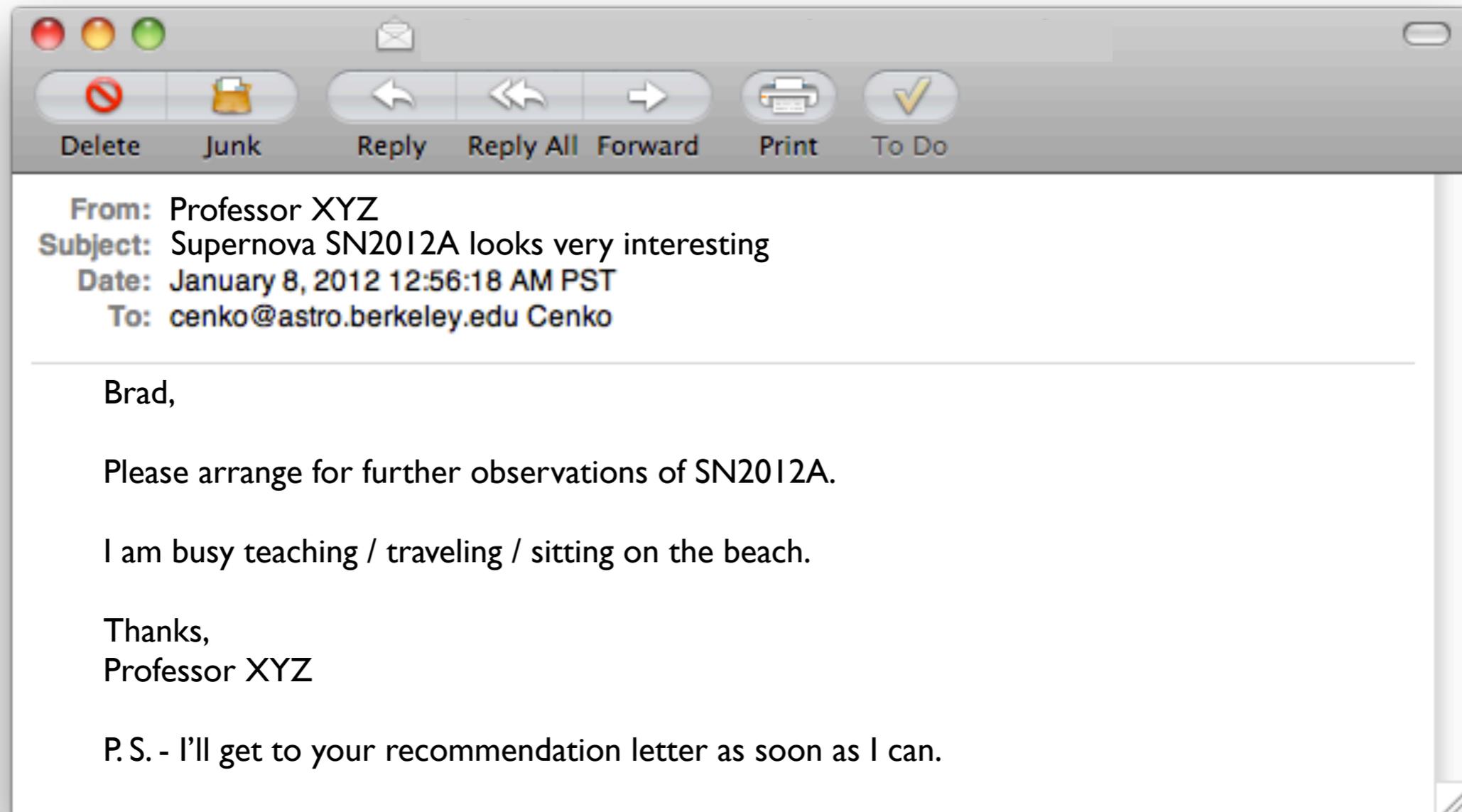


Whetting Your Appetite: Interacting with the Outside World

Brief Outline

- Interweb ([urllib](#), [urllib2](#), [etree](#))
- Relational Databases ([sqlite3](#))
- Email ([email](#), [smtplib](#))
- Science (at least my version of it)

How Science Gets Done: I



How Science Gets Done: II

- Write a python script `do_science`:
 - Extract additional information about a supernova from a webpage
 - Select a random graduate student from an sqlite database
 - Email the graduate student to request the observations

