records from Wikipedia

Estimated time needed: **40** minutes

In this lab, you will be performing web scraping to collect Falcon 9 historical launch records from a Wikipedia page titled `List of Falcon 9 and Falcon Heavy launches`

https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches (https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches? utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id: SkillsNetwork-Channel-SkillsNetworkCoursesIBMDS0321ENSkillsNetwork26802033-2021-01-01)

Falcon 9 first stage will land successfully

Several examples of an unsuccessful landing are shown here:

More specifically, the launch records are stored in a HTML table shown below:

Objectives

Web scrap Falcon 9 launch records with `BeautifulSoup`:

- Extract a Falcon 9 launch records HTML table from Wikipedia
- Parse the table and convert it into a Pandas data frame

First let's import required packages for this lab

In [1]:

```
!pip3 install beautifulsoup4
!pip3 install requests
```

'pip3' 不是內部或外部命令、可執行的程式或批次檔。
'pip3' 不是內部或外部命令、可執行的程式或批次檔。

In [2]:

```
import sys

import requests
from bs4 import BeautifulSoup
import re
import unicodedata
import pandas as pd
```

and we will provide some helper functions for you to process web scraped HTML table

```python
def date_time(table_cells):
    """
    This function returns the data and time from the HTML  table cell
    Input: the  element of a table data cell extracts extra row
    """
    return [data_time.strip() for data_time in list(table_cells.strings)][0:2]

def booster_version(table_cells):
    """
    This function returns the booster version from the HTML  table cell
    Input: the  element of a table data cell extracts extra row
    """
    out=''.join([booster_version for i,booster_version in enumerate( table_cells.string
s) if i%2==0][0:-1])
    return out

def landing_status(table_cells):
    """
    This function returns the landing status from the HTML table cell
    Input: the  element of a table data cell extracts extra row
    """
    out=[i for i in table_cells.strings][0]
    return out


def get_mass(table_cells):
    mass=unicodedata.normalize("NFKD", table_cells.text).strip()
    if mass:
        mass.find("kg")
        new_mass=mass[0:mass.find("kg")+2]
    else:
        new_mass=0
    return new_mass


def extract_column_from_header(row):
    """
    This function returns the landing status from the HTML table cell
    Input: the  element of a table data cell extracts extra row
    """
    if (row.br):
        row.br.extract()
    if row.a:
        row.a.extract()
    if row.sup:
        row.sup.extract()

    colunm_name = ' '.join(row.contents)

    # Filter the digit and empty names
    if not(colunm_name.strip().isdigit()):
        colunm_name = colunm_name.strip()
        return colunm_name
```

To keep the lab tasks consistent, you will be asked to scrape the data from a snapshot of the `List of Falcon 9 and Falcon Heavy launches` Wikipage updated on `9th June 2021`

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_He
avy_launches&oldid=1027686922"
```

Next, request the HTML page from the above URL and get a `response` object

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

In [5]:

```
# use requests.get() method with the provided static_url

# assign the response to a object
response = requests.get(static_url)
```

Create a `BeautifulSoup` object from the HTML `response`

In [6]:

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
response_BS = BeautifulSoup(response.text, 'html5lib')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

In [7]:

```
# Use soup.title attribute
print(response_BS.title)
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

TASK 2: Extract all column/variable names from the HTML table header

Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the external reference link towards the end of this lab

In [8]:

```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = response_BS.find_all('table')
```

Starting from the third table is our target table contains the actual launch records.

```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

Flight No.	Date and time (UTC)	Version, Booster [b]	Launch site	Payload [c]	Payload mass	Orbit	Customer	Launch outcome	Booster landing
1	4 June 2010, 18:45	F9 v1.0 [7] B0003.1 [8]	CCAFS, SLC-40	Dragon Spacecraft Qualification Unit		LEO	SpaceX	Success	Failure [9][10] (parachute)

