PYTHON FOR DELPHI DEVELOPERS

Webinar by Kiriakos Vlahos (aka PyScripter)
and Jim McKeeth (Embarcadero)

CONTENTS

PYTHON: WHY SHOULD I (DELPHI DEVELOPER) CARE?

Massive increase in popularity

Language of choice for Data Analytics and Machine Learning/Artificial Intelligence

Rapidly replacing Java as the core programming language in Computer Science degrees

Huge number of packages available (250K at PyPI)

All the latest and greatest open-source libraries are available to Python immediately

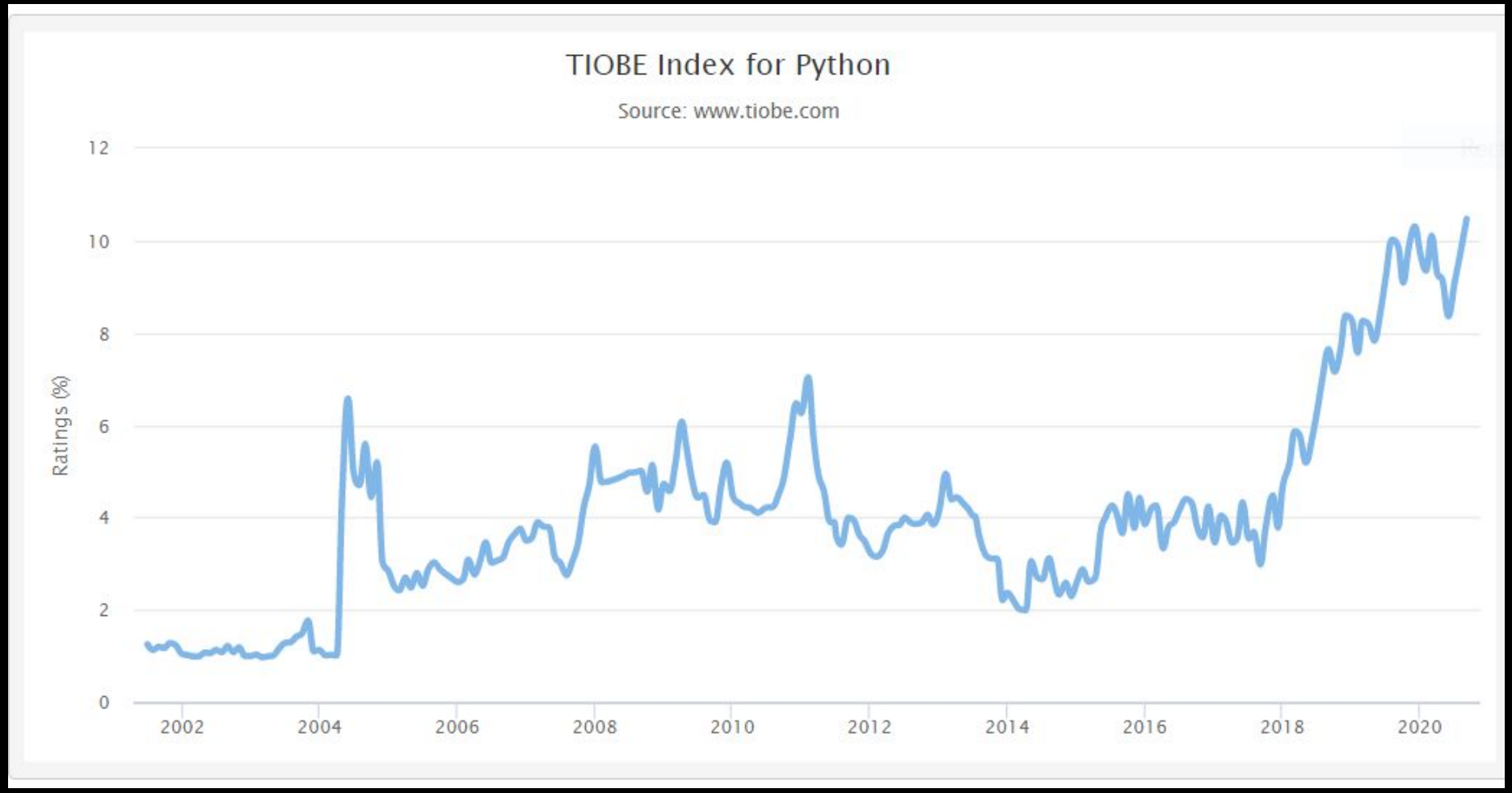
Perceived as productive and easy to learn