`urllib` & `urllib2`

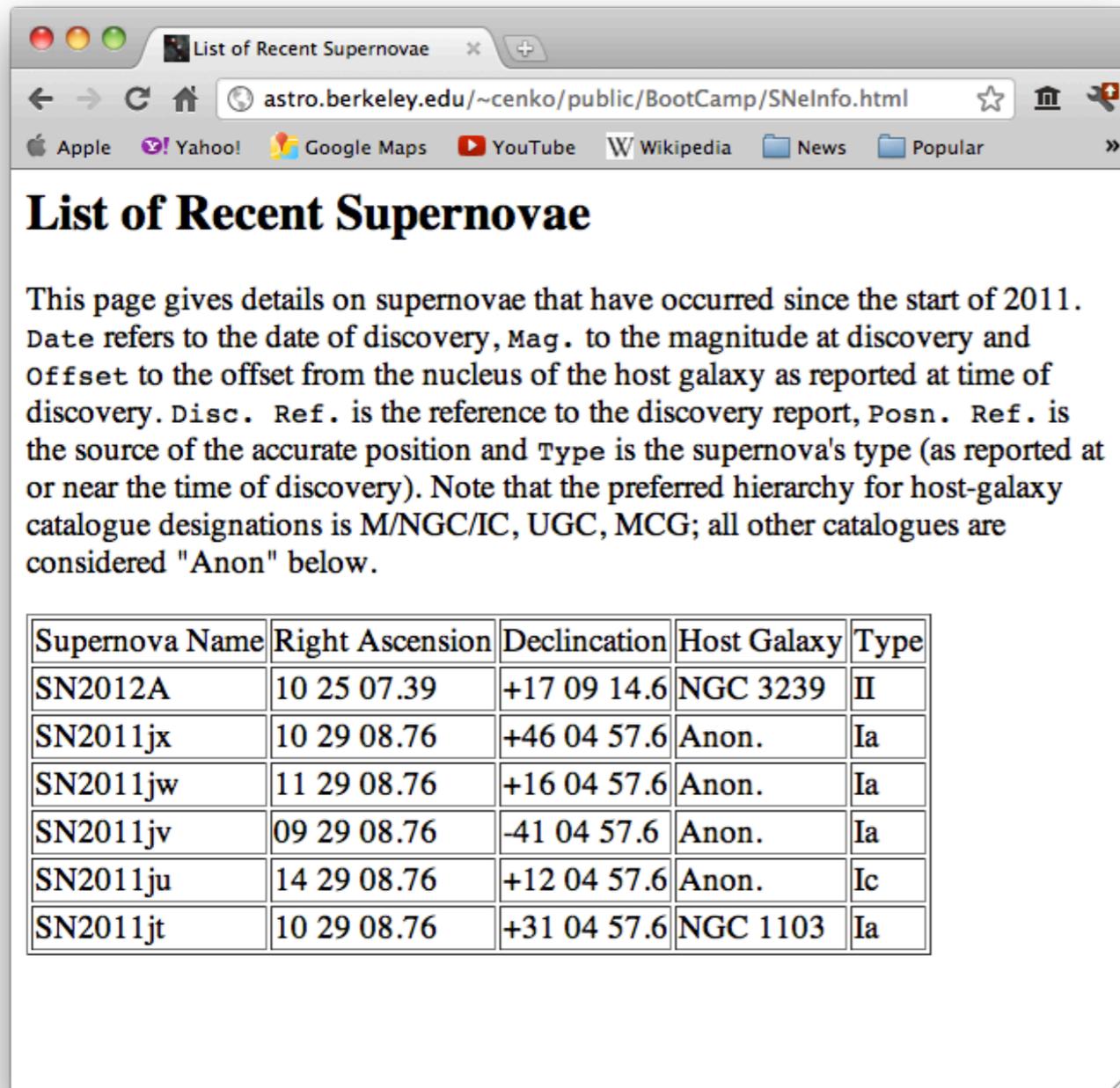
These modules provide access to any URL (uniform resource locator), the most common URL scheme being HTTP. (Others are HTTPS, FTP, FTPS, etc.)

The `urllib` module provides tools and functions for high-level, but less modern, interactions.

The `urllib2` module is more suited for complex interactions, supporting basic and digest authentication, redirections, cookies, and more.

