

# I want to connect weird stuff to the internet.

Brandon Stafford  
Only Person at Rascal Micro

Sketching in Hardware

July 31, 2011

Creative Commons Attribution-Sharealike License, except for the logos

rascal micro

I can't find the damn thing's  
IP address

# Zero configuration networking

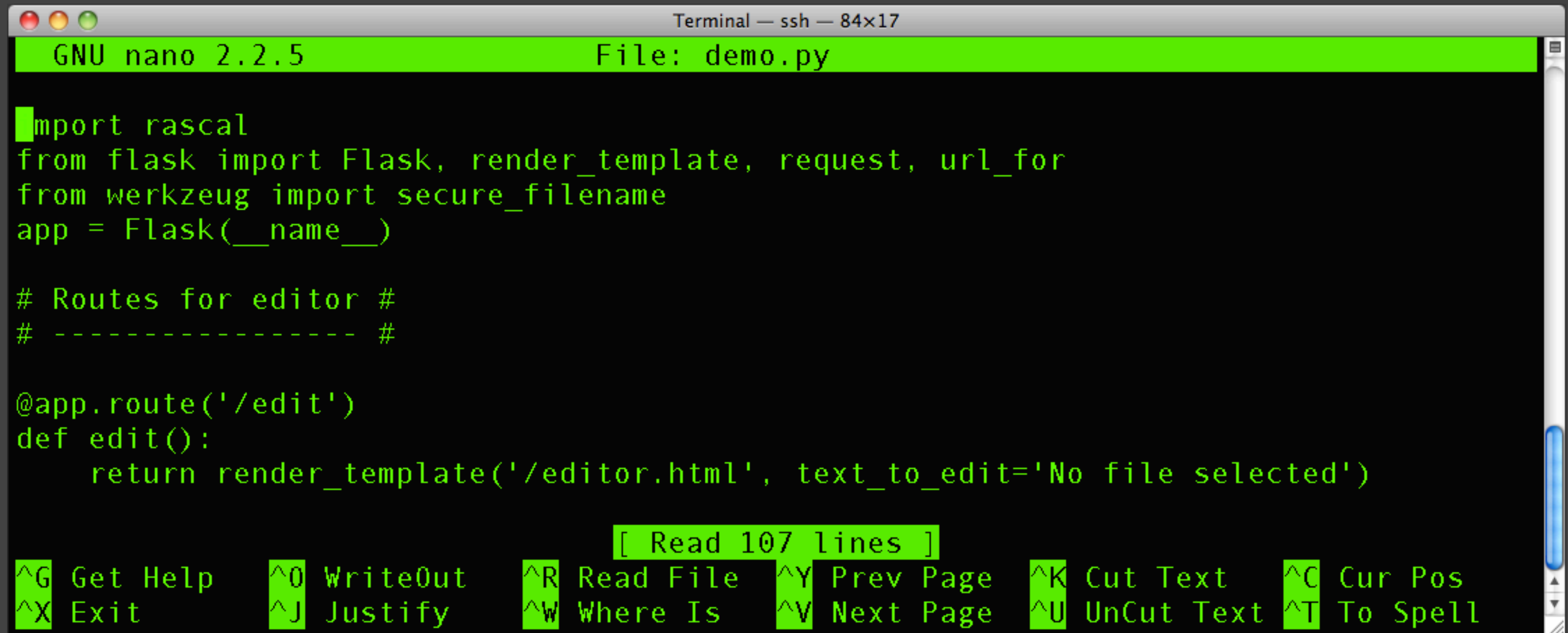


AKA “Bonjour”

# SSH is baffling.

```
Terminal — zsh — 84x17
Last login: Mon Jul 11 16:23:44 on ttys001
[brandon@fosco:~]$ ssh root@192.168.1.4 (07-11 16:24)
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!    @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the RSA host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
84:39:4f:ac:b3:9b:41:cf:71:2d:a6:09:88:32:00:c8.
Please contact your system administrator.
Add correct host key in /Users/brandon/.ssh/known_hosts to get rid of this message.
Offending key in /Users/brandon/.ssh/known_hosts:2
RSA host key for 192.168.1.4 has changed and you have requested strict checking.
Host key verification failed.
[brandon@fosco:~]$ (07-11 16:26)
```

