**THE ALIENS PROGRAMMING LANGUAGE**

Documentation

Version 1.0

Friday, August 28, 2020

//Content of documentation

1. Invoking the Interpreter
2. The Interpreter and Its Environment
3. Variables in Aliens
   1. Numbers

>>>> here copy a terminal exemple

* 1. Strings

>>>> here copy a terminal exemple

* 1. Lists

>>>> here copy a terminal exemple

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. Built in functions
   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 |
| $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= “The Aliens Language”  newText = Text.split(“ ”)  show(newText[1])  >>>The Aliens |
| [str].replace(text1,text2) | replaces a given text within the text | Text= “The PHP Language”  newText = Text. replace(“PHP”,“Aliens”)  show(newText)  >>> The Aliens Language |
| [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= “The Aliens Language”  times = Text.count (“e”)  show(times)  >>> 3 |

* 1. List basic Functions

|  |  |  |
| --- | --- | --- |
| Function | Mining | example |
| [list].add(object) | add object to list | listPlanet= [“Earth”, “Mars”,“Neptune”]  newList = listPlanet.add(“Pluto”)  newList.loop(p) {show(p+ “ - ”)}  >>>Earth – Mars – Neptune – Pluto – |
| [list].insert(object,index) | insert object into list | listPlanet= [“Earth”, “Mars”, “Neptune”]  newList = listPlanet.insert(“Pluto”,1)  newList.loop(p) {show(p+ “ - ”)}  >>>Earth – Pluto – Mars – Neptune – |
| [list].remove(object) | remove object from list | listPlanet= [“Earth”, “Mars”, “Neptune”]  newList = listPlanet.remove(“Mars”)  newList.loop(p) {show(p+ “ - ”)}  >>>Earth – Neptune – |
| [list].sort() | return list sorted ascending | listPlanet= [“Pluto”, “Mars”, “Neptune”]  newList = listPlanet.sort()  newList.loop(p) {show(p+ “ - ”)}  >>>Mars – Neptune – Pluto – |
| [list].reverse() | return list reverse | listPlanet= [“Earth”, “Mars”, “Neptune”]  newList = listPlanet.reverse()  newList.loop(p) {show(p+ “ - ”)}  >>>Neptune – Mars – Earth – |
| [list].size() | return size of list | listPlanet= [“Earth”, “Mars”, “Neptune”]  size = listPlanet.size()  show(size)  >>>3 |
| [list].clear() | Delete all element of list | listPlanet= [“Earth”, “Mars”, “Neptune”]  newList = listPlanet.clear()  size = newList.size()  show(size)  >>>0 |

* 1. Global Use functions

|  |  |  |
| --- | --- | --- |
| Function | Mining | example |
| $system(command) | Execute the command (a string) in a subshell. | command = “date”  $system(command)  >>> Sat Aug 29 16:44:18 UTC 2020 |

1. Built in galaxies

|  |  |  |
| --- | --- | --- |
| Function | Mining | example |
| $tonumber | convert value to number | Text= “2019”  number = $tonumber(Text)  show(number + 1)  >>>2020 |
| $totext | convert value to text(string) | Number= 5432  Text = $totext(Number)  show(Text + “1”)  >>>54321 |
| $toxml | convert list to xml | Text= “The Aliens Language”  newText = Text.split(“ ”)  show(newText[1])  >>>The Aliens |
| $tojson | convert list to json | Text= “The PHP Language”  newText = Text. replace(“PHP”,“Aliens”)  show(newText)  >>> The Aliens Language |
| $todict | return length of text | Text= “Aliens”  length = Text.len()  show(length)  >>>6 |
| $toascii | return number of times the text is present | Text= “The Aliens Language”  times = Text.count (“e”)  show(times)  >>> 3 |

Chapiter 5: Introduction

Chapiter 6: Introduction

Chapiter 7: Introduction