```
<tr>
<td colspan="9">First flight of Falcon 9 v1.0.<sup class="reference" id="c
ite_ref-sfn20100604_17-0"><a href="#cite_note-sfn20100604-17">[11]</a></su
p> Used a boilerplate version of Dragon capsule which was not designed to
separate from the second stage.<small>(<a href="#First_flight_of_Falcon_
9">more details below</a>)</small> Attempted to recover the first stage by
parachuting it into the ocean, but it burned up on reentry, before the par
achutes even deployed.<sup class="reference" id="cite_ref-parachute_18-0">
<a href="#cite_note-parachute-18">[12]</a></sup>
</td></tr>
<tr>
<th rowspan="2" scope="row" style="text-align:center;">2
</th>
<td>8 December 2010,<br/>15:43<sup class="reference" id="cite_ref-spacefli
ghtnow_Clark_Launch_Report_19-0"><a href="#cite_note-spaceflightnow_Clark_
Launch_Report-19">[13]</a></sup>
</td>
<td><a href="/wiki/Falcon_9_v1.0" title="Falcon 9 v1.0">F9 v1.0</a><sup cl
ass="reference" id="cite_ref-MuskMay2012_13-1"><a href="#cite_note-MuskMay
2012-13">[7]</a></sup><br/>B0004.1<sup class="reference" id="cite_ref-bloc
k_numbers_14-1"><a href="#cite_note-block_numbers-14">[8]</a></sup>
</td>
<td><a href="/wiki/Cape_Canaveral_Space_Force_Station" title="Cape Canaver
al Space Force Station">CCAFS</a>,<br/><a href="/wiki/Cape_Canaveral_Space
_Launch_Complex_40" title="Cape Canaveral Space Launch Complex 40">SLC-40
</a>
</td>
<td><a href="/wiki/SpaceX_Dragon" title="SpaceX Dragon">Dragon</a> <a clas
s="mw-redirect" href="/wiki/COTS_Demo_Flight_1" title="COTS Demo Flight
1">demo flight C1</a><br/>(Dragon C101)
</td>
<td>
</td>
<td><a href="/wiki/Low_Earth_orbit" title="Low Earth orbit">LEO</a> (<a hr
ef="/wiki/International_Space_Station" title="International Space Statio
n">ISS</a>)
</td>
<td><div class="plainlist">
<ul><li><a href="/wiki/NASA" title="NASA">NASA</a> (<a href="/wiki/Commerc
ial_Orbital_Transportation_Services" title="Commercial Orbital Transportat
ion Services">COTS</a>)</li>
<li><a href="/wiki/National_Reconnaissance_Office" title="National Reconna
issance Office">NRO</a></li></ul>
</div>
</td>
<td class="table-success" style="background: #9EFF9E; vertical-align: midd
le; text-align: center;">Success<sup class="reference" id="cite_ref-ns2011
0930_15-1"><a href="#cite_note-ns20110930-15">[9]</a></sup>
</td>
<td class="table-failure" style="background: #FFC7C7; vertical-align: midd
le; text-align: center;">Failure<sup class="reference" id="cite_ref-ns2011
0930_15-2"><a href="#cite_note-ns20110930-15">[9]</a></sup><sup class="ref
erence" id="cite_ref-20"><a href="#cite_note-20">[14]</a></sup><br/><small
>(parachute)</small>
</td></tr>
<tr>
<td colspan="9">Maiden flight of <a class="mw-redirect" href="/wiki/Dragon
_capsule" title="Dragon capsule">Dragon capsule</a>, consisting of over 3
hours of testing thruster maneuvering and reentry.<sup class="reference" i
d="cite_ref-spaceflightnow_Clark_unleashing_Dragon_21-0"><a href="#cite_no
te-spaceflightnow_Clark_unleashing_Dragon-21">[15]</a></sup> Attempted to
```

recover the first stage by parachuting it into the ocean, but it disintegrated upon reentry, before the parachutes were deployed.^{[12]} <small>(more details below)</small> It also included two CubeSats,^{[16]} and a wheel of Brouère cheese.
</td></tr>
<tr>
<th rowspan="2" scope="row" style="text-align:center;">3
</th>
<td>22 May 2012,
07:44^{[17]}
</td>
<td>F9 v1.0^{[7]}
B0005.1^{[8]}
</td>
<td>CCAFS,
SLC-40
</td>
<td>Dragon demo flight C2+^{[18]}
(Dragon C102)
</td>
<td>525 kg (1,157 lb)^{[19]}
</td>
<td>LEO (ISS)
</td>
<td>NASA (COTS)
</td>
<td class="table-success" style="background: #9EFF9E; vertical-align: middle; text-align: center;">Success^{[20]}
</td>
<td class="table-noAttempt" style="background: #EEE; vertical-align: middle; white-space: nowrap; text-align: center;">No attempt
</td></tr>
<tr>
<td colspan="9">Dragon spacecraft demonstrated a series of tests before it was allowed to approach the International Space Station. Two days later, it became the first commercial spacecraft to board the ISS.^{[17]} <small>(more details below)</small>
</td></tr>
<tr>
<th rowspan="3" scope="row" style="text-align:center;">4
</th>

Date and time (UTC)	Version, Booster	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Booster landing
8 October 2012, 00:35[21]	F9 v1.0[7] B0006.1[8]	CCAFS, SLC-40	SpaceX CRS-1[22] (Dragon C103)	4,700 kg (10,400 lb)	LEO (ISS)	NASA (CRS)	Success	No attempt
			Orbcomm-OG2[23]	172 kg (379 lb)[24]	LEO	Orbcomm	Partial failure[25]	
CRS-1 was successful, but the secondary payload was inserted into an abnormally low orbit and subsequently lost. This was due to one of the nine Merlin engines shutting down during the launch, and NASA declining a second reignition, as per ISS visiting vehicle safety rules, the primary payload owner is contractually allowed to decline a second reignition. NASA stated that this was because SpaceX could not guarantee a high enough likelihood of the second stage completing the second burn successfully which was required to avoid any risk of secondary payload's collision with the ISS.								