# Console text editors are ugly.



```
Terminal - ssh - 84x17
GNU nano 2.2.5 File: demo.py

import rascal
from flask import Flask, render_template, request, url_for
from werkzeug import secure_filename
app = Flask(__name__)

# Routes for editor #
# ----- #

@app.route('/edit')
def edit():
    return render_template('/editor.html', text_to_edit='No file selected')

[ Read 107 lines ]

^G Get Help    ^O WriteOut   ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit        ^J Justify    ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

# Self-serve programming tools

The screenshot displays the Rascal Micro web interface in a browser window. The browser's address bar shows the URL `demo.rascalmicro.com:81/edit`. The interface features a dark theme with a top navigation bar containing the "rascal micro" logo and links for "log", "config", and "monitor". A red "Save" button is positioned on the left, and a "Reload pytronics" button is on the right. A central console area contains the text "In theory, there will be console messages here." The main workspace is a code editor displaying HTML code for a "Rascal demo" page. The code includes a meta charset, title, and links to a stylesheet and favicon. The body contains an h1 "Rascal demo" and a link to "index.html" labeled "Sensor demo". A file explorer on the left lists files such as `jquery.jqplot.css`, `demo.py`, `video-bourne.html`, `demo.html`, `some-js-files`, `overlay.js`, `runmode.js`, and `video.html`. The browser's status bar at the bottom shows `demo.rascalmicro.com:81/edit#`.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Rascal demo</title>
  <link rel="stylesheet" type="text/css" href="/static/style.css">
  <link rel="shortcut icon" href="/static/favicon.ico">
</head>
<body>
  <div class="rascalcontent">
    <h1>Rascal demo</h1>
    <p><a href="/index.html">Sensor demo</a></p>
  </div>
</body>
</html>
```

# Embedded C, Makefiles, and gcc are scary.

```
Terminal — vim — 128x16
// SPI Port Init
USICTL0 |= USIPE6;           // SDO port enable
USICTL0 |= USIPE5;           // Enable SPI clock
USICTL0 &= ~USILSB;         // MSB first
USICTL0 |= USIMST;          // Master mode
USICTL0 |= USIGE;           // Set to "output latch always enabled and transparent"
USICTL0 |= USIOE;           // Output enabled
USICTL1 &= ~USII2C;         // I2C disabled
USICTL1 |= USICKPH;         // Clock phase select
USICTL1 &= ~USIIE;         // Disable interrupt
USICKCTL = USIDIV_1 + USISSEL_2; // Clk divided by 16 (clk = 71.5 kHz); use SMCLK
USICKCTL |= USICKPL;        // Inactive state high
USICNT = USI16B;            // 16 bit shift register mode, no bits ready for transfer
```

