

# BlitzType

A Predictive Text Completion Software  
in Python

Wong Jiang Fung

*I will wait for you...*

# Contents

- What's Dyslexia?
- What's BlitzType? - demo
- 3 most exciting experiences
- 3 most dreadful experiences
- Conclusions
- Questions and Answers

*I will wait for you...*

# Dyslexia

Dyslexia is a difficulty in the acquisition of fluent reading and writing skills that is neurological in origin.

*I will wait for you...*

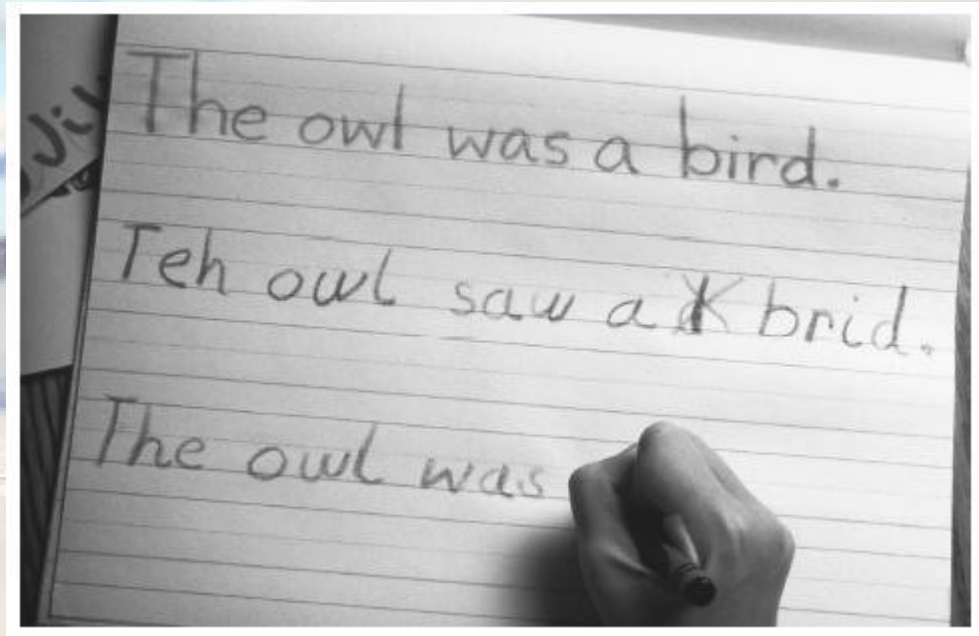
# Dyslexia

For our purpose, *Dysgraphia* is a deficiency in the ability to write (or type), not due to intellectual impairment.

p, q, b, d

*I will wait for you...*

# Dyslexia



*I will wait for you...*

# Famous Dyslexics

Walt Disney (1901 - 1966)



*I will wait for you...*

Keira Knightley, and...