```
="reference" id="cite_ref-OrbcommTotalLoss_32-0"><a href="#cite_note-Orbco
mmTotalLoss-32">[26]</a></sup><sup class="reference" id="cite_ref-sn201210
11_33-0"><a href="#cite_note-sn20121011-33">[27]</a></sup><sup class="refe
rence" id="cite_ref-34"><a href="#cite_note-34">[28]</a></sup>
</td></tr>
<tr>
<th rowspan="2" scope="row" style="text-align:center;">5
</th>
<td>1 March 2013,<br/>15:10
</td>
<td><a href="/wiki/Falcon_9_v1.0" title="Falcon 9 v1.0">F9 v1.0</a><sup cl
ass="reference" id="cite_ref-MuskMay2012_13-4"><a href="#cite_note-MuskMay
2012-13">[7]</a></sup><br/>B0007.1<sup class="reference" id="cite_ref-bloc
k_numbers_14-4"><a href="#cite_note-block_numbers-14">[8]</a></sup>
</td>
<td><a href="/wiki/Cape_Canaveral_Space_Force_Station" title="Cape Canaver
al Space Force Station">CCAFS</a>,<br/><a href="/wiki/Cape_Canaveral_Space
_Launch_Complex_40" title="Cape Canaveral Space Launch Complex 40">SLC-40
</a>
</td>
<td><a href="/wiki/SpaceX_CRS-2" title="SpaceX CRS-2">SpaceX CRS-2</a><sup
class="reference" id="cite_ref-sxManifest20120925_28-1"><a href="#cite_not
e-sxManifest20120925-28">[22]</a></sup><br/>(Dragon C104)
</td>
<td>4,877 kg (10,752 lb)
</td>
<td><a href="/wiki/Low_Earth_orbit" title="Low Earth orbit">LEO</a> (<a cl
ass="mw-redirect" href="/wiki/ISS" title="ISS">ISS</a>)
</td>
<td><a href="/wiki/NASA" title="NASA">NASA</a> (<a href="/wiki/Commercial_
Resupply_Services" title="Commercial Resupply Services">CRS</a>)
</td>
<td class="table-success" style="background: #9EFF9E; vertical-align: midd
le; text-align: center;">Success
</td>
<td class="table-noAttempt" style="background: #EEE; vertical-align: middl
e; white-space: nowrap; text-align: center;">No attempt
</td></tr>
<tr>
<td colspan="9">Last launch of the original Falcon 9 v1.0 <a href="/wiki/L
aunch_vehicle" title="Launch vehicle">launch vehicle</a>, first use of the
unpressurized trunk section of Dragon.<sup class="reference" id="cite_ref-
sxf9_20110321_35-0"><a href="#cite_note-sxf9_20110321-35">[29]</a></sup>
</td></tr>
<tr>
<th rowspan="2" scope="row" style="text-align:center;">6
</th>
<td>29 September 2013,<br/>16:00<sup class="reference" id="cite_ref-pa2013
0930_36-0"><a href="#cite_note-pa20130930-36">[30]</a></sup>
</td>
<td><a href="/wiki/Falcon_9_v1.1" title="Falcon 9 v1.1">F9 v1.1</a><sup cl
ass="reference" id="cite_ref-MuskMay2012_13-5"><a href="#cite_note-MuskMay
2012-13">[7]</a></sup><br/>B1003<sup class="reference" id="cite_ref-block_
numbers_14-5"><a href="#cite_note-block_numbers-14">[8]</a></sup>
</td>
<td><a class="mw-redirect" href="/wiki/Vandenberg_Air_Force_Base" title="V
andenberg Air Force Base">VAFB</a>,<br/><a href="/wiki/Vandenberg_Space_La
unch_Complex_4" title="Vandenberg Space Launch Complex 4">SLC-4E</a>
</td>
<td><a href="/wiki/CASSIOPE" title="CASSIOPE">CASSIOPE</a><sup class="refe
rence" id="cite_ref-sxManifest20120925_28-2"><a href="#cite_note-sxManifes
```

