In [1]:
```
from IPython.display import HTML, Image
```

Learning Javascript while working with Brython

What it is Brython?

Brython derives from Browser Python

Brython wants to be a 100% compliant Python3 implementation made with Python and Javascript that runs in the browser

Some time ago the question was: Why should I use javascript?

Two answers:

- **The first is that you have no choice**
- **The second is that javascript could be really good (*)**

Douglas Crockford (http://www.crockford.com/index.html), 'Javascript, the good parts'

(*) or at least it wouldn't be so bad as many people believe_

Really? ;-)

In [2]:
```
HTML('<iframe src=https://www.destroyallsoftware.com/talks/wat width=900 height=500></
```

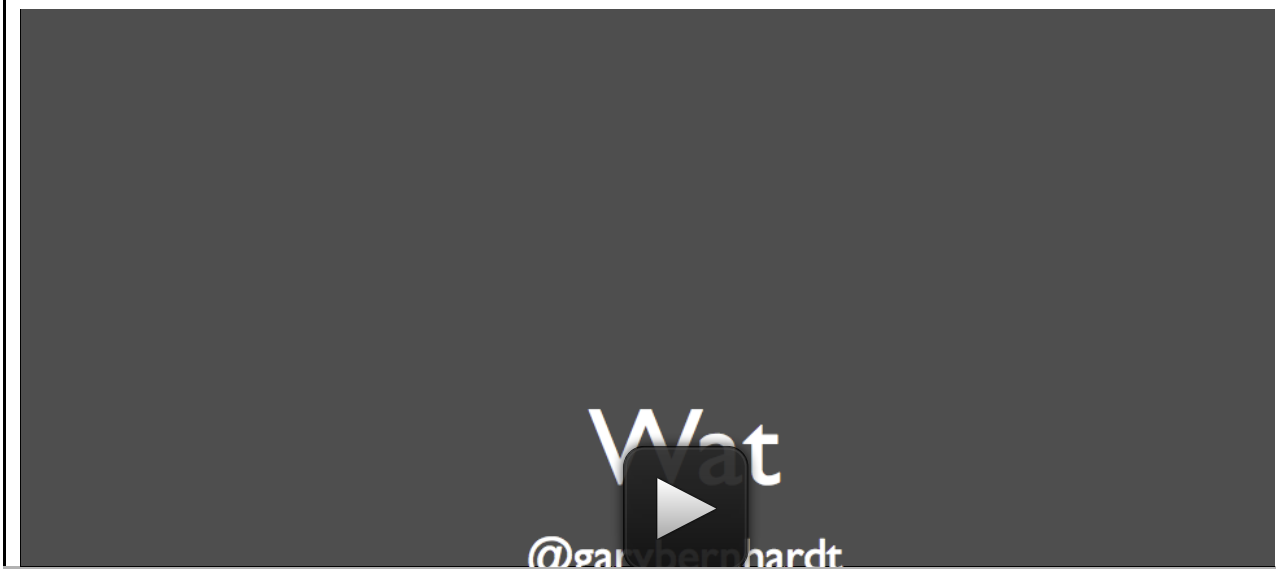
Out[2]:

Destroy All Software Talks

Screencasts — Blog — Talks

Wat

A lightning talk by Gary Bernhardt from CodeMash 2012



But it was some time ago. Nowadays people could

choose other options...

In [3]:
```
HTML('<iframe src=http://altjs.org/ width=900 height=350></iframe>')
```

Out[3]:

In [4]:
```
from IPython.display import HTML
HTML('<iframe src=http://stromberg.dnsalias.org/~strombrg/pybrowser/python-browser.htm
me>')
```

Out[4]:

Project name	Hide column Gist	Hide column Actively developed?	Hide column Can import?	Hide column Compatibility with CPython 2 or 3	Hide column Python stdlib?	Hide co Can c Javas code?
	Python 3 translator			Python 3.x. Some esoteric	Some: datetime, hashlib, json, math,	

Credits: Stromberg (http://stromberg.dnsalias.org/~strombrg/pybrowser/python-browser.html)

Other similar lists (https://github.com/jashkenas/coffee-script/wiki/List-of-languages-that-compile-to-JS).

WTF?? Why are you trying to do that?

Even GvR doesn't recommend to do that!!

