

FUTURE DECODED

6-7 OTT '16 / MILANO

IN PARTNERSHIP WITH:



CommunityDays.it

www.futuredecoded.it

 #FutureDecoded

Python Tools for Visual Studio

Introducing Python to the .NET crowd

by Nicola Iarocci

@nicolaiarocci

nicolaiarocci.com

www.futuredecoded.it

 #FutureDecoded



OVERVIEW

- Python in 5 minutes
- Python from the C# perspective
- Python Tools for Visual Studio
- Demo
- Take aways

PYTHON IN A NUTSHELL

- High level, open source programming language
- Object oriented
- Interpreted (sometimes JIT compiled)
- Strongly typed with dynamic semantics
- Syntax emphasizes readability
- Supports modules and packages
- Cross platform since the beginning
- Very mature: 20+ years in the making
- Comes with batteries included
- Rich ecosystem
- Strong, welcoming and passionate community

EMPHASIS ON READABILITY

```
def hello(name):  
    if name == "david bowman":  
        print("Hi old friend")  
    else:  
        print("Nice to meet you")  
        print("My name is HAL9000")  
  
def main():  
    guest = input("What is your name?")  
    hello(guest)
```

also super easy to write!

PYTHON SYNTAX

```
def hello(name):
```

```
    if name == "Nicola":
```

```
        print("Hi old friend")
```

```
    else:
```

```
        print("Nice to meet you")
```

```
        print("My name is HAL9000")
```

```
def main():
```

```
    guest = input("What is your name?")
```

```
    hello(guest)
```

- codeblocks are defined with whitespace and colons
- code blocks start with `:`
- whitespace really matters
- there are no braces
- there are no parentheses
- there are no semicolons
- spaces over tabs by convention
- IDEs and editors follow conventions!

C# AND PYTHON

Code snippets courtesy of Michael Kennedy | @mkennedy | <http://talkpython.fm>

EVERYTHING IS AN OBJECT

C#

```
class Document : object
{
    public void Serialize()
    {
        // ...
    }

    public override string ToString()
    {
        return "I am a document";
    }
}
```

Python

```
class Document( object ):
    def serialize(self):
        # ...

    def __str__(self):
        return "I am a document."
```

IENUMERABLE + FOREACH LOOPS

C#

```
int[] numbers = new[] {1, 2, 3, 4, 5, 6};

foreach (var n in numbers)
{
    Console.Write(n + ",");
}
```

Python

```
numbers = [1, 2, 3, 4, 5, 6]

for n in numbers:
    print(n, end=', ')
```

IENUMERABLE + FOREACH LOOPS

C#

```
class ShoppingCart : IEnumerable<Tuple<string, float>>
{
    List<Tuple<string, float>> cartItems =
        new List<Tuple<string, float>>();

    public void Add(string name, float price)
    {
        cartItems.Add(new Tuple<string, float>(name, price));
    }

    public IEnumerator<Tuple<string, float>> GetEnumerator()
    {
        return cartItems.GetEnumerator();
    }

    IEnumerator IEnumerable.GetEnumerator()
    {
        return GetEnumerator();
    }
}
```

Python

```
class ShoppingCart:

    def __init__(self):
        self.items = []

    def add(self, name, price):
        self.items.append( (name, price) )

    def __iter__(self):
        return self.items.__iter__()
```

PROPERTIES

C#

```
class ShoppingCart
{
    public float TotalPrice
    {
        get
        {
            float total = 0;
            foreach (var item in cartItems)
            {
                total += item.Item2;
            }

            return total;
        }
    }

    Console.WriteLine("Total price: {0}", cart.TotalPrice);
}
```

Python

```
class ShoppingCart:

    @property
    def total_price(self):
        total = 0.0
        for item in self.items:
            total += item[1]

        return total

    print("Total is {0}".format(cart.total_price))
```

LAMBDA EXPRESSIONS

C#

```
private static IEnumerable<int>
FindNumbers(Predicate<int> predicate)
{
    for (int i = 0; i < 100; i++)
    {
        if (predicate(i))
            yield return i;
    }
}

IEnumerable<int> nums =
    FindNumbers(n => n % 11 == 0)

// nums =[0, 11, 22, 33, 44, 55, 66, 77, 88, 99]
```