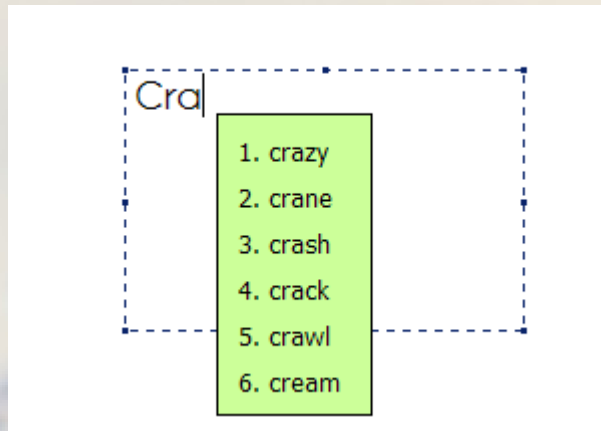
# Dyslexia



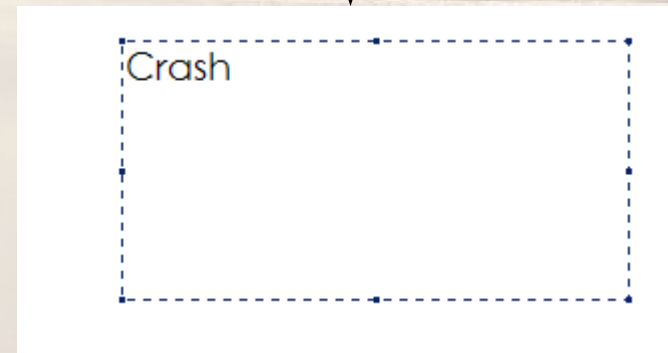
*I will wait for you...*

MM Lee Kuan Yew

# BlitzType



**Choose 3**



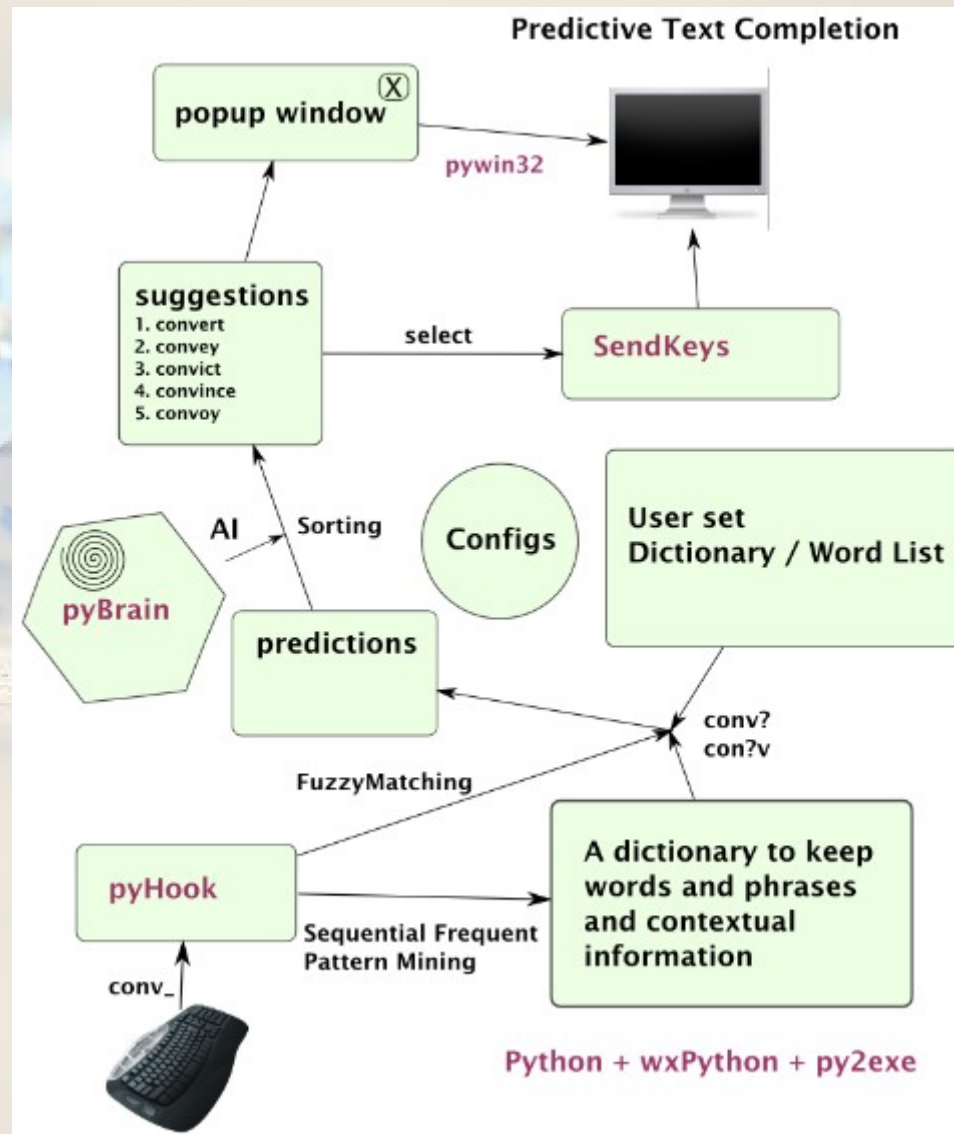
<http://www.artwinauto.com>

Companion project:  
TypingAid in AutoHotkey  
scripting language

*for you...*



# BlitzType



# 3 most exciting experiences

## #3 SendKeys

```
>>> from SendKeys import SendKeys  
>>> SendKeys("{PAUSE 3}Hello{SPACE}World!")  
>>> Hello World!
```

*I will wait for you...*

# 3 most exciting experiences

Finished!