Python in the browser ?

by Btrot69

Over the years, there have been several attempts to create a sandboxed version of python that will safely run in a web browser.Mostly this was because of problems with Javascript. Now that Javascript works -- and we have nice things like CoffeeScript -- is it time to give up on python in the browser ?

Guido: I gave up on it in 1995, so yes. And please don't try to compile Python to JavaScript. The semantics are so different that you end up writing most of a Python runtime in JavaScript, which slows things down too much. (CoffeScript's strength is that it is designed to map cleanly to JavaScript, and the two are now co-evolving to make the mapping even cleaner.)

See complete interview here (http://developers.slashdot.org/story/13/08/25/2115204/interviews-guido-van-rossum-answers-your-questions).

Brython, the good parts

Brython is a (subset of) CPython3 that runs in the browser

Python syntax

```
a = 1
b = 2
print(a + b)

a, b = 1, 2
print(a + b)

print(type(a))
print(type(a * 1.0))

print(1 + 'a')
```

In [5]:
```
HTML('<iframe src=http://brython.info/tests/console.html width=900 height=350></iframe
```

Out[5]:

Functions

```
def sum(a, b):
    return a + b
print(sum(2,2))
```

In [6]:
```
HTML('<iframe src=http://brython.info/tests/console.html width=900 height=350></iframe
```

Out[6]:

Classes

```
class Square:
    def __init__(self, length):
        self.len = length

    def area(self):
        return self.len ** 2

sq = Square(2)
print(sq.area())
```

In [7]:
```
HTML('<iframe src=http://brython.info/tests/console.html width=900 height=350></iframe
```

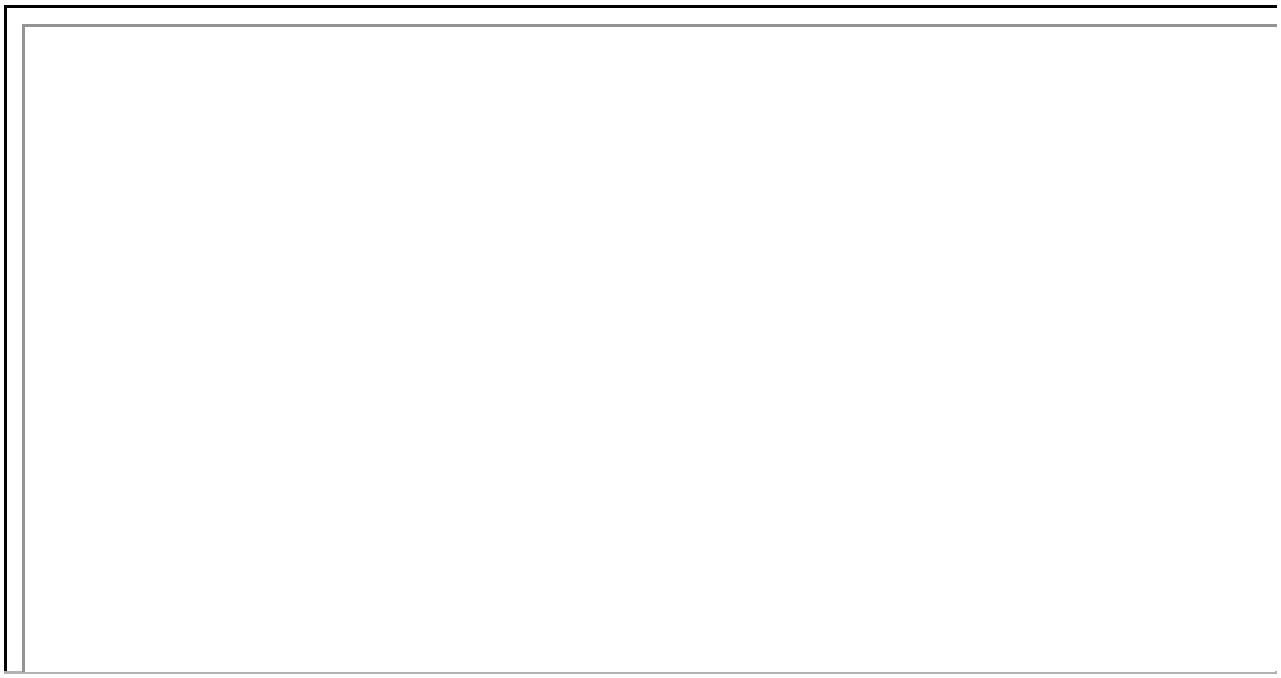
Out[7]:

Inheritance

```
class Square:
    def __init__(self, length):
        self.len = length

    def area(self):
        return self.len ** 2

sq = Square(2)
print(sq.area())

class Square2(Square):
    def perimeter(self):
        return self.len * 4

sq = Square2(10)
print(sq.area())
print(sq.perimeter())
```

In [8]:
```
HTML('<iframe src=http://brython.info/tests/console.html width=900 height=350></iframe
```

Out[8]:



How classes are implemented in Brython

Python classes (including built-in classes) are implemented with 2 different Javascript objects : a factory function that creates instances (it uses `__new__` and `__init__` when available) and a dictionary that holds the class attributes and methods.

(*) `super` is not implemented in Brython 1.2 (well, it is since some days ago (https://bitbucket.org/olemis/brython/src/a062b0a69cd064bfd4131fc9323ee47416e5545c/src/py_builtin_functions.js?at=default#cl-1147)).

Decorators

```
def world(some_func):
    def pre(arg):
        greet = some_func(arg)
        return greet + ' world!'
    return pre

@world
def whatever(word):
    return word

print(whatever('hello'))
```

In [9]:
```
HTML('<iframe src=http://brython.info/tests/console.html width=900 height=350></iframe
```

Out[9]:

Brython supports most keywords and functions of Python 3 :

Keywords

```
as, assert, break, class, continue, def, del, elif, else, except, False, final
ly, for, from, global, if, import, is, lambda, None, pass, return, True, try,
while, with, yield
```

Built-in functions

```
abs(), all(), any(), ascii(), bin(), bool(), bytes(), callable(), chr(), class
method(), delattr(), dict(), dir(), divmod(), enumerate(), eval(), exec(), fil
ter(), float(), frozenset(), getattr(), globals(), hasattr(), hash(), hex(), i
d(), input(), int(), isinstance(), iter(), len(), list(), locals(), map(), max
(), min(), next(), object(), open(), ord(), pow(), print(), property(), range(
), repr(), reversed(), round(), set(), setattr(), slice(), sorted(), str(), su
m(), tuple(), type(), zip()
```

The following are not implemented in the current version :

keywords

```
nonlocal
```

built-in functions

```
bytearray(), compile(), complex(), format(), help(), memoryview(), super(), va
rs(), __import__
```

The **complex number type** (j) is not supported

Ok, I have Python to simulate Python :-)

Show me the money!!!

Ok, let's see what Brython can do in the browser

First of all, to use Brython you need to:

- **include the following in your html file**

```
<script type="text/javascript" src="path/to/the/library/brython.js">
```

- **Include the following in the body tag**

```
<body onload="brython()">
```

- **Include your python code in a `script` tag using `text/python` or `text/python3`**

```
<script type="text/python">

...Your Python code...

</script>
```

There are some things that are different to Python

By default, `print()` will output to the web browser console and so are the error messages. `sys.stderr` and `sys.stdout` can be assigned to an object with a `write()` method, and this allows for the redirection of output to go to a window or text area, for example.

`sys.stdin` is not implemented at this time, however there is an `input()` built-in function that will open a blocking input dialog (a prompt).

To open a print dialog (to a printer), call `win.print()`.

Some keywords and built-in functions designed for operation in a browser have been added:

built-ins

```
alert(), confirm(), prompt()
```

correspond to their Javascript equivalents

the `win` keyword is the window (window object in JS) and `doc` represents the HTML document (document in JS).

In [10]:
```
HTML('<iframe src=http://curious-electric.com/brython-playground/ width=900 height=400
```

Out[10]:

HTML

```
<p>Your name? <input id="myname">
<button onclick="echo()">click!</button>
```

CSS

```
body { color: blue; }
```

Python

```
def echo():
    alert(("Hello %s !" % doc["myname"].value))
```

Your name?

Your name?

Your name?

Credits of the Brython jsFiddle clone (https://github.com/dirkk0/brython-playground): Dirk Krause.

