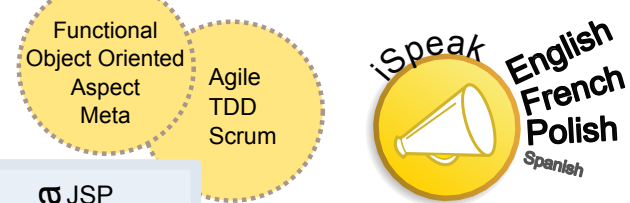


# Tomasz Bogdal

Citizenship  
CA  
PL



Interests  
Electronics  
3D Modeling  
Reverse Engineering  
Rock Climbing

**BUILD** CMake, make, ant, autotools  
**C++** QT, WebKit, libusb, jsoncpp, Py++  
**WEB** HTML, XHTML, XML, CSS, jQuery, Dojo  
**VCS** SVN, GIT, HG, BZR  
**Python** PyUnit, PyPy, Nose, Django, Sphinx, Suds, NymPy, SciPy, libVirt  
**Java** JSP, GWT, Servlets, Maven, Guice  
**Virt** Xen, XenServer, KVM, VirtualBox

Technologies

Windows 95 - Win 7  
Linux Gentoo, Arch, Ubuntu, SLED, Centos  
Mac OS X 10.2 - 10.6  
Solaris

SO



SO  
COMMUNICATOR  
DETAIL ORIENTED  
CREATIVE  
INDEPENDENT  
MOTIVATED

2012-Present **Arduino-CMake - Open Source Project** Creator/Maintainer  
<https://github.com/queezythegreat/arduino-cmake/>  
2009-Present **[NComputing] Linux Software Developer** Build Tools Specialist, Automation, Embedded Firmware (ARM), Imprivata, Virtualization/Cloud, HDX N-Series  
2009-2010 **[CocoLab] Product Release and Management Consulting**

2009 **[Cyfronet AGH] Software Developer [Internship]** Monitoring and Metering software for PLGrid infrastructure (Java/GWT)  
2009 **Developed teaching aids for Numerical Methods and Simulation class**  
WSZiB University - PyMOIS  
<https://pypi.python.org/pypi/PyMois/0.1>

2006 - 2009 **WSZiB University [Polish]** Wyższa Szkoła Zarządzania i Bankowości w Krakowie - Wydział Informatyki

2007 - 2008 **Computer Science tutoring and freelance Java programming**

2004 - 2006 **Computer Science Tutoring** Programming, networking

## EXPERIENCE

2002-2006 **High school [Polish]** Liceum Ogólnokształcące Ojców Pijarów Krakow, Poland

1999-2001 **Intermediate High school [English]** Shorewood, Milwaukee, USA

1991-1999 **Elementary School [French]** Montreal, Quebec, Canada

## EDUCATION

1986 **Initail commit:** Born, Krakow, Poland



```
# head -n 16 languages_known.py
languages = [
    ('Python', features('dynamic', 'prototype', 'os-independent')),
    ('Java', features('os-independent', 'web', 'enterprise')),
    ('C', features('system', 'embedded', 'performance')),
    ('C++', features('performance', 'object-oriented', 'system')),
    ('Pascal', features('learning', 'readability')),
    ('Bash', features('system', 'commands', 'shell')),
    ('SH', features('system', 'commands', 'shell', 'legacy')),
    ('JavaScript', features('web', 'browser', 'client')),
    ('PHP', features('web', 'server', 'web-app')),
    ('Assembler', features('bare-metal', 'ia-32', 'performance')),
]
for language, features in languages:
    if problem.requirements in features:
        print 'I can solve the problem with:', language
    else:
        print 'No language available, I need to learn something new.'
```

Languages