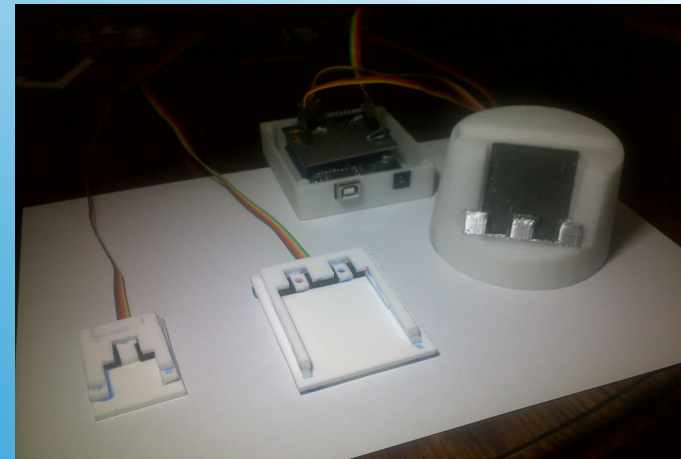


**CherryPy + CodeMirror =  
Online IDE**

**Sydney SyPy September 2012**

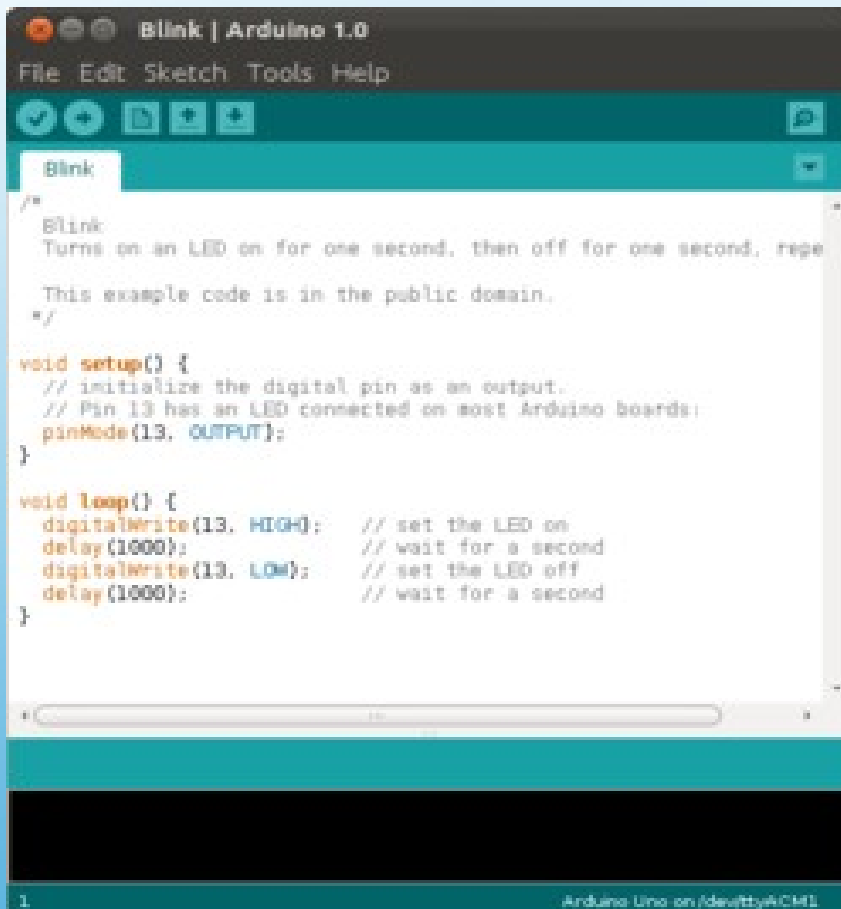
HackerPads is an Open-Source computing environment that runs across embedded systems.

It's a system that brings together the best of Arduino, Raspberry-Pi, Embed and provides them all with common Cases, Interfaces and Peripherals and perhaps programming.



**HackerPads – Online IDE**

## What's wrong with Arduino ?

A screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 1.0". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for opening, saving, and running. The main text area contains the following code:

```
/*  
 * Blink  
 * Turns on an LED on for one second, then off for one second, repeatedly.  
 * This example code is in the public domain.  
 */  
  
void setup() {  
  // initialize the digital pin as an output.  
  // Pin 13 has an LED connected on most Arduino boards:  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH); // set the LED on  
  delay(1000);           // wait for a second  
  digitalWrite(13, LOW); // set the LED off  
  delay(1000);           // wait for a second  
}
```

The status bar at the bottom indicates "1" and "Arduino Uno on /dev/ttyACM1".