Python

```
def find_numbers(predicate):
    for i in range(100):
        if predicate(i):
            yield i

nums = find_numbers(lambda n : n % 11 == 0)

# nums = [0, 11, 22, 33, 44, 55, 66, 77, 88, 99]
```

ITERATOR METHODS & YIELD RETURN

C#

```
private static IEnumerable<int>
    FibonacciGenerator()
{
    int current = 1;
    int next = 1;

    yield return current;
    while (true)
    {
        int temp = current + next;
        current = next;
        next = temp;
        yield return current;
    }
}
```

Python

```
def fibonacci_generator():
    current, nxt = 1, 1
    yield current

    while True:
        current, nxt = nxt, current + nxt
        yield current
```

PACKAGE MANAGEMENT

C#

```
PM>Install-Package mongocsharpdriver
```

Installing 'mongocsharpdriver 1.9.1'.

Successfully installed 'mongocsharpdriver 1.9.1'.

Adding 'mongocsharpdriver 1.9.1' to YourApp.

Successfully added 'mongocsharpdriver 1.9.1' to YourApp

Python

```
c:\>pip install pymongo
```

Downloading/unpacking pymongo

 Running setup.py egg_info for package pymongo

Installing collected packages: pymongo

 Running setup.py install for pymongo

 Fixing build\lib.win-amd64-3.4\bson\binary.py...

 Successfully installed pymongo

 Cleaning up...

around 57,000 packages on NuGet

around 82,000 packages on PyPI

MORE ON PYTHON

- Python: An amazing second language for .NET developers,
by Michael Kennedy
- Learn Python the Hard Way,
by Zed Shaw
- Python in 10 minuti (Italian),
by Nicola Iarocci
- The Python Tutorial,
python.org
- Google

PYTHON AND VISUAL STUDIO

Did you know that you can do Python with your favourite IDE?

