Quack.OS 1.0.0

ReadMe Manual

Developed by Pranav Saran

QUACK.OS MANUAL

“QUACK THE CODE”

PRINT STATEMENTS:

When writing a print statement for quack.os you would need to use the keyword: “quack” followed by a colon. The syntax would be as follows:

quack: text you want to print

When using a variable in the print statement, the syntax changes. You would offset the variable name with commas like the following.

quack: text you want, variable name, more text

This would successfully print out any variable within a string. If the variable ends the string you would still offset it with commas like the following.

quack: text you want, variable name,

INTS AND STRINGS:

When initializing variables in Quack.OS, you would follow a set formula:

1. Use “i” or “s” to indicate whether the variable type is an integer or string
2. Follow it with the keyword “var:”
3. Enter the variable name
4. Enter the variable value
    a. For ints you would just input the number
    b. For strings you would offset the name and value via a “.”

An example is provided:

ivar:a5 *creates variable a with value 5

svar:b. I like to play games *creates variable b that contains the pink text

USER INPUT AND OUTPUT:

In order to use inputs in Quack.OS, you would use the keyword: “(waddle)”. This would allow you to fill variable inputs as whatever you want. For example

ivar:a(waddle)

svar:b.(waddle)

*creates variable a but can be stored with whatever the user wants

You can then use the variables as regular variables in print statements.

MATH COMPUTATIONS:

Math computations are similar to regular programming except for the fact that a print statement is not required to store them. Math computations are also stand alone and can not be used in variables. #peak programming

Examples:

5+4=

5-4=

5*4=

20/4=

4%2=

These would output their results. Using them in If statements does work as stand alone numbers would work.

CONDITIONAL STATEMENTS:

These statements also work as normal with the keyword “if” followed by a “then” statement and then followed by the action you want to do. In order to use an if statement you will need 2 values and a comparator, these values can be variables or regular numbers or strings. You can not use > < operators with string functions. An example is presented below:

ivar:a5

ivar:b78

if b > a then quack: text you want to print

This would allow you to create conditional statements in Quack.OS

LOOPS:

Loops work the same way in Quack.OS except with a limitation that you can only loop one condition at a time. For example you can only follow the loop with one print statement. Loops start with the keyword “$loop” followed by the number of iterations “(x number)” followed by the keyword “do” and then the action. For example:

$loop(number of times) do action

LINE COMMENTS AND ERRORS:

Writing line comments in Quack.OS is very simple. By adding a hashtag before a line, you can write anything you want and have it serve as a line comment. For example:

#line comment can be written here

For error statements, the compiler will provide the error statement but will not stop the code entirely. This allows the user to figure out exactly what went wrong and where it went wrong. While everything else might be a step down from coding giants, this is one improvement feature you can get with Quack.OS.