```
t20120925-28">[22]</a></sup><sup class="reference" id="cite_ref-CASSIOPE_M
DA_37-0"><a href="#cite_note-CASSIOPE_MDA-37">[31]</a></sup>
</td>
<td>500 kg (1,100 lb)
</td>
<td><a href="/wiki/Polar_orbit" title="Polar orbit">Polar orbit</a> <a hre
f="/wiki/Low_Earth_orbit" title="Low Earth orbit">LEO</a>
</td>
<td><a href="/wiki/Maxar_Technologies" title="Maxar Technologies">MDA</a>
</td>
<td class="table-success" style="background: #9EFF9E; vertical-align: midd
le; text-align: center;">Success<sup class="reference" id="cite_ref-pa2013
0930_36-1"><a href="#cite_note-pa20130930-36">[30]</a></sup>
</td>
<td class="table-no2" style="background: #FFE3E3; color: black; vertical-a
lign: middle; text-align: center;">Uncontrolled<br/><small>(ocean)</small>
<sup class="reference" id="cite_ref-ocean_landing_38-0"><a href="#cite_not
e-ocean_landing-38">[d]</a></sup>
</td></tr>
<tr>
<td colspan="9">First commercial mission with a private customer, first la
unch from Vandenberg, and demonstration flight of Falcon 9 v1.1 with an im
proved 13-tonne to LEO capacity.<sup class="reference" id="cite_ref-sxf9_2
0110321_35-1"><a href="#cite_note-sxf9_20110321-35">[29]</a></sup> After s
eparation from the second stage carrying Canadian commercial and scientifi
c satellites, the first stage booster performed a controlled reentry,<sup
class="reference" id="cite_ref-39"><a href="#cite_note-39">[32]</a></sup>
and an <a href="/wiki/Falcon_9_first-stage_landing_tests" title="Falcon 9
first-stage landing tests">ocean touchdown test</a> for the first time. Th
is provided good test data, even though the booster started rolling as it
neared the ocean, leading to the shutdown of the central engine as the rol
l depleted it of fuel, resulting in a hard impact with the ocean.<sup clas
s="reference" id="cite_ref-pa20130930_36-2"><a href="#cite_note-pa20130930
-36">[30]</a></sup> This was the first known attempt of a rocket engine be
ing lit to perform a supersonic retro propulsion, and allowed SpaceX to en
ter a public-private partnership with <a href="/wiki/NASA" title="NASA">NA
SA</a> and its Mars entry, descent, and landing technologies research proj
ects.<sup class="reference" id="cite_ref-40"><a href="#cite_note-40">[33]
</a></sup> <small>(<a href="#Maiden_flight_of_v1.1">more details below</a
>)</small>
</td></tr>
<tr>
<th rowspan="2" scope="row" style="text-align:center;">7
</th>
<td>3 December 2013,<br/>22:41<sup class="reference" id="cite_ref-sfn_wwls
20130624_41-0"><a href="#cite_note-sfn_wwls20130624-41">[34]</a></sup>
</td>
<td><a href="/wiki/Falcon_9_v1.1" title="Falcon 9 v1.1">F9 v1.1</a><br/>B1
004
</td>
<td><a href="/wiki/Cape_Canaveral_Space_Force_Station" title="Cape Canaver
al Space Force Station">CCAFS</a>,<br/><a href="/wiki/Cape_Canaveral_Space
_Launch_Complex_40" title="Cape Canaveral Space Launch Complex 40">SLC-40
</a>
</td>
<td><a href="/wiki/SES-8" title="SES-8">SES-8</a><sup class="reference" id
="cite_ref-sxManifest20120925_28-3"><a href="#cite_note-sxManifest20120925
-28">[22]</a></sup><sup class="reference" id="cite_ref-spx-pr_42-0"><a hre
f="#cite_note-spx-pr-42">[35]</a></sup><sup class="reference" id="cite_ref
-aw20110323_43-0"><a href="#cite_note-aw20110323-43">[36]</a></sup>
</td>
```

