

DOCUMENTATION OF EASIA :

1) **Blocks**

Each EASIA program is made of blocks of statement which are at their turn divided into logical lines.

2) **Line Structure**

We have two kinds of line : logical line and physical line

Physical line:

A physical line is a sequence of characters terminated by an end of line sequence.

A physical line ends in whatever the current platform's convention is for terminating lines.

Logical line:

It starts at the beginning of the statement and its end is represented by the Token NEWLINE. A logical line is constructed from one or more physical line by following the implicit or explicit line joining. Logical line is what the language sees as a single line where there is actually many physical lines.

Example 1:

```
s = 'This is a string. \
```

```
This continues the string.'
```

```
Print s
```

Output:

```
This is a string. This continues the string.
```

Example 2:

```
✓ My_list = [1, 2, 3, 4] here we have one logical and physical line
```

```
✓ My_list = [1, 2,
```

```
3, 4] here we have one logical line on two physical lines
```

3) **Comments:**

- Single line of comment in EASIA will start with the # character and extends to the end of the physical line. A comment may appear at the start of a line or following whitespace or code but not within a literal string.

Example:

```
# This is a single line of comment
```

DOCUMENTATION OF EASIA :

- If it is multiple line of comments we will be using << ...>>. Character with the comments embedded inside .In other words the << start the comments from the first line and the >>. will close it when the last line of comments is reach.

Example:

```
<< This is how we represent  
Multiple lines of comment  
In EASIA >>.
```

4) Different types of tokens

The different available tokens are Newline, Indent, Dedent, identifier,Keyword,Literals, Operators and Delimiters.

5) Keywords and identifiers

- Keywords in Easia

as	is	bring	
when	dif	break	
other	fun	return	
for	or	del	
Case	not	do	
true	in	long	
false	From	and	show

6)Operations Rules

An operation is made of operand and sign operand as follows:

a+b

We can not subtract or divide something else than a numerical type by an numerical type

7)Arithmetic operations

A precedence have been decided in function of the type. Indeed if an operation is made of two different numerical types the first type will give it's type to the result of the expression . Hence it can use for casting implicitly some values. This is possible for all the arithmetical operations except the modulo one

-Ex: a is an integer and b is a float so

a*b=c which is an int

b*a=c which is a float

DOCUMENTATION OF EASIA :

8) Operation on strings:

Operations	Symbol	Examples
Concatenation	+	"my"+"name"="my name"
Repetition	*	"my "*3="my my my "
Subtract	-	"elementary"-e="lmntary"

9) loops and conditional statements

-already implemented statements

The when other statement:

This statement is structured as follows:

Structure :

```
when(boolean):  
    operations  
other:  
    operations
```

end of structure.

Rule1: If there is no other cases no need to put other.

The loops:

The for loop:

The **for** loop is structured as followed:

Structure :

```
for (variable) going from (value) to (value) by {(value)}*:  
    operations
```

end of structure. *The step is optional an the default step is 1

The as long as loop:

The **while** loop is structured as follow

Structure :

```
as long as (boolean):  
    operations
```

end of structure.

DOCUMENTATION OF EASIA :

-Future statements to be implemented

The do ...while statement

Structure:

do:

 expression

 as long as (boolean)

End of structure

The break statement:

This statement is suppose to help the user to get out of the loop no matter which loop it is .

Error raising

Syntax:

 Error("error")

end

Error handling

Syntax

try:

 risky expression

 risky expression;

catch(name of the error in one word):

 action to face the error

end