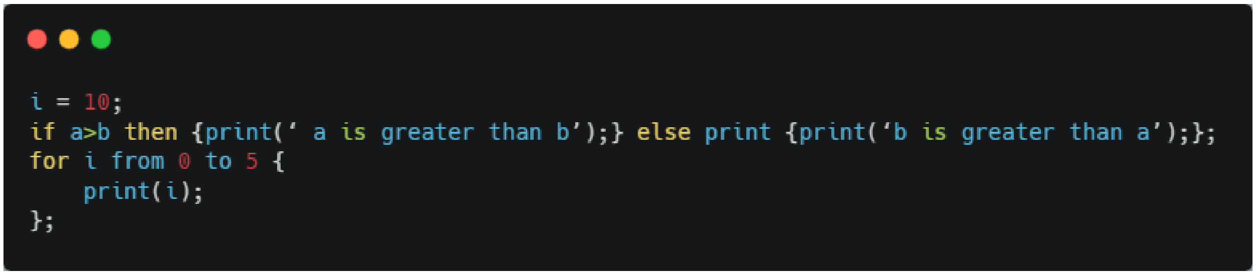
lirix

DOCUMENTATION

General Syntaxes

- The delimiter is ';', therefore use a semicolon to end each statement. Also, if-else statements and loops must end with a semicolon.

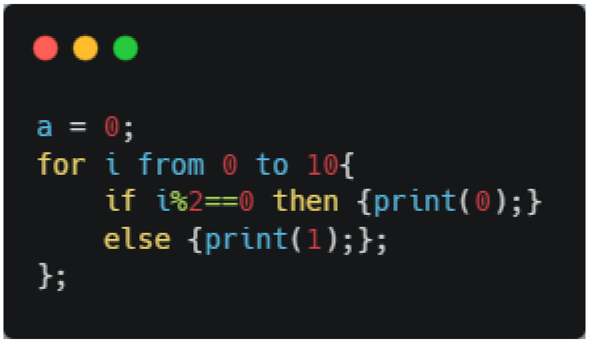
  Ex:



```
i = 10;
if a>b then {print(‘ a is greater than b’);} else print {print(‘b is greater than a’);};
for i from 0 to 5 {
    print(i);
};
```

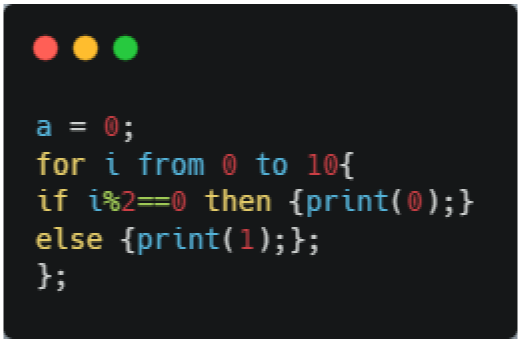
- The indentation does not matter in our language.

  Ex:



```
a = 0;
for i from 0 to 10{
    if i%2==0 then {print(0);}
    else {print(1);};
};
```

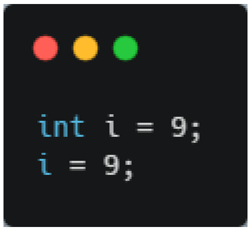
The above code is same as



```
a = 0;
for i from 0 to 10{
if i%2==0 then {print(0);}
else {print(1);};
};
```

- The variables are not to be initialized, straight away the names are to be mentioned.

  Ex:

  ```
  int i = 9;
  i = 9;
  ```

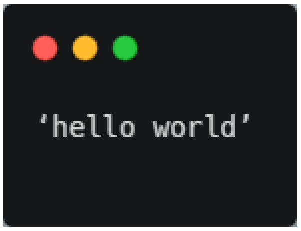
  The second is the correct method and not the first.

Features

1] Strings:

Strings in our language are surrounded by single quotation marks (“).

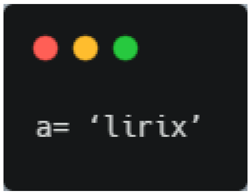
Ex:



```
‘hello world’
```

Assigning a string to a variable is done with the variable name followed by an equal sign and the string.

Ex:

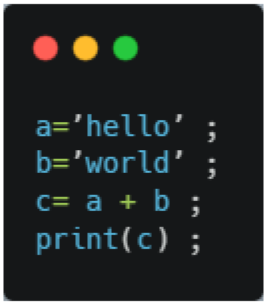


```
a= ‘lirix’
```

String Concatenation:

To concatenate or combine two strings you can use the + operator.

Ex: To merge variable **a** with variable **b** into variable **c:**



```
a=’hello’ ;
b=’world’ ;
c= a + b ;
print(c) ;
```

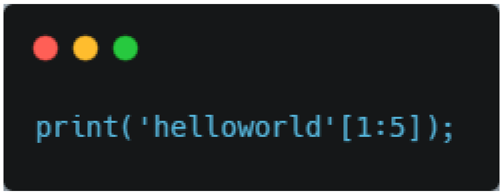
String Slicing:

You can return a range of characters by using the slice syntax.