`urllib.urlopen()` is deprecated in favor of
`urllib2.urlopen()`

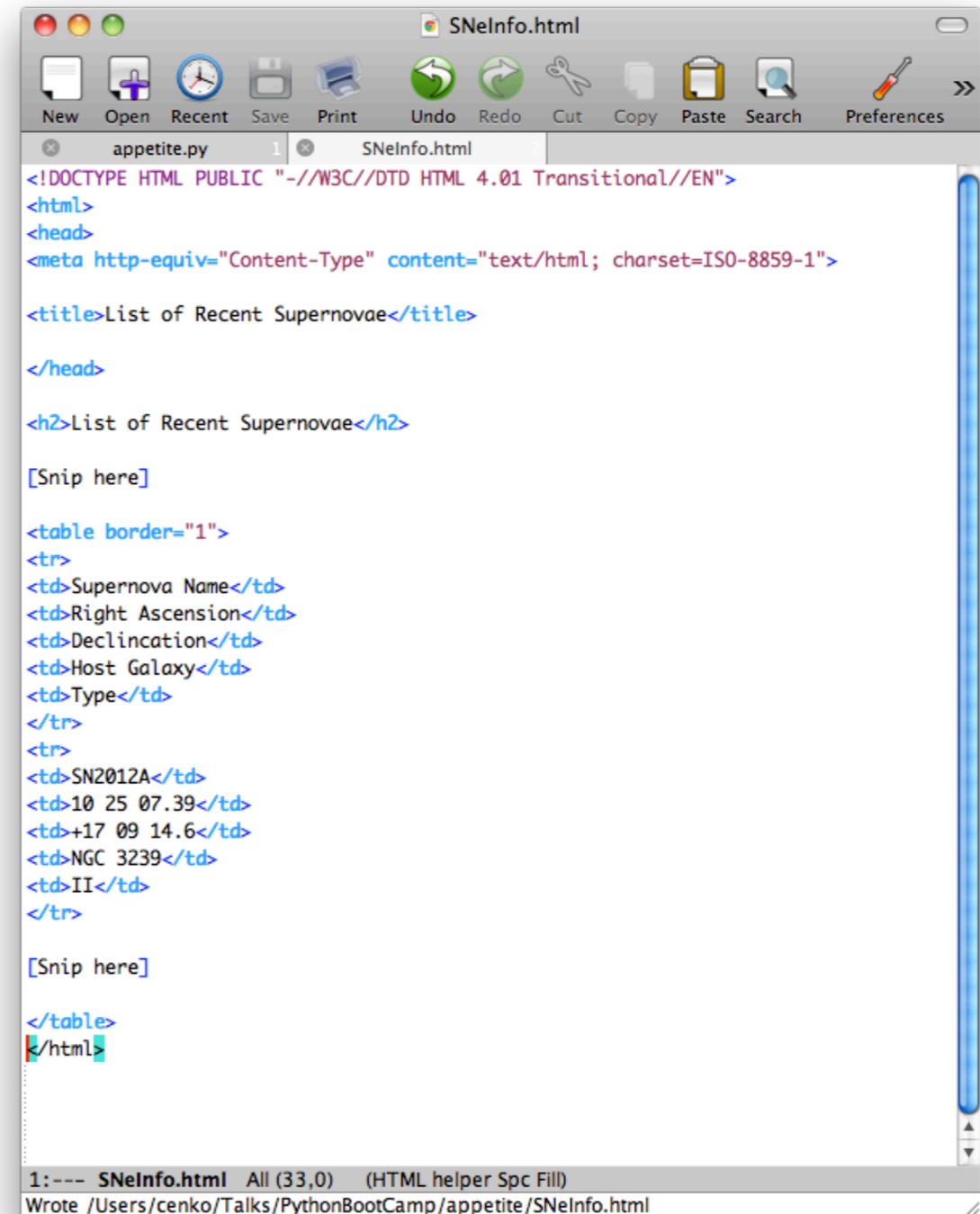
A (Simple) Webpage to Parse



List of Recent Supernovae

This page gives details on supernovae that have occurred since the start of 2011. Date refers to the date of discovery, Mag. to the magnitude at discovery and Offset to the offset from the nucleus of the host galaxy as reported at time of discovery. Disc. Ref. is the reference to the discovery report, Posn. Ref. is the source of the accurate position and Type is the supernova's type (as reported at or near the time of discovery). Note that the preferred hierarchy for host-galaxy catalogue designations is M/NGC/IC, UGC, MCG; all other catalogues are considered "Anon" below.

Supernova Name	Right Ascension	Declination	Host Galaxy	Type
SN2012A	10 25 07.39	+17 09 14.6	NGC 3239	II
SN2011jx	10 29 08.76	+46 04 57.6	Anon.	Ia
SN2011jw	11 29 08.76	+16 04 57.6	Anon.	Ia
SN2011jv	09 29 08.76	-41 04 57.6	Anon.	Ia
SN2011ju	14 29 08.76	+12 04 57.6	Anon.	Ic
SN2011jt	10 29 08.76	+31 04 57.6	NGC 1103	Ia



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>List of Recent Supernovae</title>

</head>

<h2>List of Recent Supernovae</h2>

[Snip here]

<table border="1">
<tr>
<td>Supernova Name</td>
<td>Right Ascension</td>
<td>Declination</td>
<td>Host Galaxy</td>
<td>Type</td>
</tr>
<tr>
<td>SN2012A</td>
<td>10 25 07.39</td>
<td>+17 09 14.6</td>
<td>NGC 3239</td>
<td>II</td>
</tr>
[Snip here]

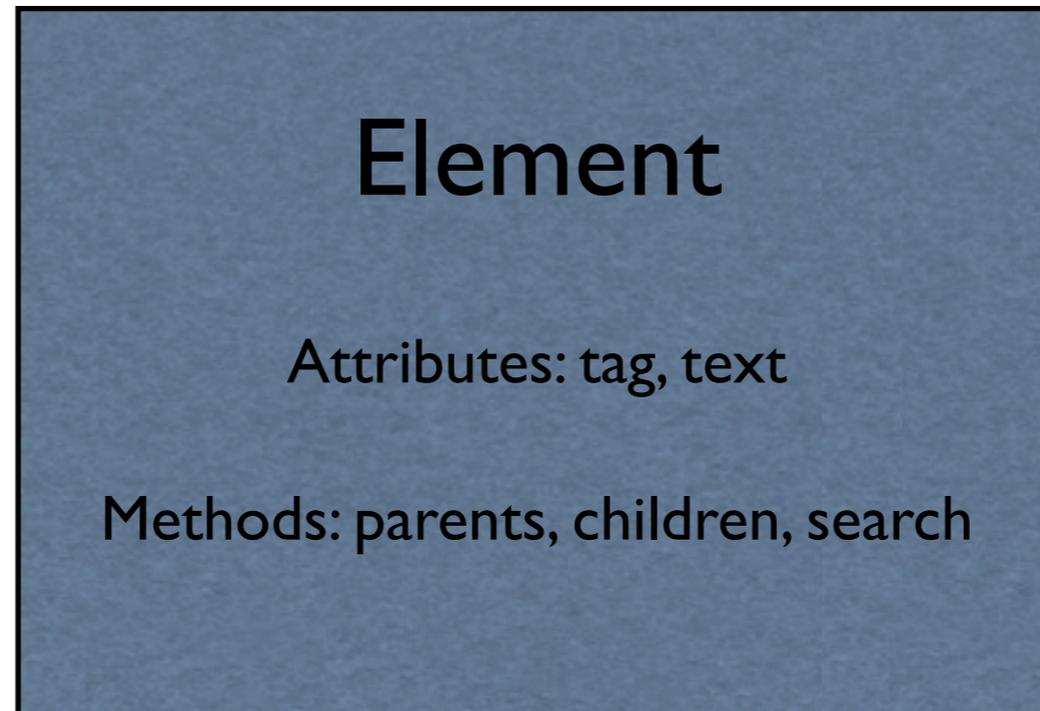
</table>
</html>
```

HowTo Retrieve HTML