Complementary strengths with Delphi

PYTHON-DELPHI : POTENTIAL SYNERGIES

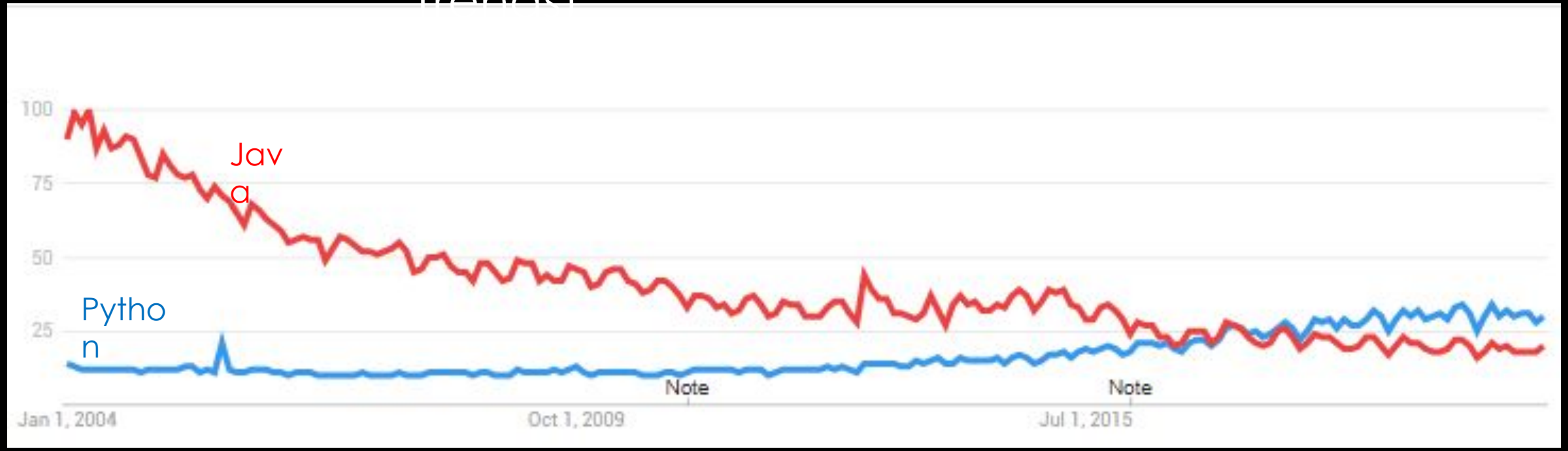
- Gain access to Python libraries from your Delphi applications
- Use Python as a scripting language for Delphi applications
- Make code developed in Delphi accessible from python scripts
- Bring together RAD and GUI Delphi development with python programming
- Combine the strengths of each language

POPULARITY OF PYTHON



PYTHON VS. JAVA

Interest over time (Google Trends)



DELPHI VS PYTHON

	Delphi/Pascal	Python
Maturity	✓ (1995/1970!)	✓ (1989)
Object orientation	✓	✓
Multi-platform	✓	✓
Verbosity	High (begin end)	Low (indentation based)
REPL	No	Yes
Typing	Strong static typing	Dynamic (duck) typing
Memory management	Manual	Reference counting
Compiled	✓	bytecode
Performance	👍	👎
Multi-threading	👍	👎
RAD	👍	👎

PYTHON FOR DELPHI (I)