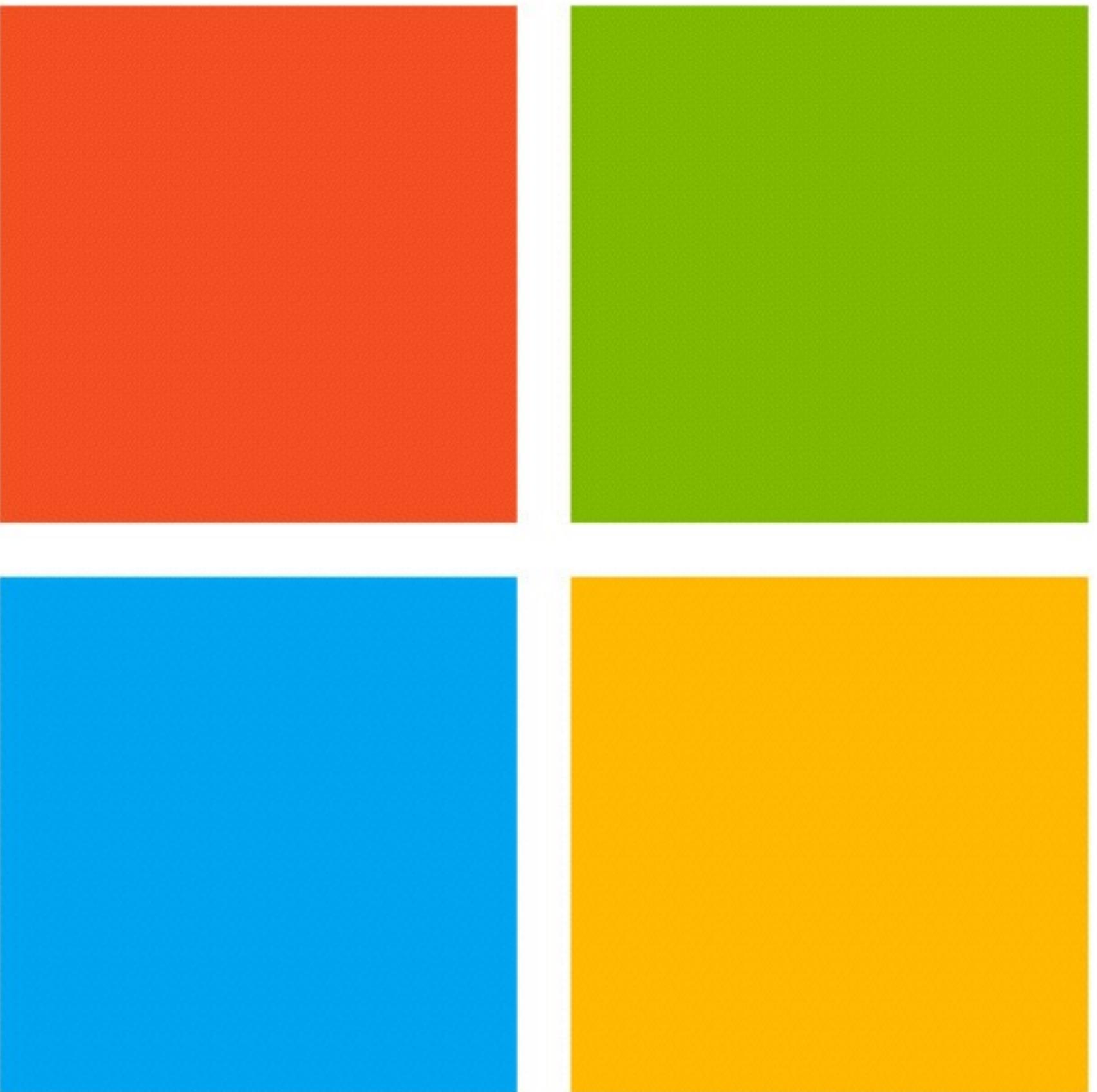
#1

OFFICIAL

MICROSOFT

EXTENSION

Created and maintained by Microsoft



Microsoft

FREE AND OPEN SOURCE

#2

Fork us on GitHub

This repository Search Pull requests Issues Gist

Microsoft / PTVS

Code Issues 421 Pull requests 1 Wiki Pulse Graphs

Python Tools for Visual Studio <https://microsoft.github.io/PTVS>

2,693 commits 7 branches 21 releases

Branch: master New pull request New file Find file SSH git@git

zooba Merge pull request #1079 from zooba/forward-port ...

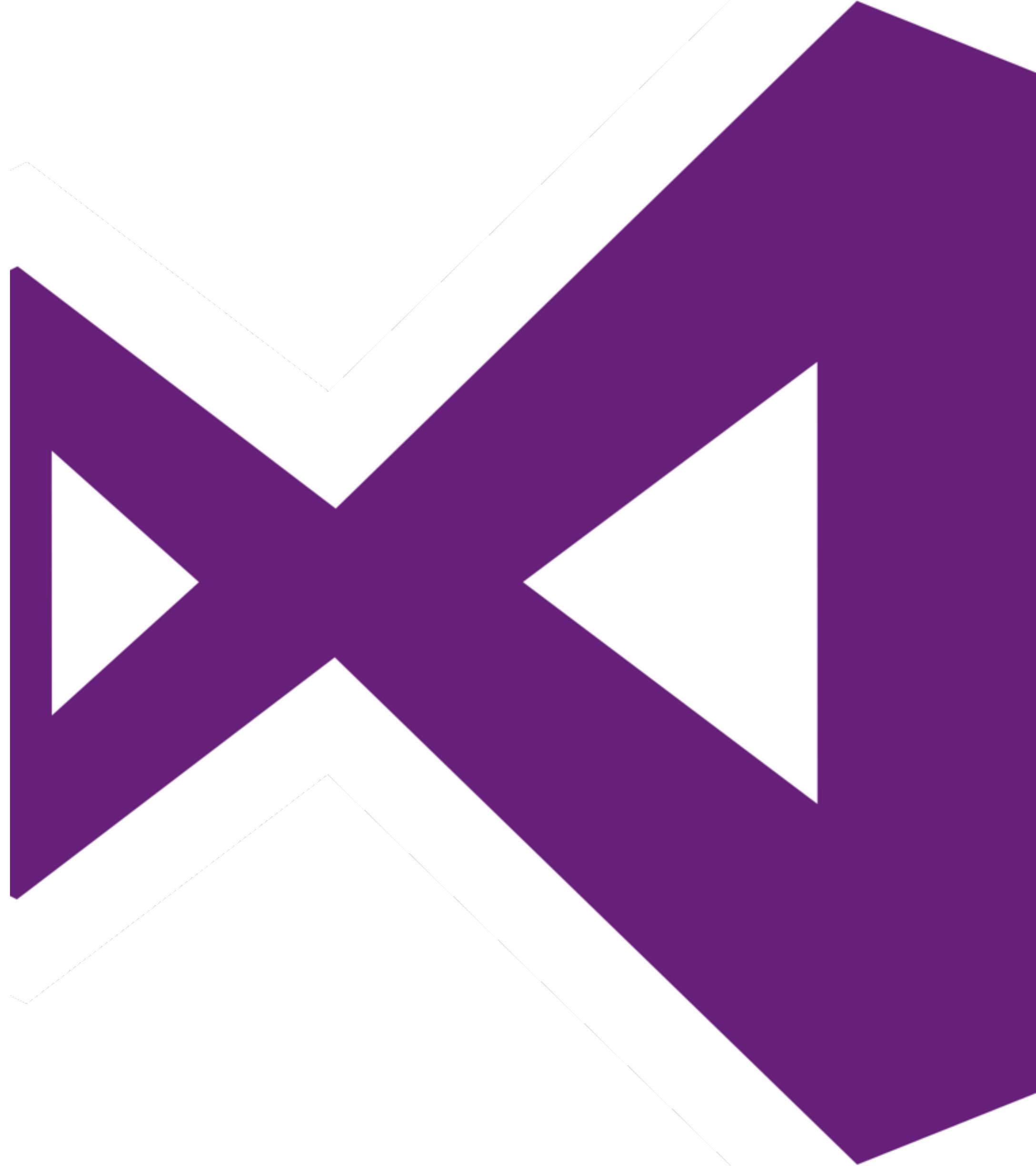
Build	Adds missing assembly resolution path.
Common	Added Common and VSCommon assemblies for shared code and
Python	Handles newline normalization
.gitattributes	Mark .sln as non-text file, since git doesn't properly handle UTF-16
.gitignore	Updates ignore files.
.hgeol	Updates ignore files.
.hgignore	Updates ignore files.
LICENSE	Restore deleted files.
PowershellEnvironment.ps1	Normalizes all line endings.
README.md	Update readme for 2.2.
build.root	Normalizes all line endings.
dirs.proj	Normalizes all line endings.
README.md	

