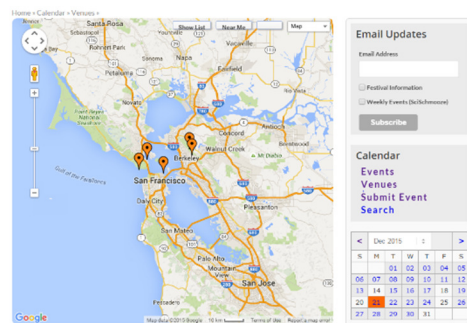


BAY AREA SCIENCE CALENDAR / BAY AREA SCIENCE FESTIVAL -- Notes on Maintaining a Science Event Website

Stats: (FOUNDING YEAR, USERS, STAFFING, TIME COMMITMENT / WEEK)

<http://www.bayareascience.org/calendar/>

[Key Contacts – Kishore Hari, director of the BASF, others]



Background from Kishore Hari, director of the Bay Area Science Festival:

The Science of Star Wars



Only a few lines of code could possibly analyze Star Wars: The Force Awakens for its science mistakes. But only a very special Caltech PhD computational astronomer or physicist (like Kishore Hari) could do so in a way that is both entertaining and educational. Kishore Hari, a physicist at Caltech, is the author of the book, and the author of the book.

Sponsored by Wikimedia and A&A Scientist

Speaker Seth Shostak, Center for SETI Research

Wednesday, 12/16/15

6:00 PM

Contact:

Website: [Click to Visit](#)

Cost:

\$2 Preferred seating

Save this Event:

Calendar

Google Calendar

Yahoo! Calendar

Windows Live Calendar

Share this Event:

Facebook

Twitter

LinkedIn

StumbleUpon

Reddit

Print

Map

Website

Google Maps

Map

Website

Google Maps

Map

Website

Google Maps

Map

Website

Google Maps

Map

Website

Google Maps

Map

Website

Google Maps

Map

Website

Google Maps

Map

Website

Google Maps

Map

Website

Google Maps

Map

Website

Google Maps

Map

Website

Google Maps

Map

Website

This calendar was started by the effort of volunteers from a science café. There was no place to go for information about local events, and there are over 100 each week. In 2008, Kishore and two others evaluated calendar software options and then customized the one they liked best. They have listed over 13,000 events over seven years and have had two million viewers. They send weekly newsletters, with an Editor's Pick for the Top 3 Events. They allow for submissions, with about half from organizations and about half found by the editors. Initially, they were to be only an information service, but have become a strong service tool with three benefits – 1) year-round touchpoint, 2) database of who is doing what in the area, and 3) gathered some data-driven metrics of the benefits.

How do you receive submssions?

Originally, we worked through Helios, which has now become an open-source program. Online submissions can be held for editor review and formatting, and we receive 15-20 each week. WordPress is used for our Festival site, and this program/format would be my recommendation to any group which is starting now. ... A site requires a huge amount of work at the startup to populate 500-600 events to become current. After this, the maintenance is manageable. Our editor is a volunteer. We use Mail Chimp to send the newsletter. So, our cost is zero. Early on, the costs were a licensing fee at \$149 per year and the time. The local PBS station calendar is populated by our calendar, "Science Events, brought to you by the Bay Area Science Festival."



Technical Notes on Maintaining a Science Event Website – More from the Bay Area

BIGGEST CHALLENGE: Finding events to list

The most difficult thing for us has been finding events to list on our calendar. We have many organizations in the area that offer science-based events, most with their own websites, newsletters, email blasts, etc. Finding new events and placing the information into our calendar in a format that works for us has led us to a somewhat automated solution described in this document.

Once upon a time there was Google Reader. Google Reader allowed you to place the URL for a web page into a list and Google would monitor the page and generate an RSS feed entry when something on that page changed. It worked pretty well. Google decided not enough people used this feature and they deleted it. So, we went in search of a similar solution, and we found a better one.

A SOLUTION: FEED43.COM

One drawback to Google Reader, and similar tools still available, is that they look at the entire page. Anything changing anywhere on the page causes an entry, and often it is a false alarm. **What was needed was something that allowed you to look at just a portion of a page. We found such a tool, called feed43.com (Feed for free).**

Feed43 requires at least a passing knowledge of HTML. It looks at the HTML code on a web url and it searches for user-defined patterns. For example, you can have it look at just the main portion of a page, not the side bars, headers, footers, etc., by specifying character strings found on the page that will serve as delimiters for the search. Then you can specify character strings that represent the items you want to extract from the page within those delimiters. This gives you a lot of power, but it comes with some drawbacks, too.