```
<td>3,170 kg (6,990 lb)
</td>
<td><a href="/wiki/Geostationary_transfer_orbit" title="Geostationary tran
sfer orbit">GTO</a>
</td>
<td><a href="/wiki/SES_S.A." title="SES S.A.">SES</a>
</td>
<td class="table-success" style="background: #9EFF9E; vertical-align: midd
le; text-align: center;">Success<sup class="reference" id="cite_ref-SNMiss
ionStatus7_44-0"><a href="#cite_note-SNMissionStatus7-44">[37]</a></sup>
</td>
<td class="table-noAttempt" style="background: #EEE; vertical-align: middl
e; white-space: nowrap; text-align: center;">No attempt<br/><sup class="re
ference" id="cite_ref-sf10120131203_45-0"><a href="#cite_note-sf1012013120
3-45">[38]</a></sup>
</td></tr>
<tr>
<td colspan="9">First <a href="/wiki/Geostationary_transfer_orbit" title
="Geostationary transfer orbit">Geostationary transfer orbit</a> (GTO) lau
nch for Falcon 9,<sup class="reference" id="cite_ref-spx-pr_42-1"><a href
="#cite_note-spx-pr-42">[35]</a></sup> and first successful reignition of
the second stage.<sup class="reference" id="cite_ref-46"><a href="#cite_no
te-46">[39]</a></sup> SES-8 was inserted into a <a href="/wiki/Geostationa
ry_transfer_orbit" title="Geostationary transfer orbit">Super-Synchronous
Transfer Orbit</a> of 79,341 km (49,300 mi) in apogee with an <a href="/wi
ki/Orbital_inclination" title="Orbital inclination">inclination</a> of 20.
55° to the <a href="/wiki/Equator" title="Equator">equator</a>.
</td></tr></tbody></table>
```

You should able to see the columns names embedded in the table header elements `<th>` as follows:

```
<tr>
<th scope="col">Flight No.
</th>
<th scope="col">Date and<br/>time (<a href="/wiki/Coordinated_Universal_Time" ti
tle="Coordinated Universal Time">UTC</a>)
</th>
<th scope="col"><a href="/wiki/List_of_Falcon_9_first-stage_boosters" title="Lis
t of Falcon 9 first-stage boosters">Version,<br/>Booster</a> <sup class="referen
ce" id="cite_ref-booster_11-0"><a href="#cite_note-booster-11">[b]</a></sup>
</th>
<th scope="col">Launch site
</th>
<th scope="col">Payload<sup class="reference" id="cite_ref-Dragon_12-0"><a href
="#cite_note-Dragon-12">[c]</a></sup>
</th>
<th scope="col">Payload mass
</th>
<th scope="col">Orbit
</th>
<th scope="col">Customer
</th>
<th scope="col">Launch<br/>outcome
</th>
<th scope="col"><a href="/wiki/Falcon_9_first-stage_landing_tests" title="Falcon
9 first-stage landing tests">Booster<br/>landing</a>
</th></tr>
```

Next, we just need to iterate through the `<th>` elements and apply the provided
`extract_column_from_header()` to extract column name one by one

In [10]:

```python
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a
 column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a lis
t called column_names
for x in first_launch_table.find_all('th'):
    name = extract_column_from_header(x)
    if name is not None and len(name) > 0:
        column_names.append(name)
```

Check the extracted column names

```
print(column_names)
```

```
['Flight No.', 'Date and time ( )', 'Launch site', 'Payload', 'Payload mas
s', 'Orbit', 'Customer', 'Launch outcome']
```

TASK 3: Create a data frame by parsing the launch HTML tables

We will create an empty dictionary with keys from the extracted column names in the previous task. Later, this dictionary will be converted into a Pandas dataframe

In [12]:

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

Next, we just need to fill up the `launch_dict` with launch records extracted from table rows.

Usually, HTML tables in Wiki pages are likely to contain unexpected annotations and other types of noises, such as reference links `B0004.1[8]`, missing values `N/A [e]`, inconsistent formatting, etc.

To simplify the parsing process, we have provided an incomplete code snippet below to help you to fill up the `launch_dict`. Please complete the following code snippet with TODOs or you can choose to write your own logic to parse all launch tables:

```python
extracted_row = 0
#Extract each table
for table_number,table in enumerate(response_BS.find_all('table',"wikitable plainrowhea
ders collapsible")):
    # get table row
     for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a num
ber
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
        if flag:
            extracted_row += 1
            # Flight Number value
            # TODO: Append the flight_number into launch_dict with key `Flight No.`
            #print(flight_number)
            datatimelist=date_time(row[0])

            # Date value
            # TODO: Append the date into launch_dict with key `Date`
            date = datatimelist[0].strip(',')
            #print(date)

            # Time value
            # TODO: Append the time into launch_dict with key `Time`
            time = datatimelist[1]
            #print(time)

            # Booster version
            # TODO: Append the bv into launch_dict with key `Version Booster`
            bv=booster_version(row[1])
            if not(bv):
                bv=row[1].a.string
            print(bv)

            # Launch Site
            # TODO: Append the bv into launch_dict with key `Launch Site`
            launch_site = row[2].a.string
            #print(launch_site)

            # Payload
            # TODO: Append the payload into launch_dict with key `Payload`
            payload = row[3].a.string
            #print(payload)

            # Payload Mass
            # TODO: Append the payload_mass into launch_dict with key `Payload mass`
            payload_mass = get_mass(row[4])
            #print(payload)

            # Orbit
            # TODO: Append the orbit into launch_dict with key `Orbit`
            orbit = row[5].a.string
```

```python
        #print(orbit)

        # Customer
        # TODO: Append the customer into launch_dict with key `Customer`
        customer = row[6].a.string
        #print(customer)

        # Launch outcome
        # TODO: Append the launch_outcome into launch_dict with key `Launch outcome
`

        launch_outcome = list(row[7].strings)[0]
        #print(launch_outcome)