```
>>> import urllib2
>>> MYSNURL = "http://astro.berkeley.edu/~cenko/public/BootCamp/SNeInfo.html"
>>>
>>> flob = urllib2.urlopen(MYSNURL)
>>> s = flob.read()
>>> flob.close()
```

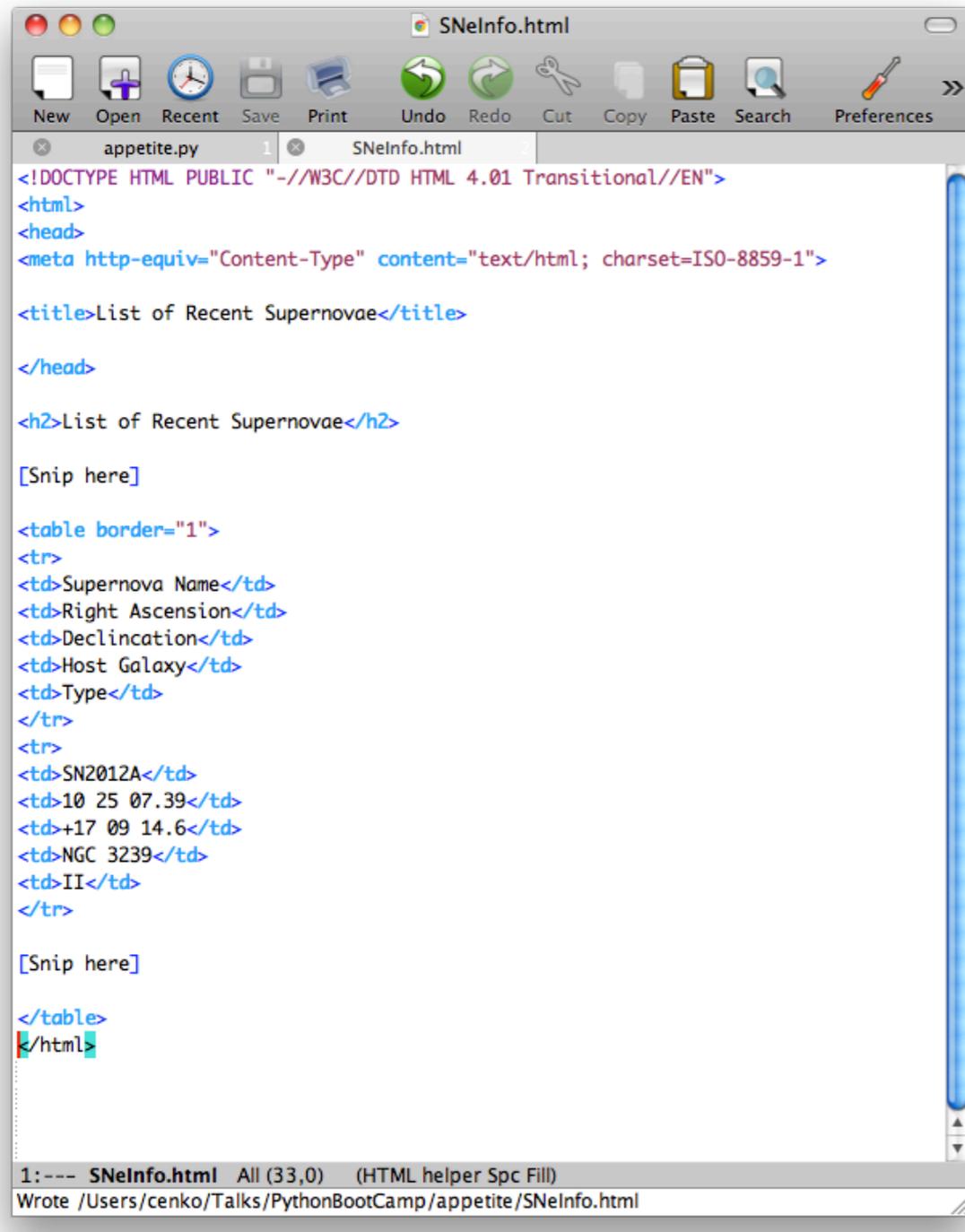
`urlopen` returns a file-like object, which can be read like any other file. As a result, `s` stores the HTML from the page in a (large) string

ElementTree (`lxml.etree`)

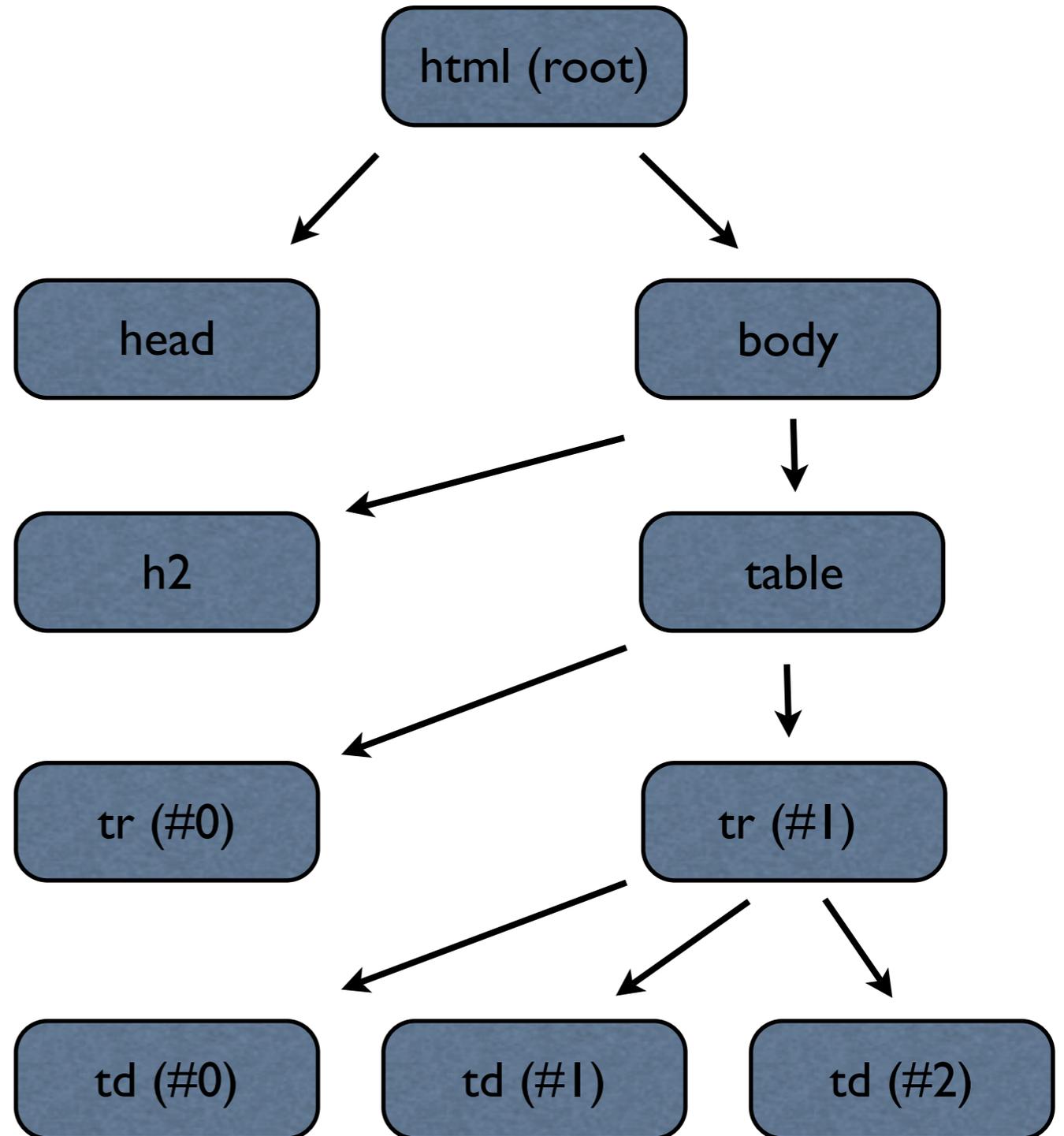


ElementTree provides a *class* that makes it extremely convenient to handle XML (and thus HTML) files. Element *instances* are used to represent each XML tag, with appropriate hierarchical relationships, and can be accessed with list syntax.

ElementTree (lxml.etree)



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>List of Recent Supernovae</title>
</head>
<h2>List of Recent Supernovae</h2>
[Snip here]
<table border="1">
<tr>
<td>Supernova Name</td>
<td>Right Ascension</td>
<td>Declination</td>
<td>Host Galaxy</td>
<td>Type</td>
</tr>
<tr>
<td>SN2012A</td>
<td>10 25 07.39</td>
<td>+17 09 14.6</td>
<td>NGC 3239</td>
<td>II</td>
</tr>
[Snip here]
</table>
</html>
```