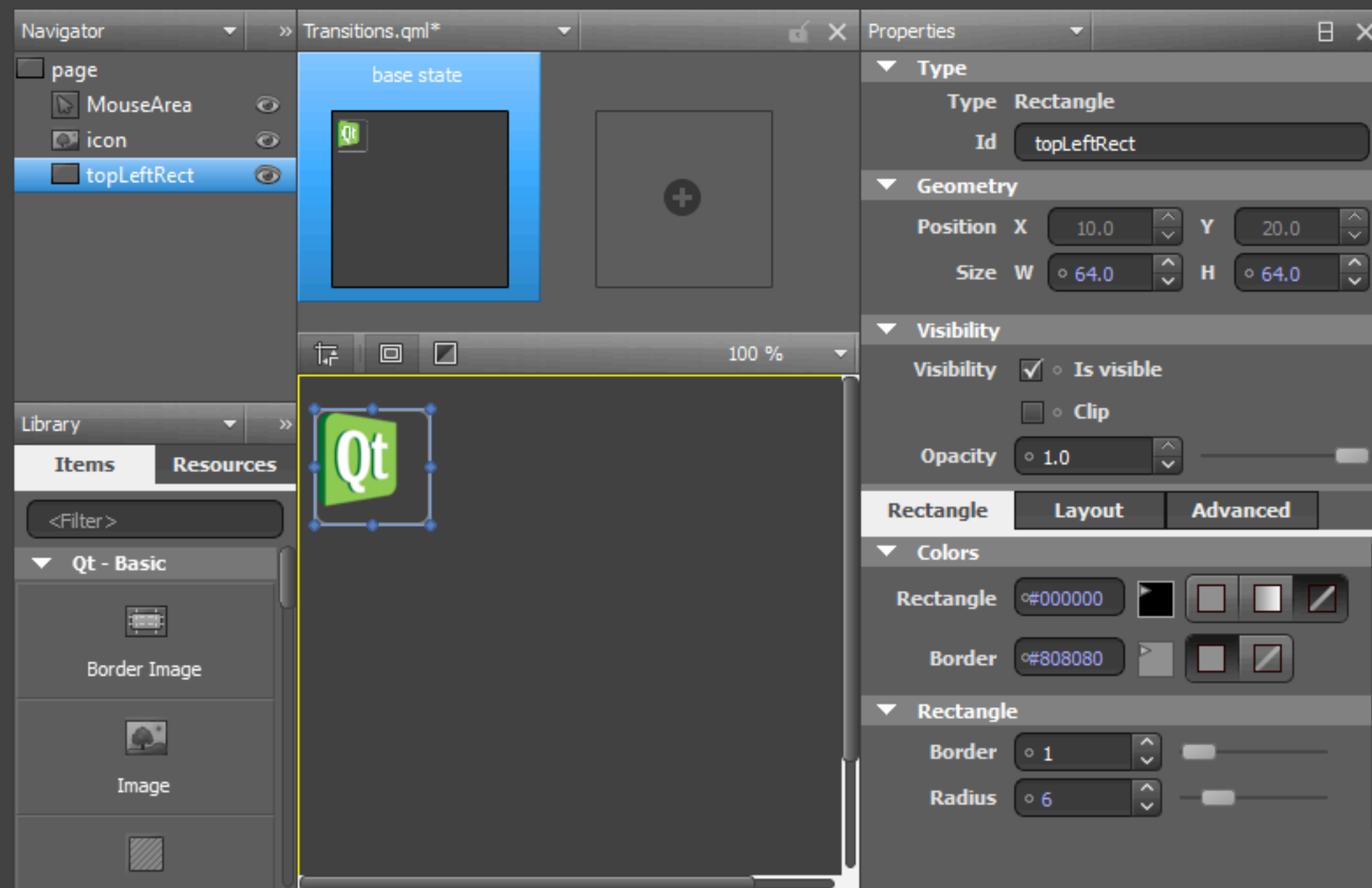
# Python is friendly.

So is Ruby.

```
def write_serial():  
    import rascal  
    return rascal.send_serial(request.form['serial_text'])
```



# GUI builders for Qt or GTK



Qt Quick IDE

# HTML and CSS have won

“[HTML] is *the* format of our age.”  
-- Mark Pilgrim

# No users, permissions, sessions

# Nailed by web frameworks



Ruby on Rails

Django (sort of “Python on Rails”)