        # Booster landing
        # TODO: Append the launch_outcome into launch_dict with key `Booster landin
g`

        booster_landing = landing_status(row[8])
        #print(booster_landing)




        #print(orbit)

        # Customer
        # TODO: Append the customer into launch_dict with key `Customer`
        customer = row[6].a.string
        #print(customer)

        # Launch outcome
        # TODO: Append the launch_outcome into launch_dict with key `Launch outcome
```

```
F9 v1.0B0003.1
F9 v1.0B0004.1
F9 v1.0B0005.1
F9 v1.0B0006.1
F9 v1.0B0007.1
F9 v1.1B1003
F9 v1.1
F9 v1.1
F9 v1.1
F9 v1.1
F9 v1.1
F9 v1.1
F9 v1.1
F9 v1.1
F9 v1.1
F9 v1.1
F9 FT
F9 v1.1
F9 FT
F9 FT
F9 FT
F9 FT
F9 FT
F9 FT
F9 FT
F9 FT
F9 FT
F9 FT
F9 FT⛏
F9 FT
F9 FT
F9 FT
F9 FTB1029.2
F9 FT
F9 FT
F9 B4
F9 FT
F9 B4
F9 B4
F9 FTB1031.2
F9 B4
F9 FTB1035.2
F9 FTB1036.2
F9 B4
F9 FTB1032.2
F9 FTB1038.2
F9 B4
F9 B4B1041.2
F9 B4B1039.2
F9 B4
F9 B5B1046.1
F9 B4B1043.2
F9 B4B1040.2
F9 B4B1045.2
F9 B5
F9 B5B1048
F9 B5B1046.2
F9 B5
```

```
F9 B5B1048.2
F9 B5B1047.2
F9 B5B1046.3
F9 B5
F9 B5
F9 B5B1049.2
F9 B5B1048.3
F9 B5[268]
F9 B5
F9 B5B1049.3
F9 B5B1051.2
F9 B5B1056.2
F9 B5B1047.3
F9 B5
F9 B5
F9 B5B1056.3
F9 B5
F9 B5
F9 B5
F9 B5
F9 B5
F9 B5
F9 B5
F9 B5
F9 B5
F9 B5
F9 B5B1058.2
F9 B5
F9 B5B1049.6
F9 B5
F9 B5B1060.2
F9 B5B1058.3
F9 B5B1051.6
F9 B5
F9 B5
F9 B5
F9 B5
F9 B5 ♻
F9 B5 ♻
F9 B5 ♻
F9 B5 ♻
F9 B5
F9 B5B1051.8
F9 B5B1058.5
```

-
AttributeError   Traceback (most recent call las
t)
<ipython-input-14-fcb50b897f15> in <module>
     60   # Customer
     61   # TODO: Append the customer into launch_dict with key
 `Customer`
---> 62   customer = row[6].a.string
     63   #print(customer)
     64

AttributeError: 'NoneType' object has no attribute 'string'

After you have fill in the parsed launch record values into `launch_dict`, you can create a dataframe from it.

In [15]:

```
df=pd.DataFrame(launch_dict)
```

We can now export it to a **CSV** for the next section, but to make the answers consistent and in case you have difficulties finishing this lab.

Following labs will be using a provided dataset to make each lab independent.

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Authors

Yan Luo (https://www.linkedin.com/in/yan-luo-96288783/?
utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id:
SkillsNetwork-Channel-SkillsNetworkCoursesIBMDS0321ENSkillsNetwork26802033-2021-01-01)

◄ ▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬ ▶

Nayef Abou Tayoun (https://www.linkedin.com/in/nayefaboutayoun/?
utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id:
SkillsNetwork-Channel-SkillsNetworkCoursesIBMDS0321ENSkillsNetwork26802033-2021-01-01)

◄ ▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬ ▶

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2021-06-09	1.0	Yan Luo	Tasks updates
2020-11-10	1.0	Nayef	Created the initial version