AZ Progarmming Language

I/O Functions

Output Function:

Show(prompt)

Syntax:

```
Show(“hi”)
```

Output:

```
>>> ‘hi’
```

Input Function:

Input(prompt)

Syntax:

```
Input(“What’s your name:”)
```

Output:

```
>>> What’s your name:        _
```

Syntax Rules

- Each expression in the code should be containing ; to separate it from an another expression.
- While assigning variables, first we have to tell whether it a dynamic or static type var, then name it(it’s name sould obey var name rules) & then if necessary then its datatype too.
- A block of code should in between { }(curvy-brackets) to keep it separate from main code. It can be used for class & function code defining.
- The syntax should be like if we remove all white spaces & \n(newline) it should be working.
- For importing modules we use load.
- It does nothing like aliasing so if we change a var value an another will not change.

Variable Assigning

Format:

(var/const) (<name>) (<datatype>) = (<value>)

Syntax:

```
Var a int = 2;
```

```
Var b string = “3”;
```

```
Const b string = “non-changing string”;
```

Variable naming rules

01. Its name always starts with a letter(capital or small).
02. Its name can’t contain numbers at the first.

03. Its name can’t contain special character.
04. Its name should not be pre-defined term.
05. Its name should not be pre-defined var name.

Comments

- You can use “//” for mentioning comment(single line).
- You can’t use multi-line comment.

Syntax

```
int i = 10;
int j = 9;
// This is a comment!
Show(i+j);
```

Data-types

int(numberic)

Ex: 1,2,3,4,5,6,7,8,9,0

float(decimal)

Ex: 1.4, 34.4, 4.3, 45.2, etc.

string(character)

Ex: “I love she”, “I like it”, “Sum”, etc.

Byte(character)

Ex: b“His name is John\nHe loves that”, etc.

boolean(bool)

Ex: True,False

Data-Structures

Array(vector)

Ex: [1,2,4,3,3], [“hsd”, ”ee”, ”sd,], etc.

keymap(hash-map)

Ex: {

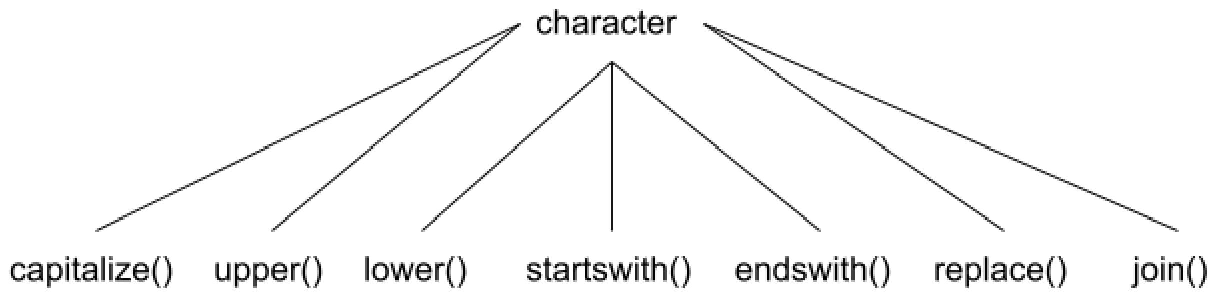
“Hi”: 1

“hello”:2

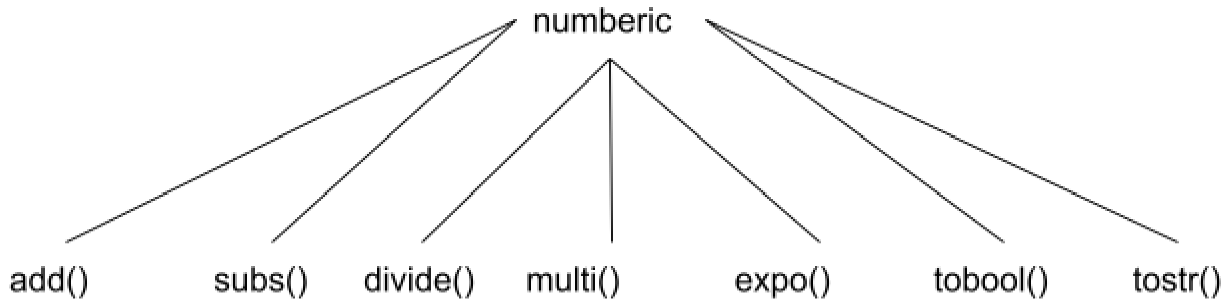
}, etc.