Low-level access to the python API

High-level bi-directional interaction with Python

Access to Python objects using Delphi custom variants

Wrapping of Delphi objects for use in python scripts using RTTI

Creating python extension modules with Delphi classes and functions

PYTHON FOR DELPHI (II)

- Delphi version support
  - 2009 or later
- Platform support
  - Windows 32 & 64 bits
  - Linux
  - MacOS
- Mostly non-visual components
  - Can be used in console applications
- Lazarus/FPC support

GETTING STARTED – INSTALLING PYTHON

- Select a Python distribution
  - www.python.org (official distribution)
  - Anaconda (recommended for heavy data-analytics work)
- Python 2 vs. **Python 3**
- 32-bits vs. **64-bits**
- Download and run the installer
- Installation options (location, for all users)
- Install python packages you are planning to use (can be done later)
  - Use the python package installer (pip) from the command prompt
  - eg. > pip install numpy

GETTING STARTED – INSTALLING PYTHON FOR DELPHI

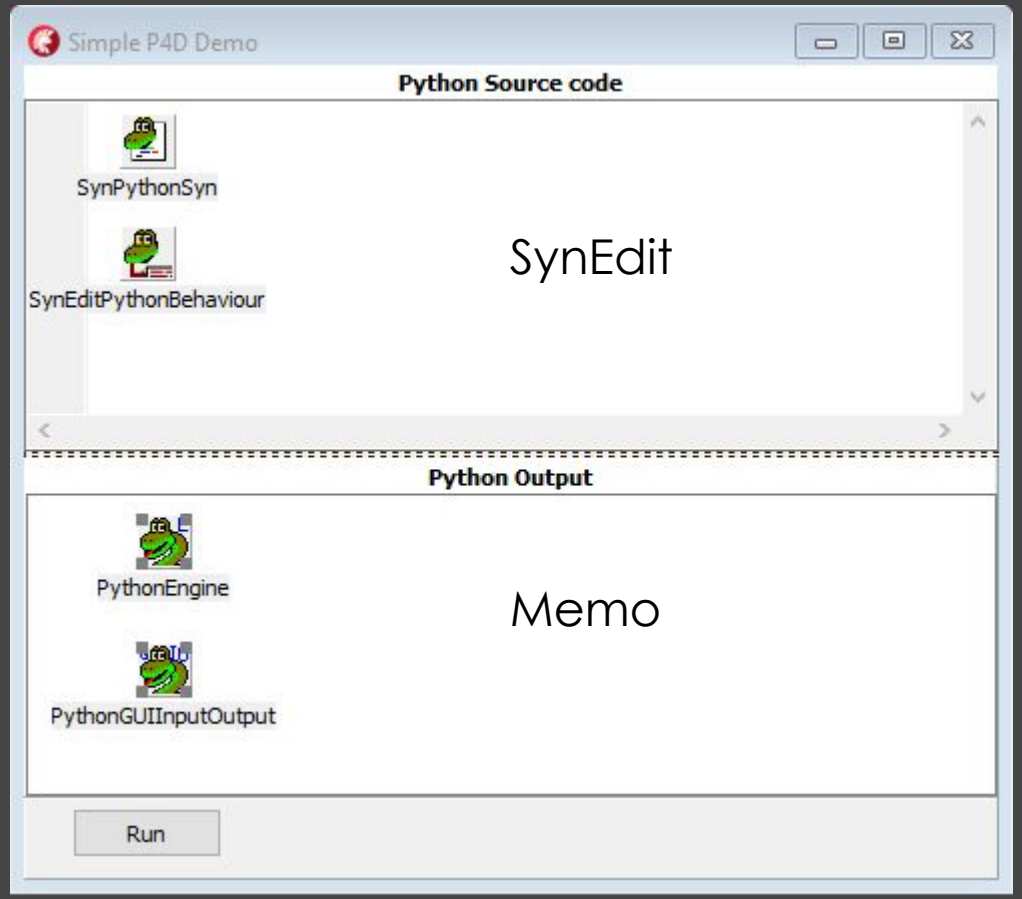
- **Clone** or download and unzip the Github repository into a directory (e.g., D:\Components\P4D).
- Start RAD Studio.
- Add the source subdirectory (e.g., D:\Components\P4D\Source) to the IDE's library path for the targets you are planning to use.
- Open and install the Python4Delphi package specific to the IDE being used. For Delphi Sydney and later it can be found in the Packages\Delphi\Delphi 10.4+ directory. For earlier versions use the package in the Packages\Delphi\Delphi 10.3- directory.