Python Tools for Visual Studio

#3

WORKS WITH VISUAL STUDIO COMMUNITY EDITION

100% free Python experience



#4

CYTHON
PYPY
IRONPYTHON
AND MORE

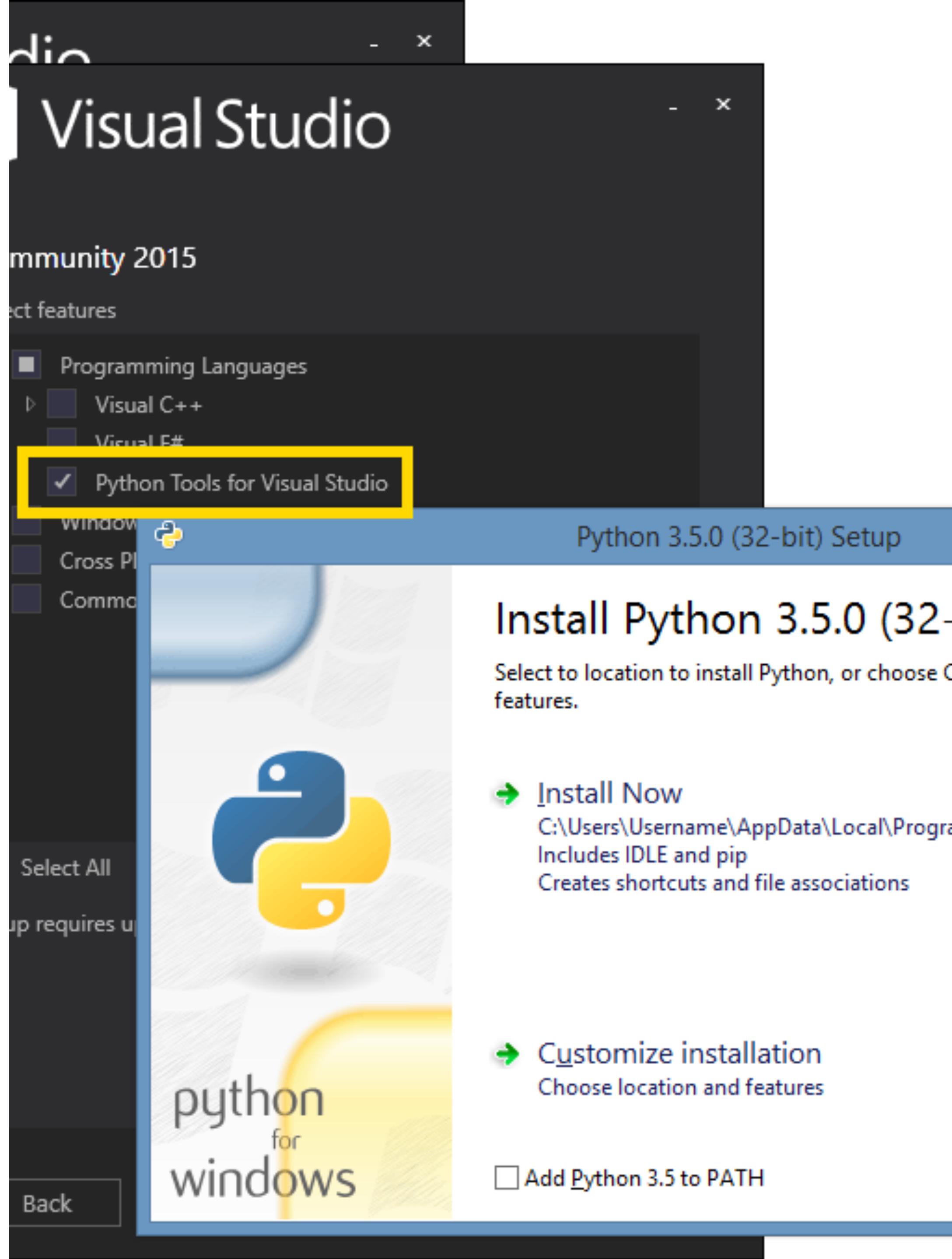
all kind of Pythons supported



#5

SUPPORTED BY SETUP TOOL

select in custom install , or install later



#6 PACKAGE MANAGEMENT

find and install the packages you need

A screenshot of a search results page for "django" on PyPI. The search bar at the top contains the text "django". Below the search bar, there is a blue square icon. The search results list several packages:

- Django (1.7.8)** - This item is highlighted with a green icon and has a blue "Install django" button below it.
- "pip install django" from PyPi
- Install django
- Install django12factor
- Install django2schematics
- Install django403handler
- Install django4facebook

A hand cursor icon is shown pointing at the "Install django" button for the Django package. A callout box with the text "Django is a high-level Python Web framework" is positioned over the "Install django" button for the Django package.