Pre-Defined Object Functions

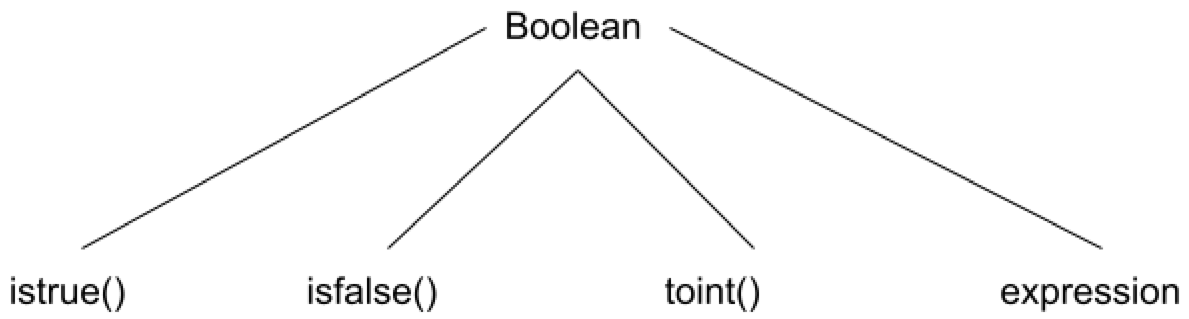
character functions map



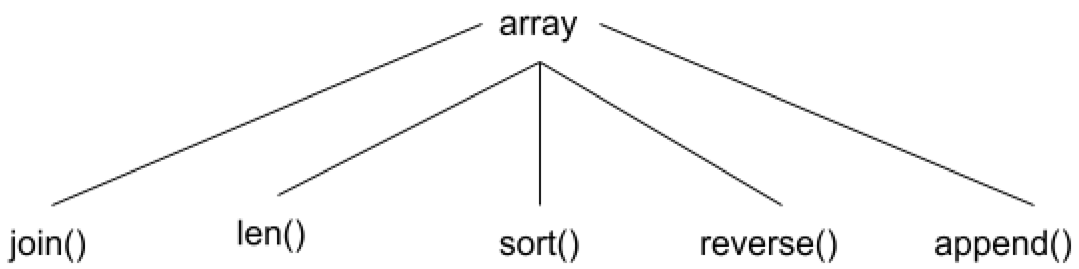
int functions map



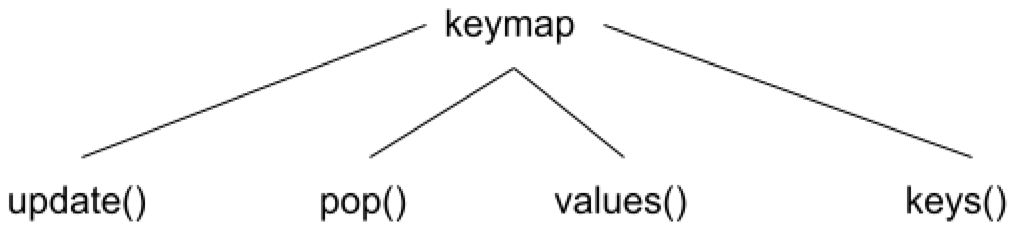
boolean functions map



Array functions map



Keymap functions map



Function defining

```
fun <name>(<argu>,<argu>) => <returnDataType>{
# Code #
};
```

Loops

while loop

- Keyword is while.
- It’s for satisfy a condition.

```
while (condition){
#  Code  #
};
```

For loop

- Keyword is for.
- It used for iteration.

```
for(i (in/of) iterable){
#  Code  #
};
```

Classes & Custom Objects

- Classes can be used with the keyword class.
- Classes has a function __init__() which can be used to run an initial code.
- Classes has a function __call__() which can be used to define what to give when directly called.
- In classes, from can be used to define where its inheriting from an another class.
- Private functions should contain ” _” before its name.

Syntax

```
class main from parentMain{
fun __init__(){
# Code #
};
fun __call__(){
# Code #
};
fun func(parameters){ # Code # } };
```