Note: The package is Design & Runtime together

P4D COMPONENTS

Component	Functionality
PythonEngine	Load and connect to Python. Access to Python API (low-level)
PythonModule	Create a Python module in Delphi and make it accessible to Python
PythonType	Create a python type (class) in Delphi
PythonInputOutput	Receive python output
PythonGUIInputOutput	Send python output to a Memo
PyDelphiWrapper	Make Delphi classes and objects accessible from Python (hi-level)
VarPython	Hi-level access to Python objects from Delphi using custom variants (unit not a component)

SIMPLE DEMO (I)

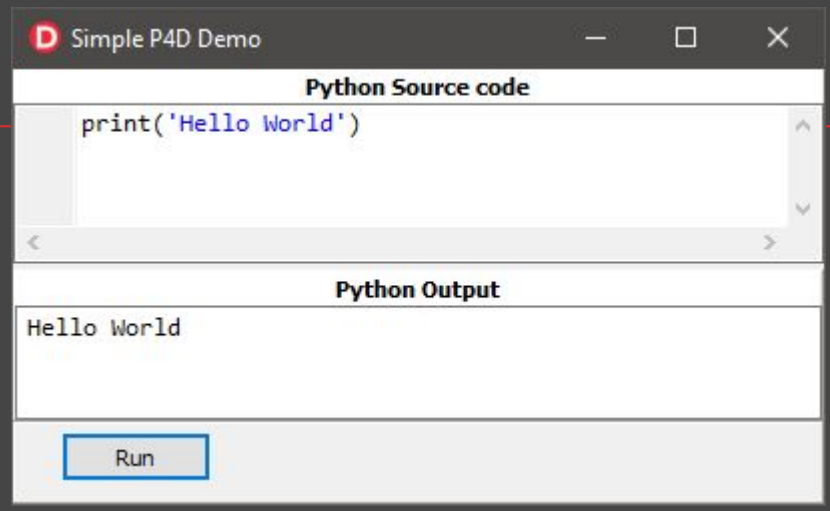


SIMPLE DEMO (II)

- All components are using default properties
  - PythonGUIInputOutput linked to PythonEngine and Memo
- Source Code:

```
procedure TForm1.btnRunClick(Sender: TObject);
begin

GetPythonEngine.ExecString(UTF8Encode(sePythonCode.Text));
end;
```



USING TPYTHONMODULE (I)

Python

```
def is_prime(n):
    """ totally naive implementation """
    if n <= 1:
        return False

    q = math.floor(math.sqrt(n))
    for i in range(2, q + 1):
        if (n % i == 0):
            return False
    return True
```

Delphi

```
function IsPrime(x: Integer): Boolean;
begin
  if (x <= 1) then Exit(False);

  var q := Floor(Sqrt(x));
  for var i := 2 to q do
    if (x mod i = 0) then
      Exit(False);
  Exit(True);
end;
```

USING TPYTHONMODULE (II)

Python

```
def count_primes(max_n):
    res = 0
    for i in range(2, max_n + 1):
        if is_prime(i):
            res += 1
    return res

def test():
    max_n = 1000000
    print(f'Number of primes between 0 and {max_n} = {count_primes(max_n)}')

def main():
    print(f'Elapsed time: {Timer(stmt=test).timeit(1)} secs')

if __name__ == '__main__':
    main()
```