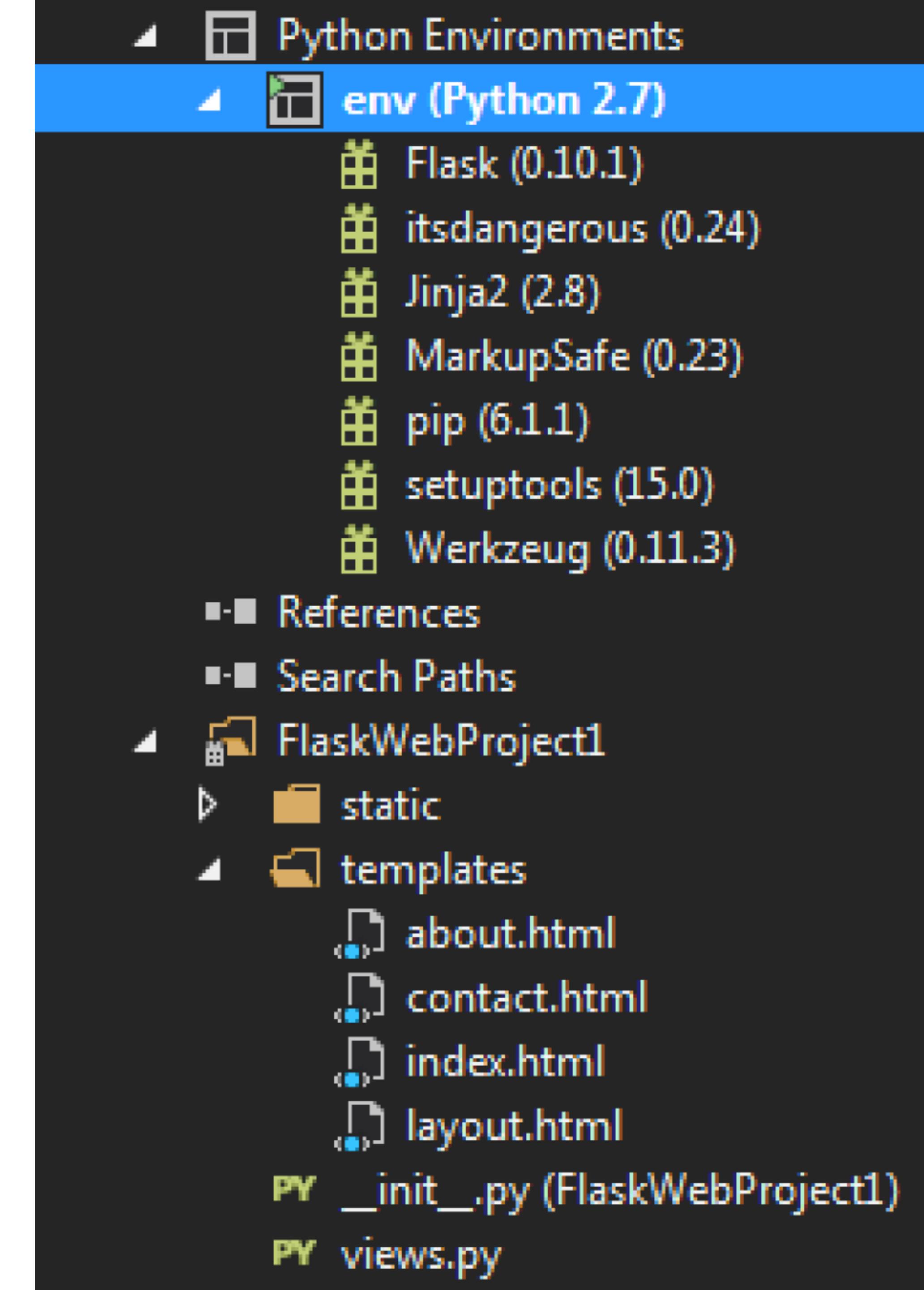
SHALL HE PLAY A GAME?