When feed43 determines that a change has been made to the monitored page, it generates one or more RSS feed items for that page. Each feed43 feed has a unique RSS address that feed43 creates. You define the content of the RSS items based on how much you extract from the page via your search parameters. You can then put the RSS feed into any RSS feed reader, including those built into browsers, and it will notify you when new items appear. **We use NetVibes Reader, but any one will do.**

ANOTHER CHALLENGE: NOT ALL SITES USE PERFECT HTML

Having looked at hundreds of web pages, we've learned that not all are created using good HTML techniques. Ideally each developer uses HTML as it was designed, namely as a method of identifying the parts of a page, such as headings, paragraphs, list elements, tables, links, etc. Unfortunately, many developers use HTML to provide the look they want, not to define the content of the.

Pages that follow good HTML coding are easy to monitor as the content is consistent. Each item in a list of events has identical HTML coding in it. This makes it easy for feed43 to pick out each item and to monitor for changes.

Pages that don't follow coding rules as well can still be monitored, but it may be more difficult. And any little change to the HTML or content that affects your global boundary or the item itself will cause feed43 to stop seeing changes.

BE SURE TO PERIODICALLY DOUBLE-CHECK FEEDS

Many websites are built using WYSIWYG packages. The end user never deals with HTML, just a front end that shows them what the end result will look like. The package generates the HTML, and usually it is consistent, although not always well coded. If the website installs an update to their package, it may change the HTML that is generated without the website owners even knowing it, and suddenly your RSS feed gets no updates. I just found one feed that has worked for years that stopped working on May 12th. A minor change to the HTML generated for each listing on their calendar caused the feed43 code to stop detecting items. A simple inspection of the page and an edit to the feed43 code solved the problem, but it means that **every so often you have to review your feeds to make sure they are still working and make adjustments as appropriate.**

OTHER ISSUES TO CONSIDER

Other design issues abound. Some websites only list in calendar format, and that can't be monitored. Some create lists of upcoming events, but list a finite number on a given page. This means you either have to create a feed43 search for each individual page, or just one for the top page and hope things percolate to that page early enough to get you timely listings. Some pages use java to provide information on the page real-time, and feed43 can't see that as it isn't HTML.

TIME INVESTMENT

How long does it take to maintain the calendar? It depends! During the summer it is fairly easy, but during the academic year it takes about an hour a day on average to sift through all the new listings. Since we can't filter for science events, we get a lot of event listings from local

universities that we aren't interested in. With a little practice you learn to skip over those just based on the titles.

A DEEPER DIVE INTO THE TECHNICAL ASPECTS OF USING FEED43.COM

Here's an example of a feed43 search. The California Academy of Sciences holds several series of events that we list. One such event is their Nightlife series. The URL is <http://www.calacademy.org/nightlife/>

The section of the page that we are interested in starts with the text "Next Event". Here's a screen capture of the feed43 form showing the Academy's web page source as well as our search patterns:

Below is the HTML source of the retrieved page. Use it to setup extraction rules (see next step).

Page Source: [\[?\]](#)

```
next-upcoming-tri skewed-tri-grid >

<h2 class="pane-title">Next Event</h2>

<div class="view view-nightlife-upcoming view-id-nightlife_upcoming view-display-id-next_upcoming_nl
view-dom-id-d7f3bf106de539365574eeb9511b1310">

<div class="view-content">
<div class="views-row views-row-1 views-row-odd views-row-first views-row-last">
```

Step 2. Define extraction rules

Global Search Pattern: [\[?\]](#)

```
Next Event</h2> {%}
```

Item (repeatable) Search Pattern*: [\[?\]](#)

```
"field-content"> <a href="{%}"> {%} </a> {%} </p>
```

Next Event is the string defining the start of the global area, and the `{%}` tells feed43 to search everything after that string. We could have added a delimiter to end the global area also, but did not in this case. Scrolling down in the html a bit, we find the start of the code for a listing:

Below is the HTML source of the retrieved page. Use it to setup extraction rules (see next step).