- Not much, but it's Java based. I want Python.
- Everything these days is web based, or even Cloud based. I want that.
- Edit.. compile and then upload is so old school.

# What's wrong with Arduino ?

Possible reasons:

- Compile, upload, run is too annoying.
- Too many devices, too far to walk (across the room to get one for reprogramming). Just reprogram it using wifi.
- More and more embedded systems are coming with Network interfaces these days.
- Embedded Editors are too powerful to ignore. If they are there, then why not use one.

**Why do devices need a Web IDE?**

Ace (editor)
CodeMirror
CodePress
CodeTextArea
EditArea
Helene
Orion
MDK-Editor

## Web Editors

There's a number of different Web Editing widgets out there. They are pretty amazing. After reviewing the common ones, I chose CodeMirror in the end because of its small footprint which would make it good for embedded devices.

# Web Editors

Which Web Server is best on an embedded system?

```
import cherrypy

class HelloWorld(object):
    def index(self):
        return "Hello World!"
    index.exposed = True

cherrypy.quickstart(HelloWorld())
```

- After lots of testing of Snorkel and other embedded webservers I gave up and went back to Python
- CherryPy is what I wanted: small and customisable.
- Should work on a Raspberry-Pi

**Web Servers ? Embedded or  
CherryPy ?**



# CodeMirror: Theme demo

```
1 function findSequence(goal) {
2   function find(start, history) {
3     if (start == goal)
4       return history;
5     else if (start > goal)
6       return null;
7     else
8       return find(start + 5, "(" + history + " + 5)") ||
9             find(start * 3, "(" + history + " * 3)");
10  }
11  return find(1, "1");
12 }
```

Select a theme:

- CodeMirror is a widget with comprehensive editing capabilities
- You simply load it as a widget on a html page for instant editing capabilities.
- Numerous features and capabilities.

## Basic CodeMirror

## CodeMirror: Python mode

```
64 # Python 2 Types (otherwise Identifiers)
65 basestring buffer file long unicode xrange
66
67 # Python 3 Types (otherwise Identifiers)
68 bytearray bytes filter map memoryview open range zip
69
70 # Some Example code
71 import os
72 from package import ParentClass
73
74 @nonsenseDecorator
75 def doesNothing():
76     pass
77
78 class ExampleClass(ParentClass):
79     @staticmethod
80     def example(inputStr):
81         a = list(inputStr)
82         a.reverse()
83         return ''.join(a)
84
85     def __init__(self, mixin = 'Hello'):
86         self.mixin = mixin
87
88
```

- Built in themes
- Built in Language support.
- Nice ways to do tabs and auto-complete
- Python mode

# Code Mirror Features



- I just used `easy_install` to install CherryPy

```
Sudo easy_install cherrypy
```

- To run the server:

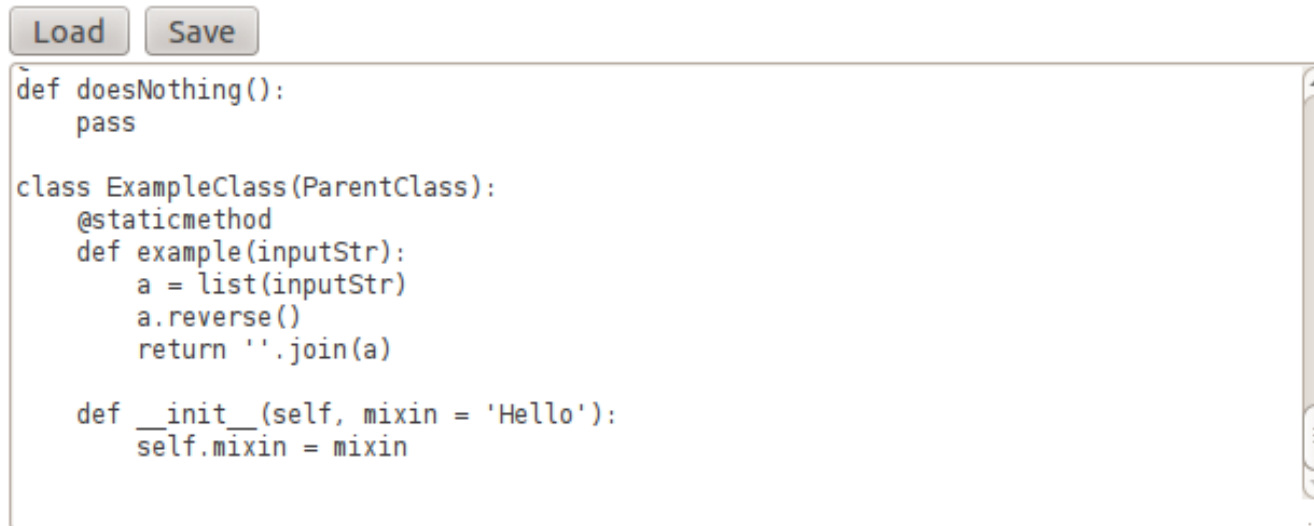
```
Python ide.py
```

```
dlyon@dlyon-dc7800p:~/hackerpads/WebIDE$ python ide.py
[06/Sep/2012:15:57:51] ENGINE Bus STARTING
[06/Sep/2012:15:57:51] ENGINE Started monitor thread 'Autoreloader'.
[06/Sep/2012:15:57:51] ENGINE Started monitor thread '_TimeoutMonitor'.
[06/Sep/2012:15:57:51] ENGINE Serving on 127.0.0.1:8080
[06/Sep/2012:15:57:51] ENGINE Bus STARTED
```