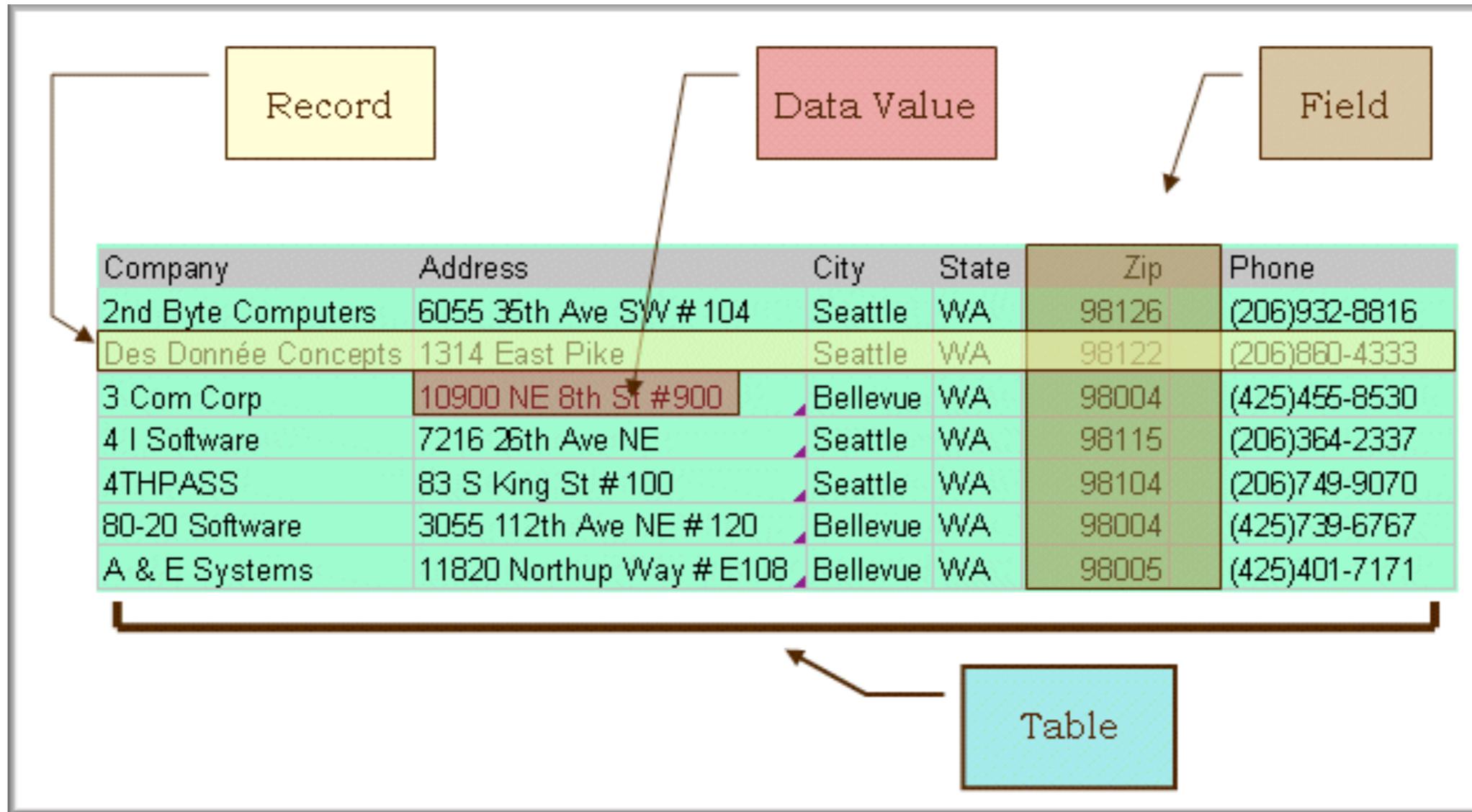


Parsing HTML with `etree`

```
>>> from lxml import etree
>>> s = flob.read()
>>> html = etree.HTML(s)
>>> rows = html.find('.//table')
>>> for row in rows:
...     if (sn_name == row[0].text):
...         coords = [row[1].text.replace(" ", ":"),
...                   row[2].text.replace(" ", ":")]
...         host = row[3].text
...         sntype = row[4].text
...         return [host, coords, sntype]
...
...
```

HTML converts a string into an `ElementTree` object. `find` identifies all the children of the first element named `table`. List-like indexing operations allow you to access children of nodes.

Relational Databases



Relational databases are an efficient (searchable) way to store tabular data. Most people today use some form of SQL (MySQL, PostGreSQL, etc.)

sqlite3 Overview

- Built-in SQL database access
- Database is stored as a file (or in RAM)
- Syntax similar to MySQLdb
- Not portable (machine-dependent)

HowTo Create an sqlite Database

```
>>> import sqlite3
>>> filename = "/Users/cenko/Talks/PythonBootCamp/appetite/astropeeps.sql"
>>> conn = sqlite3.connect(filename)
>>> c = conn.cursor()
>>> c.execute('''CREATE TABLE ASTROPEEPS (f_name text, l_name text,
...          email text, status text)''')
>>> c.execute('''INSERT INTO ASTROPEEPS VALUES ("Josh", "Bloom",
...          "jbloom@astro.berkeley.edu", "Faculty")''')
>>> c.execute('''INSERT INTO ASTROPEEPS VALUES ("Adam", "Morgan",
...          "amorgan@astro.berkeley.edu", "Student")''')
[etc.]
>>> conn.commit()
>>> c.close()
```

SQL commands are contained within the `execute` statement. Make sure to remember to commit the changes to the database before closing.

HowTo Query an sqlite Database

```
>>> filename = "/Users/cenko/Talks/PythonBootCamp/appetite/astropeeps.sql"
>>> conn = sqlite3.connect(filename)
>>> c = conn.cursor()
>>> student = "Student"
>>>
>>> c.execute("SELECT f_name, l_name, email FROM ASTROPEEPS WHERE status" + \
              " = '%s' ORDER BY RANDOM() LIMIT 1" % student)
<sqlite3.Cursor object at 0x10232e730>
>>> row = c.fetchall()
>>> print row
[(u'Adam', u'Morgan', u'amorgan@astro.berkeley.edu')]
>>> conn.commit()
>>> c.close()
>>> return [row[0][0], row[0][1], row[0][2]]
```

After `execute`, we need to perform a `fetchall` in order to retrieve the result from the query.