2393505 function calls (2392466 primitive calls) in 36.761 CPU seconds

Ordered by: standard name

ncalls	tottime	percall	cumtime	percall	filename:lineno(function)
1	0.000	0.000	0.000	0.000	<string>:1(<module>)
1	0.000	0.000	0.000	0.000	<string>:1(ArgInfo)
1	0.000	0.000	0.000	0.000	<string>:1(ArgSpec)
1	0.000	0.000	0.000	0.000	<string>:1(Arguments)
1	0.000	0.000	0.000	0.000	<string>:1(Attribute)
1	0.000	0.000	0.000	0.000	<string>:1(DBNotFoundError)
1	0.000	0.000	0.000	0.000	<string>:1(Match)
1	0.000	0.000	0.000	0.000	<string>:1(ModuleInfo)
1	0.000	0.000	0.000	0.000	<string>:1(Traceback)

## #2 Profiling

*I will wait for you...*

```
python -m cProfile myscript.py
```

# 3 most exciting experiences

<u>ncalls</u>	<u>tottime</u>	<u>percall</u>	<u>cumtime</u>	<u>percall</u>	<u>filename:lineno(function)</u>
1	0.03	0.03	36.76	36.76	{execfile}
1	0.05	0.05	36.74	36.74	blitztype.pyw:1(<module>)
1	0	0	24.4	24.4	_core.py:7296(MainLoop)
1	0	0	24.4	24.4	_core.py:8005(MainLoop)
1	12.37	12.37	24.4	24.4	{wx_core_PyApp_MainLoop}
16	0	0	10.66	0.67	blitztype.pyw:81(OnEvent)
14	0	0	10.63	0.76	TypingAnalyzer.py:667(MatchChar)
15/14	1.41	0.09	10.63	0.76	TypingAnalyzer.py:508(MatchChar)
17	0	0	10.56	0.62	HookManager.py:329(KeyboardSwitch)
12	0	0	10.56	0.88	AWAHook.py:61(OnKeyDown)
1	1.83	1.83	6.62	6.62	PhraseListLoader.py:6(__init__)
58011	1.27	0	6.5	0	FuzzyMatching.py:60(match)
112639	1.9	0	4	0	SimpleTools.py:37(index)
1	0	0	3.4	3.4	TypingAnalyzer.py:368(__init__)
2	0	0	3.4	1.7	DB.py:31(__init__)
2	0.02	0.01	3.4	1.7	DB.py:48(LoadData)
2	0.01	0	3.29	1.65	pickle.py:1369(load)
2	0.72	0.36	3.28	1.64	pickle.py:845(load)
41412	1.22	0	2.55	0	FuzzyMatching.py:38(TryToMatch)
1	0	0	2.48	2.48	TypingAnalyzer.py:228(__init__)
70035	1.11	0	2.26	0	SimpleTools.py:27(CountFilter)
29425	0.24	0	1.65	0	FuzzyMatching.py:120(NoOfIncorrectMatches)

# 3 most exciting experiences

```
#PluginFramework.py  
  
class PluginMount(type):  
    def __init__(cls, name, bases, attrs):  
        if not hasattr(cls, 'plugins'):  
            cls.plugins = []  
        else:  
            cls.plugins.append(cls)
```

## #1 Simple Plugin Framework by Marty Alchin

Define a Mount Point	The place where you want to extend the program.	Subclassing PluginMount (metaclass)
Define a Plugin	Program extension	Subclassing the Mount Point
Iterate over all Plugins		Iterate over 'plugins' class attribute of the Mount Point

# 3 most exciting experiences

Define a  
mount point:

Subclass  
PluginMount

```
from PluginFramework import PluginMount

class MountPoint:
    """
    Documentation
    """
    __metaclass__ = PluginMount

    #other class attributes...
    #other methods...
```

*I will wait for you...*

# 3 most exciting experiences

Define a  
Plugin:

Subclass  
Mount  
Point

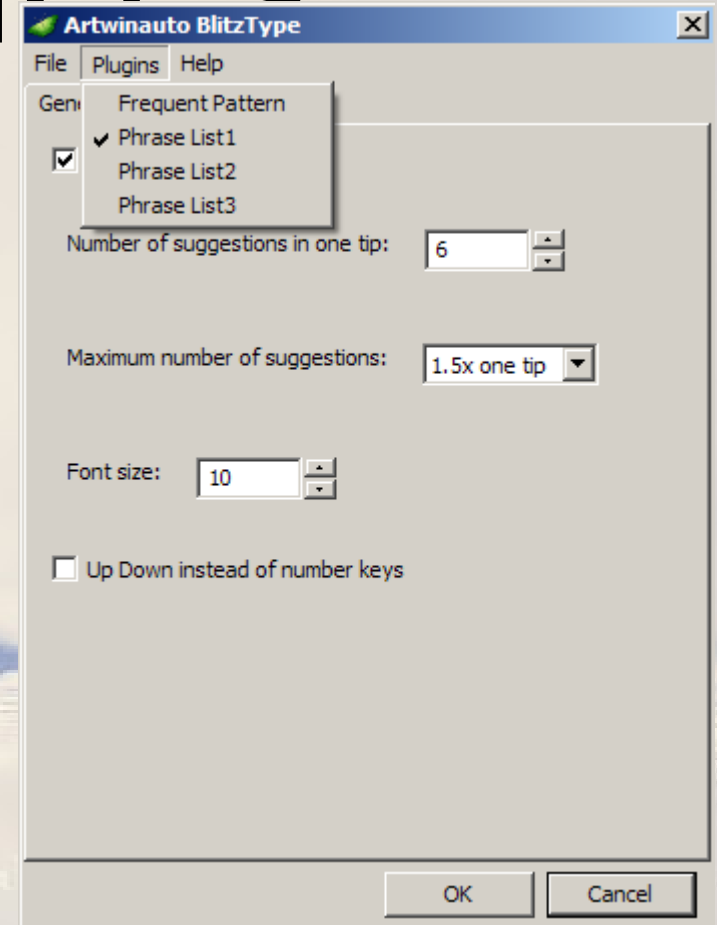
```
class FrequentPatternDB(DB.TimeStampDB, dictionary.MountPoint):  
    """A database class to store frequent patterns  
    It is a dictionary plugin.  
    """  
  
    name = "Frequent Pattern"  
  
    def __init__(self, *args, **kwargs):  
        DB.TimeStampDB.__init__(self, *args, **kwargs)  
        dictionary.MountPoint.__init__(self)  
        self.SortingFunctionParameters=(-0.002, 0.04, -0.5, 0.3, 0.5, 1.5)  
        self.PreviousTimeStamp = []  
        self.activated = False
```

*I will wait for you...*

# 3 most exciting experiences

Iterate over all plugins:

**for DictionaryPlugin in dictionary.MountPoint.plugins:**





# 3 most dreadful experiences

```
#in AHK scripting language:
#CoordMode, ToolTip, Screen
#A_CaretX, A_CaretY

def GlobalGetCaretPos():
    """Get the coordinates of the caret in the active window"""
    caretW = win32gui.GetForegroundWindow()
    if caretW:
        FGWinThreadProcessId = win32process.GetWindowThreadProcessId(caretW)
        CurrentThreadId = win32api.GetCurrentThreadId()
        if FGWinThreadProcessId[0] != CurrentThreadId:
            win32process.AttachThreadInput(CurrentThreadId, FGWinThreadProcessId[0], True)
            cPosR = win32gui.GetCaretPos()
            focusW = win32gui.GetFocus()
            if focusW:
                cPos = win32gui.ClientToScreen(focusW, cPosR)
            else:
                cPos = win32gui.ClientToScreen(caretW, cPosR)
        else:
            cPos = (0, 0)
    return cPos
```

## #3 Pywin32 - not fun

# 3 most dreadful experiences

```
# in AHK scripting language
# clipboard = my text ;Give the clipboard entirely new contents.

def SetText(string):
    """Copy string into clipboard"""
    try:
        win32clipboard.OpenClipboard()
        try:
            win32clipboard.EmptyClipboard()
            win32clipboard.SetClipboardText(string)
        except Exception, e:
            log.exception(e)
            return 1
        finally: #to prevent locking of clipboard
            win32clipboard.CloseClipboard()
            time.sleep(CloseDelay)
    except Exception, e:
        log.exception(e)
        return 1
```

#3 Pywin32 – not fun

# 3 most dreadful experiences

## #2 wxPython

- Unstable
- Crash on exit, bug occurs at C++ level
- (can't be caught by Python exception handling)
- Lucky! SVN comes to rescue.

*I will wait for you...*

# 3 most dreadful experiences

## #1 py2exe - nightmare

- msvcp90.dll not found
- Failed to find certain modules in wxPython
- DLLs not found: need MS Visual C++ runtime
- Code compiled in Vista couldn't run on XP.

# Conclusions

1. Overall, it's fun / fast / productive using Python.

2. With the correct libraries, techniques, one can do almost anything with Python.

3. Time and patience will pay off.

*I will wait for you...*

# I need a job

Wong Jiang Fung

Mobile No.: 96826204

Email: [kakarukeys@gmail.com](mailto:kakarukeys@gmail.com)

From science background. Experience in engineering field, R&D, software development in Python.

Programming Languages: Python, C.

Web programming: Python/Django framework, HTML, CSS, Javascript (AJAX & jQuery), SQL databases.

Operating Systems: Windows, Linux.

Looking for Software Engineer position.

*I will wait for you...*