Output

```
Number of primes between 0 and 1000000 = 78498
Elapsed time: 3.4528134000000037 secs
```

USING TPYTHONMODULE (III)

- Add a TPythonModule to the form and link it to the PythonEngine
  - ModuleName: delphi_module
  - Implement python function delphi_is_prime by writing a Delphi event

```
procedure TForm1.PythonModuleEvents0Execute(Sender: TObject; PSelf, Args: PPyObject; var Result: PPyObject);
Var
  N: Integer;
begin
   with GetPythonEngine do
     if PyArg_ParseTuple(Args, 'i:delphi_is_prime',@N) <> 0 then
     begin
       if IsPrime(N) then
         Result := PPyObject(Py_True)
       else
         Result := PPyObject(Py_False);
       Py_INCREF(Result);
     end else
       Result := nil;
end;
```

USING TPYTHONMODULE (IV)

Python

```
from delphi_module import delphi_is_prime
def count_primes(max_n):
    res = 0
    for i in range(2, max_n + 1):
        if delphi_is_prime(i):
            res += 1
    return res
```

Output

```
Number of primes between 0 and 1000000 = 78498
Elapsed time: 0.30737420000000017 secs
```

10x + improvement!

But hold on. Delphi can do something python can't do easily: Use threads and multiple CPU cores

USING TPYTHONMODULE (V)

- Implement delphi_count_primes using TParallel.For

```
function CountPrimes(MaxN: integer): integer;
begin
  var Count := 0;
  TParallel.&For(2, MaxN, procedure(i: integer)
    begin
      if IsPrime(i) then
        AtomicIncrement(Count);
    end);
  Result := Count;
end;
```

70x + improvement!

Output

```
Number of primes between 0 and 1000000 = 78498
Elapsed time: 0.0470959000000219 secs
```

Python

```
from delphi_module import delphi_count_primes
from timeit import Timer
import math

def test():
    max_n = 1000000
    print(f'Number of primes between 0 and {max_n} = {delphi_count_primes(max_n)}')
```

USING PYDELPHIWRAPPER

- PyDelphiWrapper allows you to expose Delphi objects, records and types using RTTI and cusomised wrapping of common Delphi objects.
- Add a TPyDelphiWrapper on the form and link it to a PythonModule.
- In this demo we will wrap a Delphi record containing a class function.

```
type
  TDelphiFunctions = record
    class function count_primes(MaxN: integer): integer; static;
  end;

var
  DelphiFunctions: TDelphiFunctions;
```

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  var Py := PyDelphiWrapper.WrapRecord(@DelphiFunctions,
    TRttiContext.Create.GetType(TypeInfo(TDelphiFunctions))
    as TRttiStructuredType);
  PythonModule.SetVar('delphi_functions', Py);
  PythonEngine.Py_DecRef(Py);
end;
```

WRAPDELPHI DEMO31

- Shows you how you can create Delphi GUIs with Python
  - Create forms
  - Subclass Forms (and other Delphi types)
  - Add python Form event handlers
- Use customized wrapping of common RTL and VCL objects
  - Common Dialogs
  - StringLists
- Exception handling

CONCLUSIONS

- With Python for Delphi you can get the best of both worlds
- P4D makes it very easy to integrate Python into Delphi applications in RAD way
- Expose Delphi function, objects, records and types to Python using low or high-level interfaces
- In a future webinar we will cover
  - Using python libraries and objects in Delphi code (VarPyth)
  - Python based data analytics in Delphi applications
  - Creating Python extension modules using Delphi
  - Python GUI development using the VCL

RESOURCES

- Python4Delphi library
  - https://github.com/pyscripter/python4delphi
- PyScripter IDE
  - https://github.com/pyscripter/pyscripter
- Thinking in CS – Python3 Tutorial
  - https://openbookproject.net/thinkcs/python/english3e/
- PyScripter blog
  - https://pyscripter.blogspot.com/
- Webinar blog post
  - https://blogs.embarcadero.com/python-for-delphi-developers-webinar/