

1 . Advantage:

```
def fun(a, b):  
    return a+b  
  
result = fun(1, 2)  
  
result = fun(3.3, 4.7)  
  
result = fun( "aaa" , "bbb" )
```

Only one function need to be written, the function can calculate integer, float, string type and so on.

Disadvantage:

```
A[0] = 1  
  
A = 2  
  
A[1] = 3
```

The mistake above cannot be detected by the compiler, only be carried out at runtime, waste of time.

2. In SurvivalGame.java

a:

```
for (Object o : teleportObjects) {  
    if (o instanceof Player) {  
        Pos pos = ((Player) o).getPos();  
        if (pos.getX() == randx && pos.getY() == randy)  
            return true;  
    } else {  
        Pos pos = ((Obstacle) o).getPos();  
        if (pos.getX() == randx && pos.getY() == randy)  
            return true;  
    }  
}
```

b:

```
for (Object obj : teleportObjects) {  
    if (obj instanceof Human)  
        ((Human) obj).teleport();  
    else if (obj instanceof Chark)  
        ((Chark) obj).teleport();  
    else if (obj instanceof Obstacle)  
        ((Obstacle) obj).teleport();  
}
```

In survival\_game.py

```
for i in range(self.n + self.O):  
    pos = self.teleportObjects[i].getPos()  
    if pos.getX() == randx and pos.getY() == randy:  
        return True
```

```
for i in range(self.n + self.O):  
    self.teleportObjects[i].teleport()
```

in java code, when object call its function, it need to specify which class it belongs to, but python needn't, so the python code is more concise than java

3.python:

```
for i in range(self.n/2):
```

```
    self.teleportObjects[i] = Human.Human(0,0,i,self)
```

```
    self.teleportObjects[i+self.n/2] = Chark.Chark(0,0,i,self)
```

```
# last one get Wand
```

```
self.teleportObjects[(self.n/2) - 1].equipment = Wand.Wand(self.teleportObjects[(self.n/2) - 1])
```

```
self.teleportObjects[self.n - 1].equipment = Wand.Wand(self.teleportObjects[self.n - 1])
```

java:

```
for (int i = 0; i < (n/2 - 1); i++) {
```

```
    teleportObjects[i] = new Human(0, 0, i, this, false);
```

```
    teleportObjects[i + n / 2] = new Chark(0, 0, i, this, false);
```

```
}
```

```
teleportObjects[n/2 - 1] = new Human(0, 0, n/2 - 1, this, true);
```

```
teleportObjects[n-1] = new Chark(0, 0, n/2 - 1, this, true);
```

in python, I can directly let the last player on each race get the wand

but in java I need to add a parameter to control weather the player will get the wand