# Installing and running CherryPy

# Embedded Web Development System

For Raspberry-Pi, ARM, Teensy, Chumby, Arduino



```
def doesNothing():
    pass

class ExampleClass(ParentClass):
    @staticmethod
    def example(inputStr):
        a = list(inputStr)
        a.reverse()
        return ''.join(a)

    def __init__(self, mixin = 'Hello'):
        self.mixin = mixin
```

## HackerPad Development IDE:

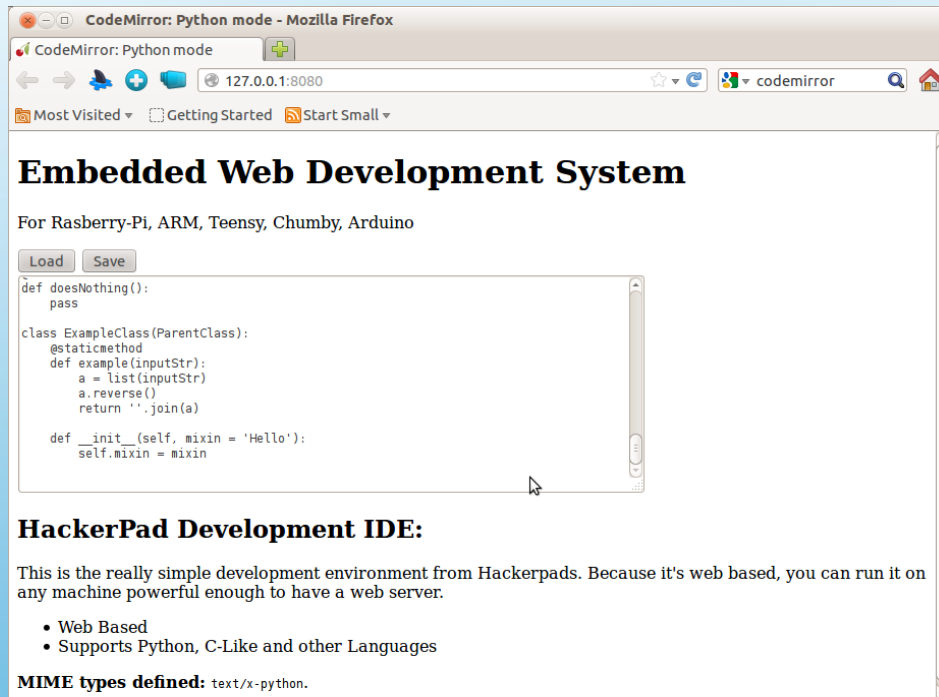
This is the really simple development environment from Hackerpads. Because it's web any machine powerful enough to have a web server.

# Add a programming Window..

```
index.html x
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>CodeMirror: Python mode</title>
6 <link rel="stylesheet" href="/css/codemirror.css">
7 <script src="/lib/jquery.js"></script>
8 <script src="/lib/codemirror.js"></script>
9 <script src="/lib/python.js"></script>
10 <link rel="stylesheet" href="/css/styles.css">
11 <style type="text/css">.CodeMirror {border-top: 1px solid black;
12 </head>
13 <body>
14 <h1>Embedded Web Development System</h1>
15 <p>For Raspberry-Pi, ARM, Teensy, Chumby, Arduino</p>
16 <button id="load">Load</button>
17 <button id="save">Save</button>
18 <div><textarea id="code" name="code">
19 # Literals
20 1234
21 0.0e101
```

- I added some basic buttons for Loading and Saving
- There's more coding to go in.
- Lot's of jquery to add.

# Add Compile and run buttons



- That's it for now
- Developers are welcome to join my project.
- [www.bitbucket.org/djlyon/hackerpads](http://www.bitbucket.org/djlyon/hackerpads)

Server is ready to deploy