

Java Notes

```
public class Hello { // the classic starter!

    public static void main(String[] args) {
        System.out.println("Hello World!");
    }

}
```

Primitive types and wrappers

Primitive	Object
int	Integer
long	Long
float	Float
double	Double
char	Char
boolean	Boolean

You can create a Scanner, like `new Scanner(System.in)`. Methods:

Return type	Method name	Description
boolean	hasNext()	returns true if more data is present
boolean	hasNextInt()	returns true if the next thing to read is an integer
boolean	hasNextFloat()	returns true if the next thing to read is a float
boolean	hasNextDouble()	returns true if the next thing to read is a double
Integer	nextInt()	returns the next thing to read as an integer
Float	nextFloat()	returns the next thing to read as a float
double	nextDouble()	returns the next thing to read as a double
String	next()	returns the next thing to read as a String
String	nextLine()	returns all before the next newline

String manipulation: Examples below use a string variables called `w` and `u`

Python	Java	Description
<code>123gPg456[3]</code>	<code>w.charAt(3)</code>	Return character in 3rd position
<code>w[2:4]</code>	<code>w.substring(2,4)</code>	Return substring from 2nd up to but not including 4th
<code>len(w)</code>	<code>w.length()</code>	Return the length of the string

Python	Java	Description
<code>w.find('x')</code>	<code>w.indexOf("x")</code>	Find the first occurrence of x
<code>w.split(',')</code>	<code>w.split(",")</code>	Split the string at ', ' into a list/array of strings
<code>w.split()</code>	<code>w.trim().split("\\s+")</code>	Split out non-whitespace strings
<code>w + u</code>	same as Python	Java: One of the operands does not have to be a string
<code>w.strip()</code>	<code>w.trim()</code>	Remove any whitespace at the beginning or end
<code>w.replace("me", "I")</code>	same as Python	Replace all occurrences of first parameter by second

List methods: Suppose w and w2 are lists of strings --

Python	Java	notes
value of <code>w[3]</code>	<code>w.get(3)</code>	Return item in 3rd position
<code>w[3] = 'a'</code>	<code>w.set(3, "a")</code>	Set item in 3rd position
<code>w.append('a')</code>	<code>w.add("a")</code>	Append item
<code>len(w)</code>	<code>w.size()</code>	Return the number of items in the list
<code>w.find('x')</code>	<code>w.indexOf("x")</code>	Find index of the first occurrence of x, or -1
<code>w += w2</code>	<code>w.addAll(w2)</code>	add all of list w2, modifying w
<code>'x' in w</code>	<code>w.contains("x")</code>	membership test
<code>not bool(w)</code>	<code>w.isEmpty()</code>	Python w is True if not empty
<code>w[:]=[]</code>	<code>w.clear()</code>	Remove all items
<code>w.pop(2)</code>	<code>w.remove(2)</code>	remove and return item at position 2
<code>w.sort()</code>	<code>Collections.sort(w)</code>	sort according to natural ordering

To separate with ", " the strings in list sList:

`','.join(sList)` in Python
`String.join(", ", sList)` in Java

Here is a function to return a new array containing the squares of the numbers in a passed array. See the use of the length attribute:

```
public static int[] squares(int[] nums)
{
    int n = nums.length;
    int[] sq = new int[n];
    for (int i = 0; i < n; i++) {
        sq[i] = nums[i]*nums[i];
    }
    return sq;
}
```

It could be called with:

```
int[] vals = {2, 5, 6, 22},
      vals2 = squares(vals);
```

Dict/Hashmap

Python	Java
<code>d = dict()</code>	<code>d = new HashMap<keyType, valueType>()</code>
<code>d[key] = v</code>	<code>d.set(key, v)</code>
<code>v = d[key]</code>	<code>v = d.get(key) // may be null</code>
<code>d.keys()</code>	<code>d.getKeys()</code>

Print formatted string with fieldwidth, float/double b with precision:

```
Python print('{:20} and {:.2f}'.format(a, b))
Java System.out.format("%20s and %7.2f%n", a, b);
```

Note `%n` is needed at the end of the Java format to go on to the next line.

Boolean Operators

Python	Java
<code>and</code>	<code>&&</code>
<code>or</code>	<code> </code>
<code>not</code>	<code>!</code>

Conditionals in Java

```
if (condition) {
    statement1
    statement2
    ...
} else {
    statement1
    statement2
    ...
}
```

Loops and Iteration

```
for (int i = 0; i < 10; i++) { // Java
    System.out.println(i);
}

for (start clause; stop clause; step clause) {
    statement1
    statement2
    ...
}
```

If `s` is a sequence (of element type `tp` in Java)

```
for e in s:           # Python
    print(e)

for (tp e : s) {     //Java
    System.out.println(e)
}

while (condition) {
    statement1
    statement2
    ...
}
```

Interfaces

```
public interface Response    // interface, not class
{
    boolean execute(String[] tokens); //public assumed
    String getCommandName();    // ; instead of body
    String help();
}
```

Heading for class using Response interface:

```
public class Goer implements Response
```