In Python, assignment operator = assigns value to Variable from right to left. For example,

```
name = “Sarah”
another_name = name
```

Here, it assigns the string “Sarah” to a Variable and then assigns the same string to another Variable. Which is the same as,

```
<name><Sarah> is pointer-none.
<another_name>{name}
```

Here, Variables don’t need quotes to represent a string. Instead, the string remains floated!

While learning Python, we have undoubtedly encountered a one-line program utilizing the print() function for output:

```
print(“Hello, world!”)
```

Unlike Python, in Saalang, inOperator and outOperator are just part of Variable syntax instead of being a Function.

So, <out><Hello\, world!> does the same!

```
if name == “Sarah”:
    # Go Crazy!!
```

Which in Saalang is,

```
if name = “Sarah”:
   ..remove h<out>{name}

$ Sara
```

```
try:
    input = int(in(“Type a number: “))

except:
    Bug!!


In Saalang,

loop:
    : <input>{in} Type a number\: {}
    then <input>{in} {}

    : try:
        int(input)
        collect(“f”)

   except FIE:
        next; # it goes to next loop.

finally:
    <any(dict)><num: ‘input’> is pointer-any.

<out>{dict}

$ 10.0
$ <num: (10)>


A member of a dictionary can have many key-value pairs, as
long as each key is unique.


<name.list><name: John  wife: Yara> is pointer-list.


<temp.list><name: Sarah  son: John>
.insert (1), (*temp.list)<name.list> is main exhaust.

<names.list>{name.list}  del(name.list) is renaming.
```

```
<out>{names.list}

$ <name: Sarah  son: John,
name: John  wife: Yara>

<out>{name.list} and <out>{temp.list} now generate
NameError..


loop:
    : for name.list in names.list

    : loop:
        : for name in name.list AND key in keys
        : <out><‘name’\’s /db>
        then <out><‘key’ is ‘name’.>

$ Sarah’s son is John.
John’s wife is Yara.
```

String syntax also supports function(s):

```
<out><Total items\: ‘item’ ‘sp(item, “item only”,
“items”)’/b Cart total\: ‘easy_read(get_int(cart, 3)-1, 3,
2)’>
```

Here, sp() gives singular or plural. get_int() gives “discount” of upto 1000 rupees! easy_read() combines the last three digits and then groups each two digits together to enhance readability.

```
For <item>(1), <cart>(23890),
$ Total items: 1 item only
$ Cart total: 22,999

Now, <item> += 1, <cart> += 82050 ~ <cart>(105940),
$ Total items: 2 items
$ Cart total: 1,04,999
```

Suppose 4 players are playing a game in a loop. Set is useful here:

```
<players.list><a : b, b : c, c : d, d : a>
    // is_restricted()
```

Here, a gives you b. b doesn’t give you a, instead gives you c.

```
loop:
    ; <player><a> # player a’s turn.
    : ..fetch (player)<player>{players.list}
    # first loop is for player b, second for player c, ..
```

Now, suppose your program handles four games with two players each,

```
<players.list><a : k, b : l, c : m, d : n>
```
Since it is not restricted to main side, k also gives you a.