15 Differences Between Java and Python

1. Developer's View :

Python is better than java from developer?s perspective as it reduces lines of code to be written. Less code also re duces chances of making mistakes and can be maintained easily.Python is dynamically typed language and due to this reason programmers can be more productive.

2. Complexity :

Python is much simpler and compact because like java we don't need to write a large amount of code.

As the code is less it's very easy to maintain. But on the other hand it might become a little bit complicated to maintain a very large code in Python as it doesn't use braces.

3. Platform Dependency :

Platform independent means "Write Once Run Anywhere (**WORA**)".Java is platform independent as compiler creates Byte Code which is common for all platforms so it can run across any platform on the other hand Python is not.

4. Execution speed :

In terms of execution of a program Java is faster as compared to Python because the standard distributions (Java) include a JIT compiler that compiles bytecode to native code at runtime. This Just-In-Time (JIT) compilation is the reason for Java?s better performance.

5. Duck Typing :

Java is statically typed language. Python is not dynamically typed language. In Python we never declare anything but this is not the case in java.

For instance, consider we have two types Animal and Car that both have operation "run", however Car additionally supports "refuel". If this program just calls "run" on objects, it doesn't make a difference in the event that they are Animal or Car. This is called "duck typing" after the articulation "if it walks like a duck and quaks like a duck, it's a duck".

There is no need to provide data types to variable in python but in java it is not the case, we need to explicitly provide data types while declaring variable.
e.g

```
<p> //Java </p> <div> </div> <div>int a; 
 //variable of integer type</div> <div>a = 10;</div>
<div> </div> <div> </div>

<p>#Python </p> <div>a = 10;<span style="white-space:pre">
</span>#no need to specify int</div> <div> </div>
```

6. Change in Data Types :

Java doesn't allow data type of a variable to be changed while python does. e.g. Python allows us to store integer value in a variable, which was once used for String type variable.

```
<p> //Java</p> <div>String str = "Java";</div> <div>str
= 10;<span style="white-space:pre"> </span></div>
<div>System.out.println(str);<span style="white-space:pre">
</span>//error (incompatible types)</div>
```

 #Python   String str = 'Python' str = 10; print('The value of str is', str) #Output : 10

7. String functionalities:

Python provides more functionalities for String as compared to java.

a.To remove leading and trailing whitespace from string 'str'
Java :
str.trim()
Python :
str.strip()

b.To remove leading whitespace from string 'str'
Java :
No such functionality is available

Python :
str.lstrip()

c.To remove trailing whitespace from string 'str'
Java :
No such functionality is available

Python :
str.rstrip()

8. Verbosity :

Take a look at the following snippets :

// Java :   if (a > b) {   a = b;   b = c; }

  #Python   if a > b :   a = b   b = c

If we take a look at the code in Java it has more no. of control characters e.g. {} () ; than in python e.g. indentation .

Less the control characters less is the complexity and chances of making mistakes while writing code.

9. Indentation :

Python has one more extra feature than java i.e. Indentation Before going for explanation take a look at following code in python :

Python if a == b: print('Both are equal') else: print('Both are not equal') print('The End!')

The lines, print('Both are equal') and print('Both are not equal') are two separate code blocks. Python allows us to write any number of lines in a code block.

Indenting each line of block with same amount, indicates a block of code in python. In above given code if-statement and else-statement both are indented 4 spaces. And the final print('The End!') is not indented, so it is not part of the else-block. In simple words python replace braces from Java with indents.

10. Comments :

Both the languages use different syntaxes for commenting :

 //this is comment in java

 #this is comment in python

11. Null :

Unlike Java, Python uses 'None'.

 //Java : Object obj1 = null;

Python :   obj1 = None

12. Container Objects :

Container objects in Java, like ArrayList and Vector hold objects of the generic type Object, but they are unable to hold primitives like double. While we store double in a vector we have to convert double to Double. While retrieving an object we need to explicitly cast it to desired type as it cannot remember its type.

Container objects in Python like lists and dictionaries are able to hold objects of any type including numbers and lists. It doesn't need casting while retrieving as it remembers its type.

13. File I/O :

Another reason why Python is simpler than Java : take a look at the following code.

 //Java :   File dir = new File("."); // get current directory File fin = new File(dir.getCanonicalPath() + File.separator + "Code.txt"); FileInputStream fis = new FileInputStream(fin);   //Construct the BufferedReader object BufferedReader in = new BufferedReader(new InputStreamReader(fis)); String str = null; while ((str = in.readLine()) != null) {   // //Process each line, here we count empty lines if (str.trim().length() == 0) { } }   // always close the buffer reader in.close();

 #Python : demoFile = open("/home/User/Desktop/demo.txt")   print demoFile.read();

Basically in Java just to read a file we need to import a lot of classes and due to this code becomes lengthy. On the other hand in Python it is just two lines.

14. Class and Inheritance :

 //Java :   class Animal{ private String name; public Animal(String name){ this.name = name; } public void makeSound(){ System.out.println("I am " + name); } }   class Lion extends Animal{ public Lion(String name) { super(name); }   public void makeSound(){ System.out.println("I can roar"); } }   public class Main { public static void main(String[] args) { Lion lion = new lion("AfricanLion"); lion.makeSound();   } }

Python :   class Animal():   def __init__(self, name):   self.name = name   def makeSound(self):   print "I am " + self.name   class Lion(Animal):   def makeSound(self):   print "I am "+ self.name + ", and I can roar"   lion = Lion("AfricanLion") lion.makeSound()

When we extend a base class in java we have to define an explicit constructor for implicit super constructor. But in Python there is no such need.

15. Numbers :

In java we need to define data types while declaring variable but it is not the case for python after initialization we need to do it explicitly.

 //Java :   //integer   int num1 = 10;   //floating point numbers float num2 = 12.3f;   //double double num3 = 10.03;

Python :   #integer num1 = 100 num1 = int ("100")   #floating point number   num2 = 12.3 num2 = float("12.3")

Conclusion :

Python's future is looking, brilliant from where we see and accept that its future is guaranteed. Python is a long way from perfect, and the same can be said for the ecosystem around us. So yes, there are a lot of zones where Python should most likely will, progress. Be that as it may, we shouldn't dismiss the way that a large number of the issues with Python (like concurrency, binary distribution and dependancy management) are issues with software development by and large, so there's no place for individuals to go that will magically influence those issues to vanish.