HowTo Email: I

```
>>> from email.MIMEMultipart import MIMEMultipart
>>> from email.MIMEText import MIMEText
>>> import NothingToSeeHere # Email password stored in this (private) file
>>> import smtplib
>>>
>>> [address, f_name, l_name] = ["amorgan@astro.berkeley.edu", "Adam", "Morgan"]
>>> [sn_name, host, coords, sntype] =
...  ["SN2012A", "M31", ["10:00:00.00", "+31:00:00.0"], "Ic"]
>>> myemail = "bradcenko@gmail.com"
>>>
>>> msg = MIMEMultipart()
>>> msg["From"] = myemail
>>> msg["To"] = address
```

Basic email functionalities are in the [email](#) and [smtplib](#) modules. `MIMEMultipart()` will create a new instance of a message.

HowTo Email: II

```
>>> msgstr = "Hi %s %s,\n\n" % (f_name, l_name)
>>> msgstr += "I just found out about %s, and it seems neat. " % sn_name
>>> if (host == None):
...     msgstr += "The host galaxy is unknown. "
... else:
...     msgstr += "The host galaxy is %s. " % host
...
>>> if (coords == None):
...     msgstr += "I do not know the coordinates. "
... else:
...     msgstr += "The location is: RA=%s; Dec=%s. " % (coords[0], coords[1])
...
>>> if (sntype == None):
...     msgstr += "I do not know the type.\n\n"
... else:
...     msgstr += "The type is %s.\n\n" % sntype
...
>>> msgstr += "Could you please arrange some new observations? "
>>> msgstr += "I am really busy drinking right now.\n\n"
>>> msgstr += "Thanks,\nBrad"
>>> msg.attach(MIMEText(msgstr))
```

HowTo Email: III

```
>>> mailServer = smtplib.SMTP("smtp.gmail.com", 587)
>>> mailServer.starttls()
>>> mailServer.login(myemail, NothingToSeeHere.passwd)
>>>
>>> mailServer.sendmail(myemail, address, msg.as_string())
>>> mailServer.close()
```

`sendmail` is a method of the `mailServer` object.

Putting it all together

```
>>> def do_science(sn_name, filename=ASTROPEEPSDB, url=MYSNURL,
...                 myemail="bradcenko@gmail.com"):
...
...     # See if the department database exists.  If not, create it.
...     if not os.path.exists(filename):
...         create_astro_table(filename=filename)
...
...     # Select a random graduate student to do our bidding
...     [f_name, l_name, address] =
...         retrieve_random_gradstudent(filename=filename)
...
...     # Find out some information about the supernova
...     [host, coords, sntype] = retrieve_sn_info(sn_name, url=url)
...
...     # Email the student
...     email_student(address, f_name, l_name, sn_name, host, coords, sntype,
...                   myemail=myemail)
...
...     print "I emailed %s %s at %s about %s." %
...           (f_name, l_name, address, sn_name)
...
...     # Faculty job here I come!
...     return
```

Breakout Problem

- The arXiv (arXiv.org) is a pre-print repository for a variety of fields in science / engineering. Write a script to search through the new postings of one particular field, returning a list of titles, authors and abstracts that match a (provided) keyword.
- <http://arxiv.org/astro-ph/new>

Astrophysics

New submissions

Submissions received from Wed 11 Jan 12 to Thu 12 Jan 12, announced Fri, 13 Jan 12

- New submissions
- Cross-lists
- Replacements

[total of 77 entries: 1-77]
[showing up to 2000 entries per page: fewer | more]

New submissions for Fri, 13 Jan 12

[1] [arXiv:1201.2404](#) [pdf, ps, other]

Photoevaporating Proplyd-like objects in Cygnus OB2
Nicholas J. Wright, Jeremy J. Drake, Janet E. Drew, Mario G. Guarcello, Robert A. Gutermuth, Joseph L. Hora, Kathleen E. Kraemer
Comments: Accepted for publication in ApJL
Subjects: Solar and Stellar Astrophysics (astro-ph.SR)

We report the discovery of ten proplyd-like objects in the vicinity of the massive OB association Cygnus OB2. They were discovered in IPHAS H-Alpha images and are clearly resolved in broad-band HST/ACS, near-IR and Spitzer mid-IR images. All exhibit the familiar tadpole shape seen in photoevaporating objects such as the Orion proplyds, with a bright ionization front at the head facing the central cluster of massive stars, and a tail stretching in the opposite direction. Many also show secondary ionization fronts, complex tail morphologies or multiple heads. We consider the evidence that these are either proplyds or 'evaporating gaseous globules' (EGGs) left over from a fragmenting molecular cloud, but find that neither scenario fully explains the observations. Typical sizes are 50,000-100,000 AU, larger than the Orion proplyds, but in agreement with the theoretical scaling of proplyd size with distance from the ionizing source. These objects are located at projected separations of 6-14pc from the OB association, compared to 0.1pc for the Orion proplyds, but are clearly being photoionized by the 65 O-type stars in Cyg OB2. Central star candidates are identified in near- and mid-IR images, supporting the proplyd scenario, though their large sizes and notable asymmetries is more consistent with the EGG scenario. A third possibility is therefore considered, that these are a unique class of photoevaporating partially-embedded young stellar objects that have survived the destruction of their natal molecular cloud. This has implications for the properties of stars that form in the vicinity of massive stars.