Page Source: [\[?\]](#)

```
</div>
</div> </a> </div> </div>
<div class="views-field views-field-title"> <span class="field-content"> <a
href="http://www.calacademy.org/nightlife/how-to-nightlife-0">How-To NightLife</a> </span> </div>
<div class="views-field views-field-field-date"> <div class="field-content"> <span class="date-display-
single"> July 16th at 6:00 pm</span> </div> </div>
<div class="views-field views-field-body"> <div class="field-content"> <p>Back by popular demand:
NightLife is serving up a host of how-to workshops, demos, and talks.</p>
</div> </div>
<div class="views-field views-field-field-schedule-item"> <div class="field-content"> <div class="item-
list"> <ul> <li class="first last"> <div class="entity entity-field-collection-item field-collection-item-field-
schedule-item clearfix">
<div class="content">
</div>
</div>
```

Step 2. Define extraction rules

Global Search Pattern: [\[?\]](#)

```
Next Event</h2>{%}
```

Item (repeatable) Search Pattern*: [\[?\]](#)

```
"field-content"> <a href="{%}">{%}</a>{%}</p>
```

We're looking for the text string `"field-content">` is encountered. Look above at the page source and you will see that the snip should be a URL that links to the item in question. Right after that is the title of the event, so we snip until the `` and we also capture the date and description after that in the third snip, stopping with the `</p>` code.

It should be noted that often there is a lot of HTML code between the bits you want that can be skipped. You can do this with the `{*}` operator in feed43. It works just like the `{%}`, only instead of snipping code between the delimiting text, it ignores it. The `{*}` was not used for this particular feed.

What follows is the listing of items feed43 extracted from the page. As of the date I created this, there were three:

Below is list of extracted text snippets (`{%N}`). You can reference them when setting up item properties (see next step).

Clipped Data: [\[?\]](#)



The screenshot shows a window titled "Clipped Data: [?]" with a scrollable list of three items. Each item is preceded by a timestamp in angle brackets. Each item contains three lines of HTML snippets, where the first two lines are marked with `{%1}` and `{%2}` respectively. The third line is marked with `{%3}` and contains a large block of HTML code. To the right of the list, there are two small buttons: a minus sign in a box and a plus sign in a box.

```
Item 1 <Sun, 12 Jul 2015 12:33:29 GMT>
{%1} = http://www.calacademy.org/nightlife/how-to-nightlife-0
{%2} = How-To NightLife
{%3} = </span> </div> <div class="views-field views-field-field-date"> <div class="field-content"> <spa

Item 2 <Sun, 12 Jul 2015 12:33:28 GMT>
{%1} = http://www.calacademy.org/nightlife/dinosaur-nightlife
{%2} = Dinosaur NightLife
{%3} = </span> </div> <div class="views-field views-field-field-date"> <div class="field-content"> <spa

Item 3 <Sun, 12 Jul 2015 12:33:27 GMT>
{%1} = http://www.calacademy.org/nightlife/bigpicture-nightlife
{%2} = BigPicture NightLife
{%3} = </span> </div> <div class="views-field views-field-field-date"> <div class="field-content"> <spa
```

The next portion of the feed43 form allows you to format these snips into something readable that will become your new RSS feed.

Step 3. Define output format

RSS feed properties

Feed Title*: [\[?\]](#)

Academy of Sciences NightLife

Feed Link*: [\[?\]](#)

<http://www.calacademy.org/nightlife/>

Feed Description*: [\[?\]](#)

NightLife: California Academy of Sciences

RSS item properties

Item Title Template*: [\[?\]](#)

{%2}

Item Link Template*: [\[?\]](#)

{%1}

Item Content Template*: [\[?\]](#)

