Java

Java is a statically-typed general purpose programming language, it is an object-oriented and concurrent language. Java was meant to be WORA (write once run anywhere) language, it was designed to run on any platform and with as few dependencies as possible, with the help of the Java Virtual Machine (JVM).

Python

Python is a dynamically-typed general purpose programming language. Python's early development began at a research institute in the Netherlands. The original motivation behind it was to create a higher-level language to bridge the gap between C and the shell, as the author states, creating system administration utilities using C back at that time was pretty complicated. The syntax was also motivated by a few languages like Algol68, Pascal, and ABC and was meant to be readable and clean.

Now let's have a look at key difference between Python and Java.

Python vs Java: Key Differences

Performance

Languages don't have speed, they have only semantics. If you want to compare speed you must choose specific implementations to compare with each other.
Keep in mind that performance is not only a function of the language's execution speed, the program's implementation, and the third-party libraries performance is usually the number one factor in the equation.

Syntax

Python is a dynamically typed language, when you write Python, you don't need to determine variable types, as the interpreter will infer these types and the checks will be made at runtime. Which results in an easier syntax that is quite similar to the English Language. Moreover, Python doesn't use enclosing braces and follows indentation rules (like how most people right pseudocode) which makes the code quite easy to read and friendly for beginners.

```
class Fruit:
  def __init__(mysillyobject, name, color):
    mysillyobject.name = name
    mysillyobject.color = color

  def myfunction(abc):
    print("Hello I'm a  " + abc.name)
  def mycolor(abc):
    print("Hello My color is  " + abc.color)

p1 = Fruit("Apple", "red")
p1.myfunction()
```

Java, on the other hand, follows strict syntax rules, it's a statically typed language where you need to explicitly declare your variable types and shouldn't an anomaly be spotted, the code will not compile, to begin with. While it's not the easiest thing for beginners, some developers find comfort with the clarity of statically typed languages, many developers don't feel comfortable following indentation rules, especially with large code bases.

```
public class Fruit {
     string name;
     string color;
    public Fruit(string name,string color){
        this.color=color;
        this.name=name;
    }
    public void myfunction() { System.out.println("Hello I'm a :" + name ); }
    public void mycolor() { System.out.println("Hello my color is :" + color ); }
};
```

This is the equivalent to the Fruit class we have defined in Python with the exact same functionalities.

Python vs Java: Uses/Applications in various fields

Game Development

We're not going to talk about general PC game development since neither Python nor Java can really compete with C++/C# in that area with their huge ecosystem. Moreover, game development is a field that requires the highest possible performance to provide seamless experiences to the users, and while Java and Python are not slow, they don't provide the best performance for game development.

Web Development

Both languages are used in backend web development. Backend web development is the branch of web development concerned with creating the software that will run on the server. It's the most popular development field according to StackOverflow's developer survey.
Writing your own backend technology from scratch is not only hard, but it's extremely hard to cover all design requirements from **security** to **reliability** and **effectiveness**. This is why developers have created **frameworks** which is an abstraction in software that allows you to build your backend technology without reinventing the wheel.

The most two popular frameworks for **Python** are Django and Flask. Flask is a micro web framework, it gives you the basic functionalities you'd need like routing requests without much overhead. Django is a more featured option and can help you build a powerful backend while capitalizing on efficiency and security, Django is equipped with a powerful ORM layer which facilitates dealing databases and performing different operations on the data.
As for **Java**, Spring is perhaps the most well-known Java backend framework with a massive ecosystem and a huge community around it. Spring is used by Orange, Dell, GE, and many other enterprises, and while it's not as trending as Django nowadays, it is a powerful option for building enterprise level applications.

Machine Learning

Since Python is syntactically very easy yet a fully-fledged general-purpose programming language, it became a popular option for people from different disciplines who wanted to experiment with machine learning and bring the power of AI into their respective fields. That's why a lot of the development in AI and machine learning is done with Python with a huge ecosystem and libraries.
There is TensorFlow, Keras, Sickit-Learn, and Facebook's PyTorch and it's by far the most popular language in the field.
Java is also considered a good option when it comes to machine learning, it's easy to debug and use and it's already being used for large-scale and enterprise level applications. Among the libraries, you could use in that area are Weka, Mallet, DeepLearning4, and MOA.

Python vs Java Comparison Summary

To recap, here's a quick comparison between the two languages covering the main points we discussed.

Technology	Python	Java
Popularity	Very popular	Very popular
Syntax	Easy to learn and use	Complex includes a learning curve
Performance	Slower than Java in various implementations	Relatively very fast
Cross Platform	Yes	Yes, thanks to the JVM
Backend Frameworks	Django, Flask	Spring, Blade
Machine Learning Libraries	Tensorflow, Pytorch,	Weka, Mallet, Deeplearning4j, MOA
Game Development Engines	Cocos, Panda3d	JMonkeyEngine

Java and Python are both capable and popular languages, so there won't be a lack of resources once you choose one and embark on your journey. If you're new to programming, it'd be better to stick with Python just because it's really easy and uses English-like syntax, it's used in many Computer Science introductory courses around the world. However, if your goal is to build enterprise level applications coming from a C/ C++ world, then Java would probably feel pretty familiar to you. It all goes down on what you plan to build and where you feel like journeying with your new skill.