Greenfoot Syntax Guidance

1. Setting up the World
Double click on the world on the right to open the code editor, the code required to set up the world in the correct proportions is

```
super(wide, tall, 60);
```

Where **wide** is the number of cells wide, and **tall** is the number of cells tall

2. Generating a Random Number
Adding this code to the act() method of any actor will allow you to generate a random number

```
int variableName = Greenfoot.getRandomNumber(numberOfNumbers);
```

Where **numberOfNumbers** is one greater than the highest number value you want to return

3. If statements
The following code demonstrates an if statement

```
if (condition) {

    //your code here

} else if (condition2) {

    //more code here

} else {

    //your else code here

}
```

Where **condition** and **condition2** are the questioning conditions used to select the section, the code is executed inside the curly braces where the conditions match

4. Rotating an Actor
First identify the angle you want to rotate the actor in, this will be an integer value between 0 and 360, then use the following code

```
setRotation(angle);
```

Where **angle** is the clockwise angle to rotate the object, remember in Greenfoot the 0° angle is pointing **right**.

5. Moving an Actor
There are two ways of moving an actor, the easy way requires you to be rotated in the correct direction first, then

```
move( cells );
```

Will move the actor the amount of **cells** specified in the direction they are facing.

To move an actor regardless of the direction they're facing you need to identify the offset you need to either the X or Y axis that will result in it moving in the direction you want it to

```
setLocation( getX(), getY() );
```

Where **getX()** would need to become **`getX()-1`** or **`getX()+1`**, and **getY()** would become **`getY()-1`** or **`getY()+1`** as needed

6. Checking to see if a key is pressed
The condition required to identify a particular keypress is below

```
Greenfoot.isKeyDown("right")
```

Where **"right"** is the name of the right-arrow key, change this for the correct key.

7: Collision Detection
Collision detection code goes into the act() method of the actor that is **not** disappearing

```
Actor actorName;
actorName = getOneObjectAtOffset(0,0,actorName.class);
```

```
if(actorName!=null){
    WorldClass worldName = (WorldClass)getWorld();
    worldName.removeObject(actorName);
}
```

Replace **actorName** with the name of the object that will disappear.
Replace **WorldClass** with the name of the background object.
Replace **worldName** with the lowercase name of the background object.

8: Playing a sound

```
Greenfoot.playSound("pop.wav");
```

This will play the sound file **pop.wav**

9: Scoring with Counter

Add a counter to the world using the **Right click > New Counter()**, then make sure to **save the world.** Go and find this line of code in the world subclass.

```
Counter thisCounter= new Counter();
```

Move it underneath the main method, this will normally be outside the curly braces that come after **public Sky** where **Sky** is the name of the background object. You'll then need to add the following right below it.

```
public Counter getCounter(){
     return counter;
}
```

Click **Compile**.

Double click on the character on the right *that will not disappear* and add this to the **act()** method inside the curly braces { } of the if statement you used for collision detection.

```
 Counter counter = worldName.getCounter();
 counter.bumpCount(value);
```

Remember to set **value** as an **integer value** of how much the score will change. Positive numbers increase the score, negative numbers decrease the score.

10: Respawning

If an object you've removed needs to reappear in the world then one more line of code is needed

```
worldName.addObject(actorName, Greenfoot.getRandomNumber(wide),
Greenfoot.getRandomNumber(tall));
```

This will make the **actorName** object reappear in the world at a random location based on the width and height of the canvas. You will need to replace **worldName, actorName, wide** and **tall** with your own values.