**ALIENS PROGRAMMING LANGUAGE**

Documentation

Version 1.0

Friday, August 28, 2020

//Content of documentation

1. Introduction:

we have made this language to make programming as easy as possible for new learners, who have no experience in coding.

we coined this language as ‘Aliens ' programming language and the abbreviation is "Als” which we will use a lot in our programming journey.

we also designed this language to be familiar to the user especially those who already have an experience in coding, we got inspired the structure and the syntax from the famous and successful languages such as (dart, python, C#, JavaScript ...), so it inherits many of the same statements and expressions form those.

* 1. The reason why we name our language "Aliens Programming Language":

before talking extensively about the syntax and the structure of the language we should put you in the ground and reasons behind choosing this name for our language :

On 10 August 2020, our team participated in one of the biggest events that Repit organized with a prize of 10000 $ for the winning team.

our team started the brainstorming on how our language should look like after hours of negotiation and discussion, we finally agreed that the language should present something that we all know, even those how never used a computer before .the idea is space.

you must be wondering what is the relevance of "space" with computer language, alright we will explain.

the first thing you should know is the Aliens language hierarchy, well.

imagine yourself you are cruising space; in your way you may discover new planets or new galaxies. how knows everything is possible over there, so the thing is our language converts all this word or space jargon if we could say to names that have a meaning in our language.

eg: a user could create a planet with its moons and of course inside a space which contains many galaxies, it is simple isn't it?

we think now it's become more obvious for you where the name of "Aliens programming language” came from.

1. Pre-requirements:

all you need to make this language work is <https://www.python.org/>

, and Linux.

“unfortunately, this language only works in Linux for the moment. “

1. Create new project:

now let create our first project which we will name as test :

>>>> #this is how we create our project

>>>> als createProject Test

>>>>cd Test

>>>>cat main.als

Space(){

Show(‘hello world’)

}

>>>>

As you see in the example above, we created our “Test” project successfully, inside Test directory we have the file main.als as well. you noticed the keyword space. this is our entry point which will be responsible for displaying "hello world".

1. Variables in Aliens

variables are important to store your information temporarily in the computer ram in order to use them again, there is a lot of types :

* 1. Numbers

we use this type of variables when we want to represent both integer and floating-point numbers.

a=12+20

show(‘the result of this addition is’+a)

#the result will be something like this:

-------------------------------------------------------------------------------------------------------------------------------------------

The result of the addition is 32

As you see in the example above we declare a variable which we called a to get the addition result of two integers, after that we used show function to display the result to the user.

**Note:**

We use **#**  this symbol to say this is only comment .

* 1. Strings

we use this type of variables when we want to represent any kind of text, but it must be surrounded by quotes. Or double quotes.

Myname=’Adam’

Show(‘my name is ‘+Myname)

#the result will be something like this.

-------------------------------------------------------------------------------------------------------------------------------------------

My name is Adam

What we did here is no difference from what we did in the example above of number so we declare a variable of type string, after that we display it .

* 1. Lists:

Countries=[‘morocco’,‘usa’,’canada’,’France’]

#this is a list of countries

1. Conditions

>>>> here copy a terminal exemple

1. Loop Statements
   1. @loop(start=0,end) as n{...}

>>>> here copy a terminal exemple

* 1. @while(condition){..}

>>>> here copy a terminal exemple

* 1. Enumerable.loop(item){...}

>>>> here copy a terminal exemple

* 1. enumerable.loop(e,item){..}

>>>> here copy a terminal exemple

1. Defining Functions

>>>> here copy a terminal exemple

1. How to import a library?
2. Standard library
   1. Type Checking Functions

|  |  |  |
| --- | --- | --- |
| Function | Mining | example |
| $isnumber | return true if value is number | num= 10  test = $isnumber(num)  show(test)  >>>1 |
| $isalpha | return true If all characters in the string are alphabet. | Text= “Aliens”  test = $isalpha(Text)  show(test)  >>>1 |
| $isnumber | return true if value is number | num= 10  test = $isnumber(num)  show(test)  >>>1 |
| $isequal | return true if two values are equals | Text1= “mars”  Text2= “earth”  test = $isequal (Text1, Text2)  show(test)  >>>0 |
| $ismatch | return true if value respect regex expression | Text1= “Aliens”  test = $ismatch(text, “[A-z]\*”)  show(test)  >>>1 |

* 1. Strings Functions

|  |  |  |
| --- | --- | --- |
| Function | Mining | example |
| [str].lower() | return text to lowercase | Text= “ALIENS”  newText = Text.lower()  show(newText)  >>>aliens |
| [str].upper() | return text to uppercase | Text= “Aliens”  newText = Text.upper()  show(newText)  >>>ALIENS |
| [str].split(character/text) | return splitting text by character or text | Text= “Aliens Language”  newText = Text.split(“ ”)  show(newText[0])  >>> Aliens |
| [str].len() | return length of text | Text= “Aliens”  length = Text.len()  show(length)  >>>6 |
| [str].count(value) | return number of times the text is present | Text= “Aliens Language”  times = Text.count (“e”)  show(times)  >>> 2 |

* 1. List basic Functions

|  |  |  |
| --- | --- | --- |
| Function | Mining | example |
| [list].add(object) | add object to list | num= 10  test = $isnumber(num)  show(test)  >>>1 |
| [list].insert(object,index) | insert object into list | Text= “Aliens”  test = $isalpha(Text)  show(test)  >>>1 |
| [list].remove(object) | remove object from list | num= 10  test = $isnumber(num)  show(test)  >>>1 |
| [list].reverse() | return list reverse | Text1= “mars”  Text2= “earth”  test = $isequal (Text1, Text2)  show(test)  >>>0 |
| [list].size() | return size of list | Text1= “Aliens”  test = $ismatch(text, “[A-z]\*”)  show(test)  >>>1 |
| [list].clear() | Delete all element of list |  |

Chapiter 5: Introduction

Chapiter 6: Introduction

Chapiter 7: Introduction