Specify the start index and the end index, separated by a colon, to return a part of the string.

NOTE: The first character has index 0.

Ex: To get the characters from position 1 to 5 (not included)



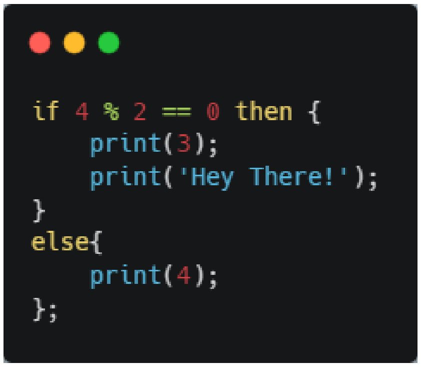
```
print('helloworld'[1:5]);
```

2] **If-Else**

The syntax of the if-else statement is if (condition) then {statements} else {statements};

Multiple statements can be added in the then and else parts separated by ‘;’.

Ex: to print 3 and ‘Hey There!’ if 4 is divisible by 2 or else print 4

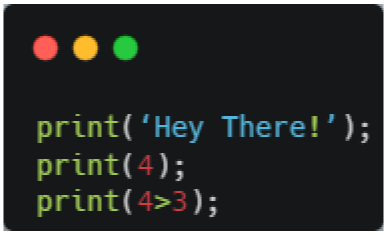


```
if 4 % 2 == 0 then {
    print(3);
    print('Hey There!');
}
else{
    print(4);
};
```

3] Print

The print statement is called by writing print(content). If the content in the parenthesis is a string, it outputs the exact same string, if it is a number, it outputs the number and if it is some boolean value, it outputs it.

Ex: Print various items



```
print(‘Hey There!’);
print(4);
print(4>3);
```

The above code prints:

Hey There!

4

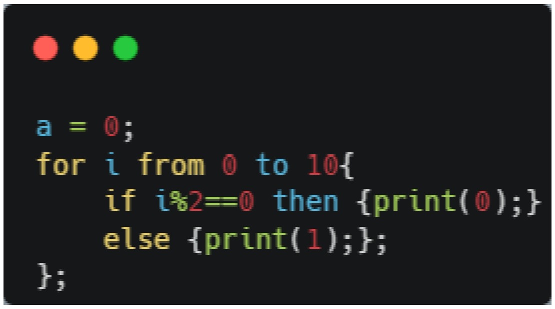
true

4] For Loop

The syntax of the for loop in *lirix*: for (range) {statements};

The statements in braces can be more than 1 and are separated by ‘;’.

Ex: for numbers from 0 to 10 (included) print 0 if the number is even number and 1 if it is not.



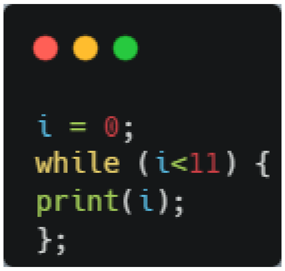
```
a = 0;
for i from 0 to 10{
    if i%2==0 then {print(0);}
    else {print(1);};
};
```

5] While Loop

The syntax of while loop is: while (condition) {statements};

There can be multiple statements in braces and should be separated by ‘;’

Ex: to print all numbers from 0 to 10

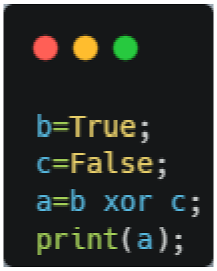


```
i = 0;
while (i<11) {
print(i);
};
```

6] **Logic Gates**

The syntax of logic gates is: var1 (name of gate) var2

Ex: AND of two variables



```
b=True;
c=False;
a=b xor c;
print(a);
```

8] **Mutable Arrays**

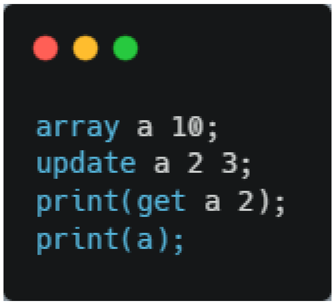
NOTE: The indexing is initialized from 1 (not 0)

The various syntaxes of arrays are:

- Initialization - array (name of the array) (size of the array)
- Updation - update (name of the array) (index at which the element has to be changed) (the new value)
- Fetch - get (name of the array) (index of the element which is to be fetched)

Initially, when an array is defined, all elements are None. They can be changed later on as per needs using *update*

Ex:



```
array a 10;
update a 2 3;
print(get a 2);
print(a);
```

An array of size 10 and name ‘a’ is initialized. The 2nd element is updated to 3 (None -> 3). In the third line, the second element (=3) is printed.