{%3}

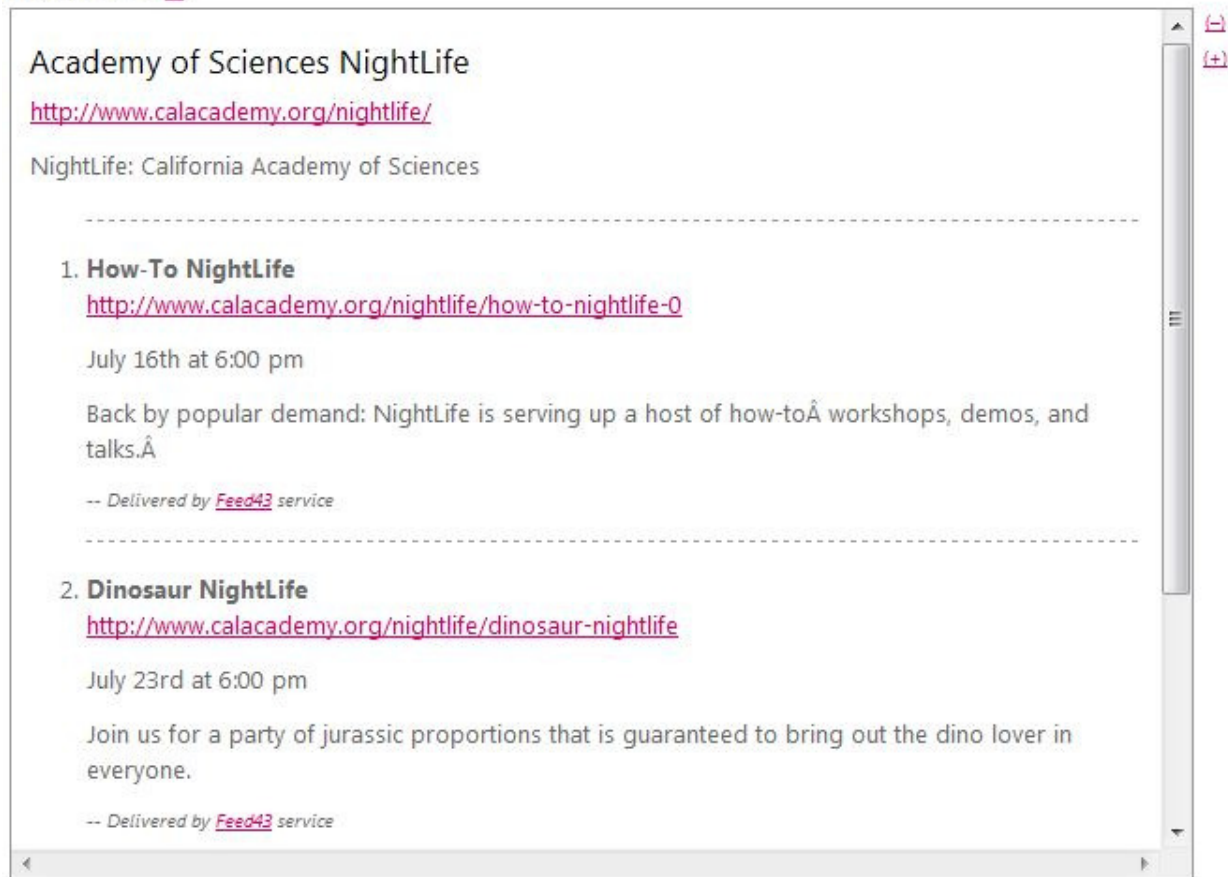
(-)

(+)


The RSS item properties contain the snips. Snip 1 that we extracted was the link to the item, snip 2 was the title, and snip 3 was the additional content. You can include text in addition to the variables, as well as your own HTML formatting info, but in this case just the variables are enough. The resulting feed looks like this:

Here is how your feed will look like in feed reader. Go to next step to get the link to your feed.

Feed Preview: [\[?\]](#)



Step 4. Get your RSS feed

» Feed URL is:  </1343371005228173.xml> [\[?\]](#)

Point your news aggregator to this URL or test this feed in browser.

Note at the bottom is the link to your new URL. If you click on it (in feed43, not here) it will open your feed. You can cut and paste the URL from that feed into whatever RSS reader you want to use. Now, anytime a change is made to any of the text in this section of the web page, or a new event is added, feed43 will generate a new item to the feed, and it will show up in your RSS reader as a new item.

This capability is very powerful, but still has limitations, mostly based on the deficiencies in various web page designs.

CONCLUDING THOUGHTS ON FEED43.COM

Using such a service eliminates the need to either monitor each page of each organization by hand, or the need to use a less restrictive page monitor, one that only says the page changed, but not what changed on it, or one that gives a lot of false alarms. {For an example of a false alarm, one need only look at a page that has a counter of clicks on it. Every time someone goes to the page, the counter goes up by one. Each such “change” might generate an entry in an RSS feed for a generic page change monitor. Being able to limit the section of the page, and extract the specific items you are looking for is a huge time saver.)

To use this RSS feed, click on the link to each listing, which will take you to the specific page for that listing. Copy the parts you want into your calendar software and create a calendar listing. Bay Area Science Calendar software includes the ability to link entries back to the original page, so we always include that URL when we can. **We urge our to click through to the originating website for updates and more details than we can list on our calendar.**