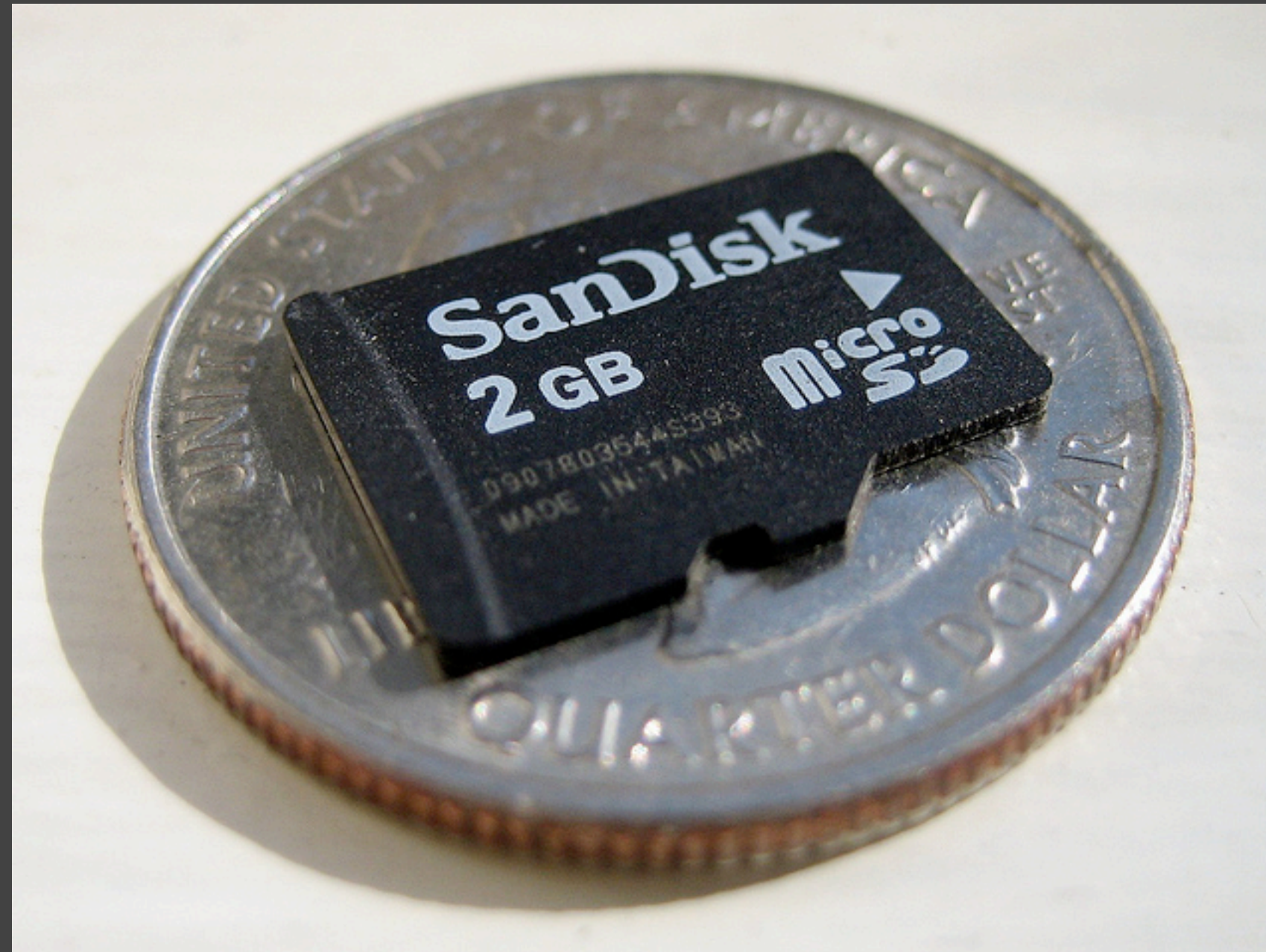
My personal favorite: Flask



# Low fault tolerance

“I edited some files, and now it won't boot.”

# Preconfigured SD cards



# Dynamic IP addresses

# Routers with free firmware

**Tomato**  
Version 1.00.0905

Status	WAN / Internet
Overview	Type <input type="text" value="DHCP"/>
Device List	MTU <input type="text" value="Default"/> <input type="text" value="1500"/>
Logs	
Bandwidth	LAN
Real-Time	Router IP Address <input type="text" value="192.168.3.1"/>
Last 5 Hours	Subnet Mask <input type="text" value="255.255.255.0"/>
Daily	Static DNS <input type="text" value="0.0.0.0"/>
Monthly	<input type="text" value="0.0.0.0"/>
Tools	<input type="text" value="0.0.0.0"/>
Ping	DHCP Server <input checked="" type="checkbox"/>
Trace	Start <input type="text" value="192.168.3.100"/>
Wireless Survey	End <input type="text" value="192.168.3.149"/>
WOL	Lease Time <input type="text" value="120"/> (minutes)
	WINS <input type="text" value="0.0.0.0"/>
Basic	Wireless
Network	Enable Wireless <input checked="" type="checkbox"/>
Identification	MAC Address <input type="text" value="00:14:"/>
Time	Wireless Mode <input type="text" value="Access Point + WDS"/>
DDNS	
Static DHCP	
Wireless Filter	
Advanced	
Port Forwarding	
QoS	
Access Restriction	
Administration	

Tomato or DD-WRT with dynamic DNS