How Can I access the HTML elements

Getting access to an element can be done in different ways. The most usual is to use its identifier, ie its attribute id : with an input field defined by

```
<input id="data">
```

we can get a reference to this field by

```
data = doc["data"]
```

doc is a built-in Brython keyword that refers to the HTML document. It behaves like a dictionary whose keys are the identifiers of the elements in the page. If not element has the specified id, the program raises a KeyError exception

We can also get all the elements of a given type, for instance all the hypertext links (HTML tag A), using the syntax

```
import html
links = doc[html.A]
```

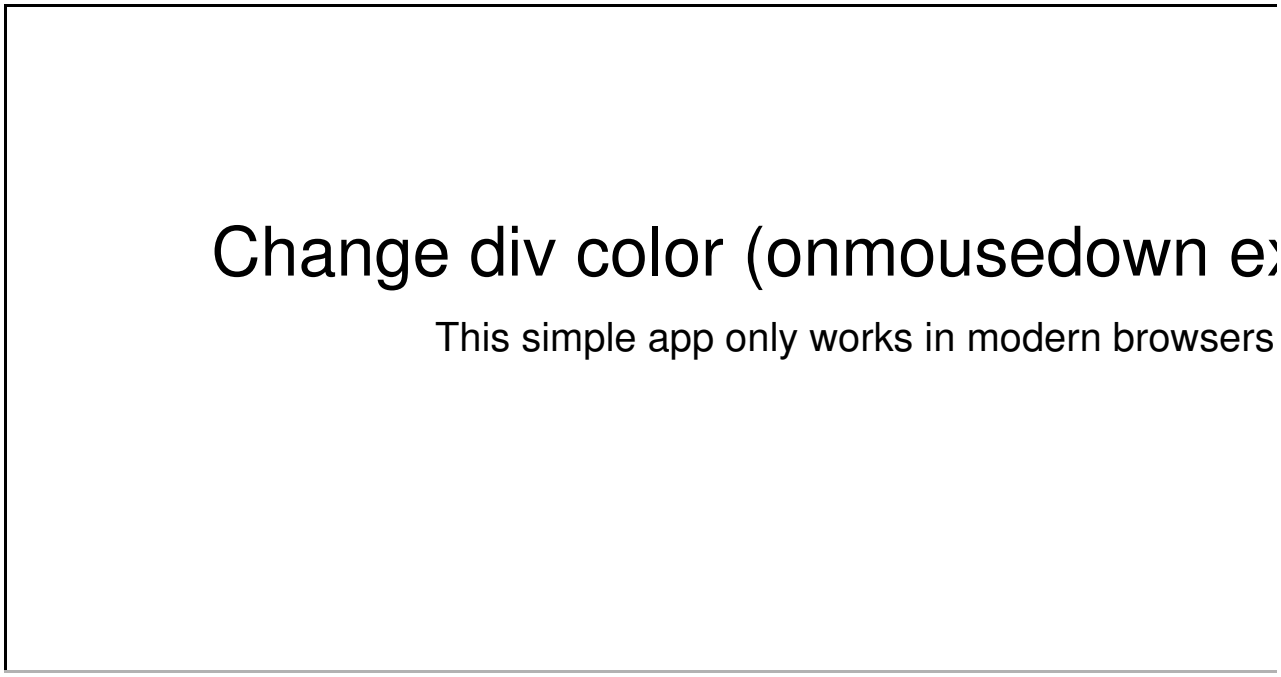
Finally, all the elements in the page have a method get() that can be used to search elements :

```
elt.get(name=N) returns a list of all the elements descending from elt whose a
ttribute name is equal to N
elt.get(selector=S) returns a list with all the elements descending from elt w
hose CSS selector matches S
```

Ok, let's see Brython in Action with some examples I created for the PyConES'2013!!

In [6]:
```
HTML('<iframe src=https://googledrive.com/host/0B4OEtv-kAaTBSllJM19hdkpCeTQ/colors_boo
ht=350></iframe>')
```

Out[6]:



Link to colors app (https://googledrive.com/host/0B4OEtv-kAaTBSllJM19hdkpCeTQ/colors_bootstrapped.html).

(Warning: to make this work you should run a server first so it is not working in the presentation)

In [9]:
```
HTML('<iframe src=https://googledrive.com/host/0B4OEtv-kAaTBSllJM19hdkpCeTQ/table_boot
t=350></iframe>')
```

Out[9]:

Link to table app (https://googledrive.com/host/0B4OEtv-kAaTBSllJM19hdkpCeTQ

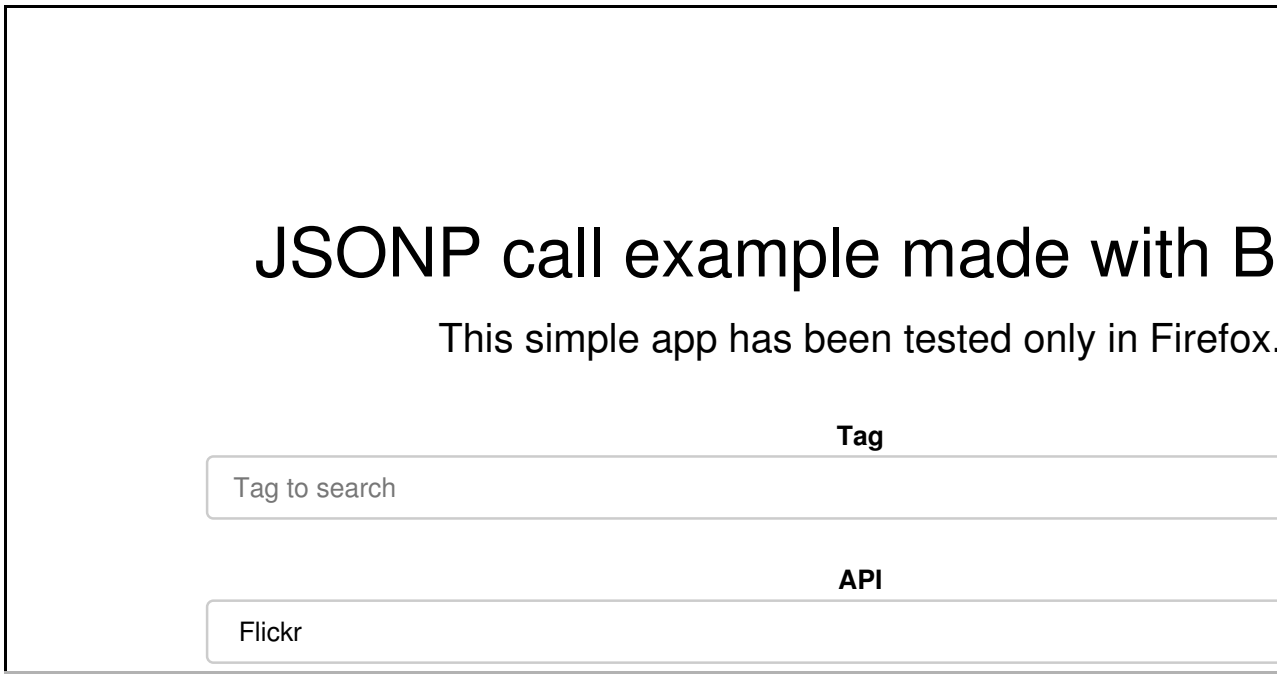
In [11]:
```
HTML('<iframe src=https://googledrive.com/host/0B4OEtv-kAaTBSllJM19hdkpCeTQ/puzzle_boo
ht=350></iframe>')
```

Out[11]:

Link to puzzle game app (https://googledrive.com/host/0B4OEtv-kAaTBSllJM19hdkpCeTQ/puzzle_bootstrapped.html).

In [12]:
```
HTML('<iframe src=https://googledrive.com/host/0B4OEtv-kAaTBSllJM19hdkpCeTQ/jsonp_boot
t=350></iframe>')
```

Out[12]:



Link to jsonp request app (https://googledrive.com/host/0B4OEtv-kAaTBSllJM19hdkpCeTQ/jsonp_bootstrapped.html).