OR, BUILDING A GAME WITH PYTHON AND VISUAL STUDIO

#7

VIRTUAL ENVIRONMENTS

add project -specific virtual environments, or use global ones



INTELLISENSE

*like it's static: an editor that knows
your code*

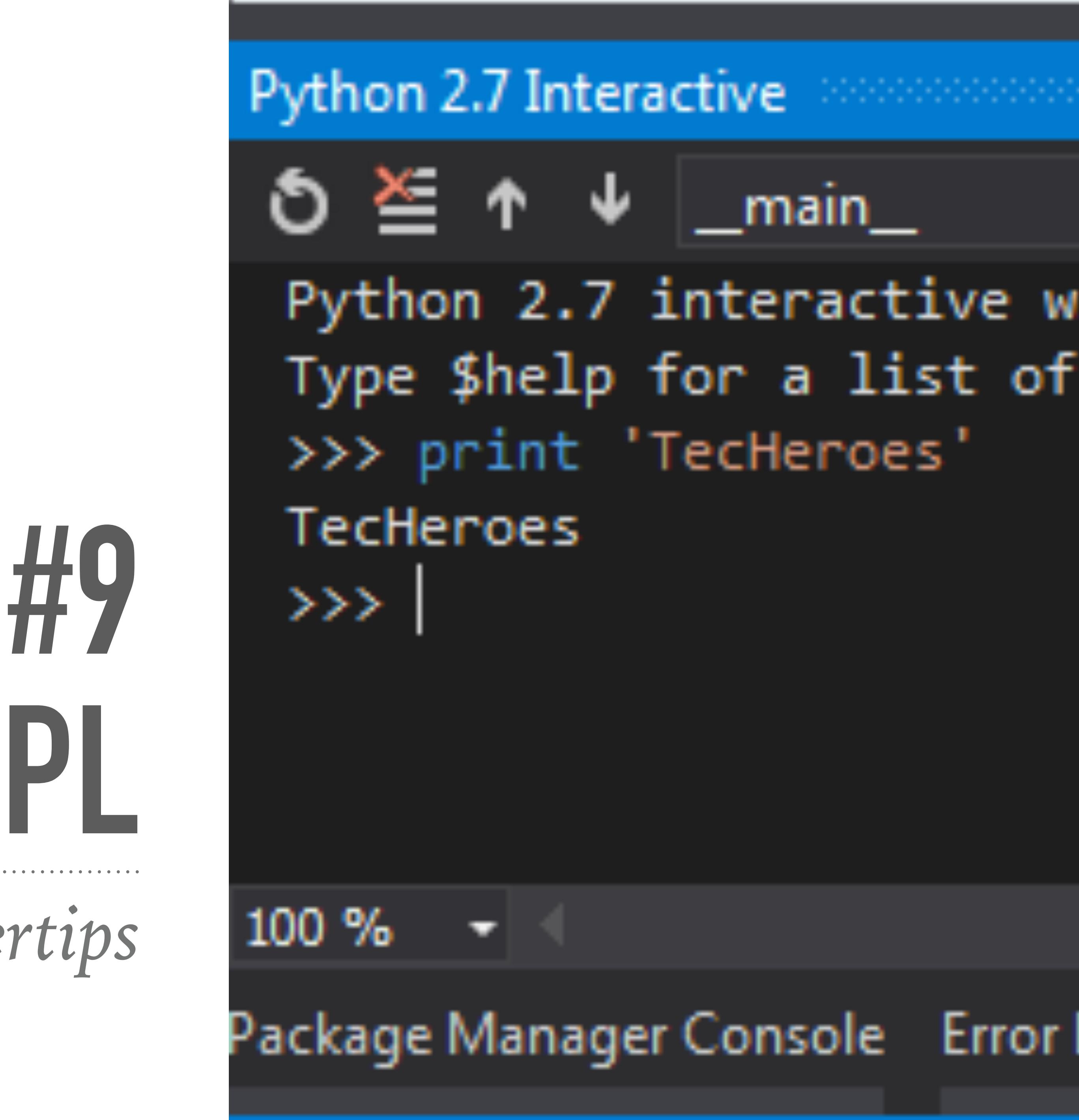
#8

```
class MyClass(object):  
    def do_stuff(self,  
                for i in value:  
                    i.|
```

```
class Person(object):  
    def __init__(self, name, age, height, location, walk, meet, is_adult):  
        self.name = name  
        self.age = age  
        self.height = height  
        self.location = location  
        self.walk = walk  
        self.meet = meet  
        self.is_adult = is_adult
```

PYTHON REPL

at your fintertips



The screenshot shows a Python 2.7 Interactive window. The title bar reads "Python 2.7 Interactive". The toolbar includes icons for file operations and a dropdown menu set to "_main_". The main area displays the Python prompt and some sample code:

```
Python 2.7 interactive w
Type $help for a list of
>>> print 'TechHeroes'
TechHeroes
>>> |
```

The bottom navigation bar includes "100 %", "Package Manager", "Console", and "Error".

#10

LIVE DEBUGGING

enjoy the debugger you are used to

The screenshot shows a debugger interface with two main sections. The top section displays a portion of Python code:

```
def process_image():
    w, h = img.size
    img = img.crop((0, 0, w, h))
    return img
```

The bottom section is a table titled "Autos" showing variable values:

Name	Value
h	2736
img	<PIL.J
img.size	(3648,

DEMO

Build a website with Python and Visual Studio (in 3 minutes)

www.futuredecoded.it

 #FutureDecoded



TAKE AWAYS

- VS Community Edition + Python = free development environment on Windows
- Python folks finally have a great development environment on Windows
- .NET crowd can build stuff in Python using their favorite IDE and toolset
- Python is a great second language for C# developers
- Diversity is good. And it matters.

The background of the slide is a photograph of a city at night, likely Hong Kong, featuring numerous skyscrapers with lights on, a highway with blurred traffic lights, and a bridge in the foreground.

Domande?

Materiale su <http://www.communitydays.it/>

www.futuredecoded.it

 #FutureDecoded



Nicola Iarocci
@nicolaiarocci
nicolaiarocci.com