Why Python?

1. Python is a dynamically typed high level language. Dynamically typed means, you don't have to declare the variable type before using the variables.
   For example:
   A = 10.0
   that means A has a value of 10.0 of type float.
2. It is designed with readability in mind. So no need to use curly braces to represent begin and end of a loop. Indentation is used to represent block of code.
3. Python maybe slow but it is versatile. It can be used to write scientific programs, for web programming, for desktop applications etc.

- Modules like Django, Flask or Pyramid can be used for web development.
- Modules like Pygame, WxPython, etc. can be to develop desktop applications.
- Modules like Numpy and Scipy are used for numerical and scientific programming.
- Modules like Pandas and Theonas are used for data science and machine learning.

4. Python has a supportive community. You can find answers to many Python related questions on stackoverflow. Many people have written blogs on programming in Python.

There are two different versions of Python that are in use, Python 2.7.x and Python 3.x. Here are some of the differences:

Python 2.7.x	**Python 3.x**
print("Python is fun") print "Python is fun"	print("Python is fun")
range(5) will create a list xrange(5) is an iterator	range(5) is an iterator There is no xrange
Raising exceptions raise IOError, "your error message”	raise IOError("your error message”)
User input a = input('enter a value') input() takes integers and floats	There is no input()
a = raw_input('enter a value') The user input for raw_input() type(a) will always be string	a = raw_input('enter a value') The user input for raw_input() type(a) will always be string
Integer division: 3/2 = 1 Unless you use from __future__ import division This will import the floating point division feature from the latest Python version	Integer division: 3/2 = 1.5 This is the default
Unicode support is not a default feature. str() supports ASCII and Unicode() supports Unicode	str() supports Unicode

Urls: http://sebastianraschka.com/Articles/2014_python_2_3_key_diff.html
https://blog.appdynamics.com/devops/the-key-differences-between-python-2-and-python-3/