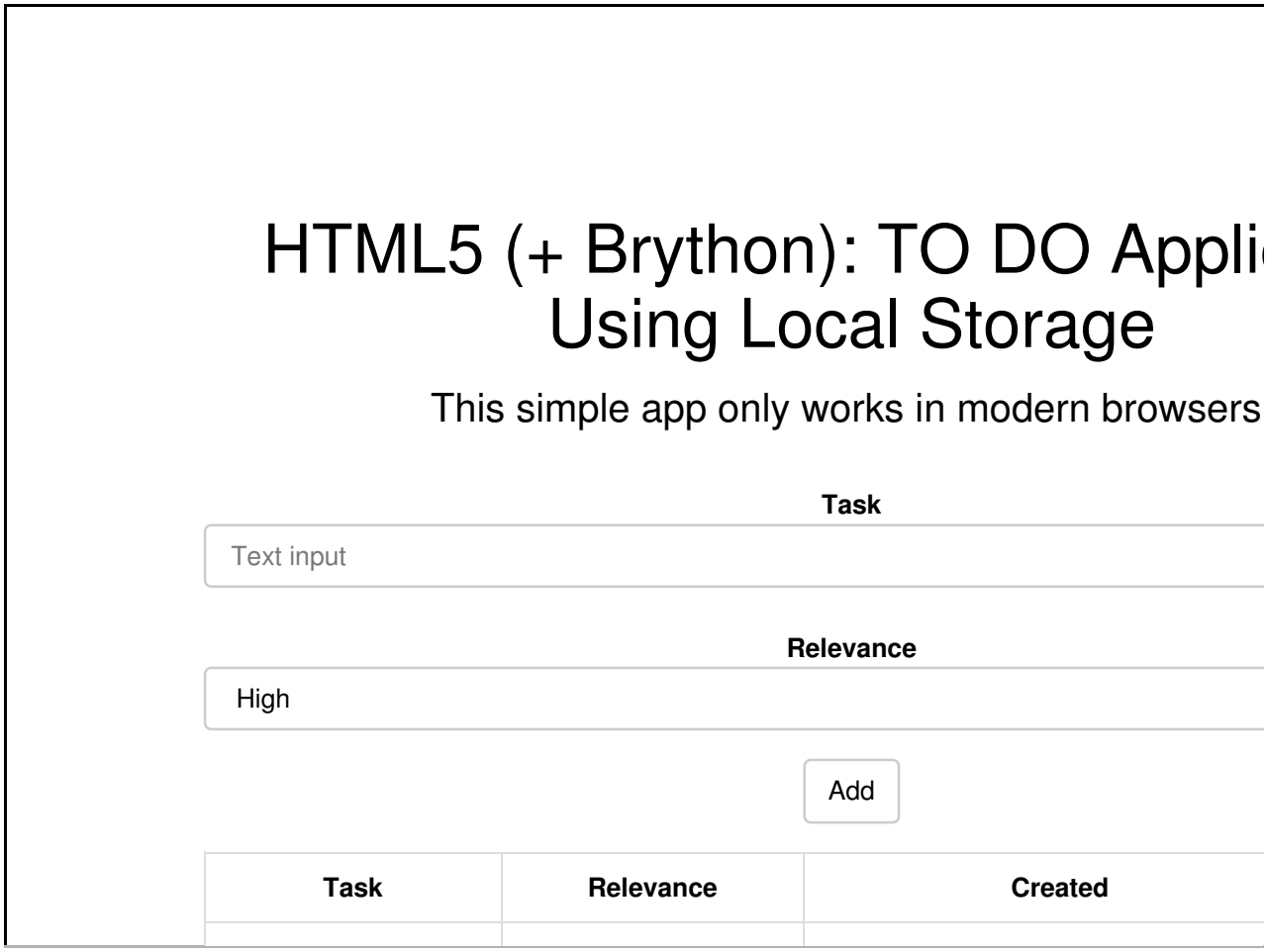
In [13]:
```
HTML('<iframe src=https://googledrive.com/host/0B4OEtv-kAaTBSllJM19hdkpCeTQ/hangman_bo
ght=500></iframe>')
```

Out[13]:

Link to hangman game app (https://googledrive.com/host/0B4OEtv-kAaTBSllJM19hdkpCeTQ/hangman_bootstrapped.html).

In [14]:
```
HTML('<iframe src=https://googledrive.com/host/0B4OEtv-kAaTBSlIJM19hdkpCeTQ/todo_boots
=500></iframe>')
```

Out[14]:



Link to TO-DO list app (https://googledrive.com/host/0B4OEtv-kAaTBSlIJM19hdkpCeTQ/todo_bootstrapped.html).

https://github.com/kikocorreoso/Brython-PyConES-2013 (https://github.com/kikocorreoso/Brython-PyConES-2013)

@pybonacci (https://twitter.com/Pybonacci)

Official repository https://bitbucket.org/olemis/brython/overview (https://bitbucket.org/olemis/brython/overview)

Official web page (http://brython.info/)

Apps running in my Gdrive (https://drive.google.com

/folderview?id=0B4OEtv-kAaTBUlE2OU9QcHVpT3c&usp=sharing).

And that's all!

Now you should love Brython :-)

In [3]:
```
Image("http://images.fanpop.com/images/image_uploads/sponge-bob-rocks-spongebob-squarepants-154588_427_600.jpg")
```

Out[3]:

Back to top