[2] [arXiv:1201.2405](#) [pdf, other]

The IRX-beta relation on sub-galactic scales in star-forming galaxies of the Herschel Reference Survey
M. Boquien, V. Buat, A. Boselli, M. Baes, G. J. Bendo, L. Ciesla, A. Cooray, L. Cortese, S. Eales, G. Gavazzi, H. L. Gomez, V. Lebouteiller, C. Pappalardo, M. Pohlen, M. W. L. Smith, L. Spinoglio
Comments: Accepted for publication in A&A, 18 pages, 14 figures. The paper with high resolution figures can be downloaded at [this http URL](#)
Subjects: Cosmology and Extragalactic Astrophysics (astro-ph.CO); Galaxy Astrophysics (astro-ph.GA)

UV and optical surveys are essential to gain insight into the processes driving galaxy formation and evolution. The rest-frame UV emission is key to measure the cosmic SFR. However, UV light is strongly reddened by dust. In starburst galaxies, the UV colour and the attenuation are linked, allowing to correct for dust extinction. Unfortunately, evidence has been accumulating that the relation between UV colour and attenuation is different for normal star-forming galaxies when compared to starburst galaxies. It is still not understood why star-forming galaxies deviate from the UV colour-attenuation relation of starburst galaxies. Previous work and models hint that the role of the shape of the attenuation curve and the age of stellar populations have an important role. In this paper we aim at understanding the fundamental reasons to explain this deviation. We have used the CIGALE SED fitting code to model the far UV to the far IR emission of a set of 7 reasonably face-on spiral galaxies from the HRS. We have explored the influence of a wide range of physical parameters to quantify their influence and impact on the accurate determination of the attenuation from the UV colour, and why normal galaxies do not follow the same relation as starburst galaxies. We have found that the deviation can be best explained by intrinsic UV colour differences between different regions in galaxies. Variations in the shape of the attenuation curve can also play a secondary role. Standard age estimators of the stellar populations prove to be poor predictors of the intrinsic UV colour. These results are also retrieved on a sample of 58 galaxies when considering their integrated fluxes. When correcting the emission of normal star-forming galaxies for the attenuation, it is crucial to take into account possible variations in the intrinsic UV colour as well as variations of the shape of the attenuation curve.

[3] [arXiv:1201.2406](#) [pdf, other]

How the merger of two white dwarfs depends on their mass ratio: orbital stability and detonations at contact
Marius Dan (1), Stephan Rosswog (1 and 2), James Guillochon (2), Enrico Ramirez-Ruiz (2) ((1) Jacobs University Bremen, (2) UC Santa Cruz)
Comments: 13 pages, 10 figures, submitted to MNRAS
Subjects: High Energy Astrophysical Phenomena (astro-ph.HE)

Despite their unique astrophysical relevance, the outcome of white dwarf binary mergers has so far only been studied for a very restricted number of systems. Here we present the results of a survey with more than two hundred simulations systematically scanning the white dwarf binary parameter space. We consider white dwarf masses ranging from 0.2 to 1.2 M_{\odot} and account for their different chemical compositions. We find excellent agreement with the orbital evolution predicted by mass transfer stability analysis.

Breakout Problem

```
>>> ARXIVURL = "http://arxiv.org/list/astro-ph/new"
>>> import bout_appetite
>>> matches = bout_appetite.search_arXiv("Cenko", url=ARXIVURL, printres=True)
Sorry, no matches were found!
>>> matches = bout_appetite.search_arXiv("polarization")
Title: Integrated Optical Polarization of Nearby Galaxies
Authors: Amy Jones, Lifan Wang, Kevin Krisciunas, Emily Freeland
Abstract: We performed an integrated optical polarization survey of 70 nearby galaxies to study the relationship between linear polarization and galaxy properties. To date this is the largest survey of its kind. The data were collected at McDonald Observatory using the Imaging Grism Polarimeter on the Otto Struve 2.1m telescope. Most of the galaxies did not have significant level of linear polarization, where the bulk is <1%. A fraction of the galaxies showed a loose correlation between the polarization and position angle of the galaxy, indicating that dust scattering is the main source of optical polarization. The unbarred spiral galaxies are consistent with the predicted relationship with inclination from scattering models of  $\sim \sin^2 i$ .
```

Breakout Hints

```
>>> class Paper(object):
...     def __init__(self, paper_etree):
...         # Parse the element here
...         self.title =
...     def __str__(self):
...         # Make the instance print out nicely here
...
>>> def search_arXiv(keyword, url=ARXIVURL, printres=True):
...     # Grab HTML from website
...     # Convert to element tree
...     # Find parent elements for each paper
...     # Create a paper object from these elements
...     # Simple regular expression match
...     # Print results if desired
...     # return matches
```

Hint: You will have to take a relatively close look at the source code for the arXiv page of interest. Note that new papers are indicated by the tag `<div class="meta">`. You can search for this tag with: `html.findall(//div[@class="meta"])`