### Transition

**Source:** String  
*Read:* Character  
*Destination:* String  
*Write:* Character  
*Move:* Direction

**Transition (in source state: String,**  
*in read character: Character,*  
*in destination state: String,*  
*in write character: Character,*  
*in move direction: Direction*)

**Source state:** String  
*Read character:* Character  
*Destination state:* String  
*Write character:* Character  
*Move direction:* Direction
for each line in file:

```plaintext
for each line in file:

    line = line.split('#') [0] // removes comments

    if (line[0] == 'E'):
        if (TempTarget.Complexed):
            targetList.append(TempTarget)
        TempTarget = new Target
    else:
        TempTarget = new Target
    "otherwise, split on "

    if (str(line.split('=', 2))
        if (strSplit[0]) is in FuncDict.Keys():
            FuncDict[strSplit[0]](strSplit[1])
```

```

    line = line.split('#') [0] // removes comments

    if (line[0] == 'E'):
        if (TempTarget.Complexed):
            targetList.append(TempTarget)
        TempTarget = new Target
    else:
        TempTarget = new Target

    "otherwise, split on "

    if (str(line.split('=', 2))
        if (strSplit[0]) is in FuncDict.Keys():
            FuncDict[strSplit[0]](strSplit[1])
```

```
func Dict = {}

"x" : StoreX,
"y" : StoreY,
"z" : StoreZ,
"id" : StoreId,
"name" : StoreName,
"points" : StorePoints,
"flashRate" : setFlash

Each storage method has a prototype of the form:

    SetFunc (string input, Target inTarget) =>

Checks:
- Check for complete [Target] headers
- Check all attributes for a target are set

% StoreX ('3.222', tempTarget) % Some Comment%

`x = 3.222` % Some Comment%

`x = 3.222`

["x", "3.222"]