A cartoon of a duckling

Description automatically generated

New Language: Puddle

Overview: This will be framed around the art of ‘rubber duck debugging’. It will be a generic language where the programmer can …

Brainstorming:

Waddle

Webbed feet

Flock

Scrooge mcduck

~~~~~~~~

Ugly duckling

<:>

,,,,,,

!

Demo Code:

~ This is a file that just prints items to the console.

mcduck var1Q << 10 !

mcduck var2Q << 25 !

mcduck var3Q << var1Q m var2Q !

daffy s1Q << ,,,Welcome to my first program,,, !

quack var3Q ,,,and,,, s1Q !

File Extension: .pud

Features:

[\* unique feature, or improvement from other languages]

-????

Syntax Rules:

|  |  |
| --- | --- |
| Single Line Comments | ~ This is a comment |
| Multi-line comments | ~~~  This is a multi-line comment.  ~~~ |
| Single Commands | Each command ends with an ! |
| Blocks of Code | Blocks of code begin and end with a duck, represented by the symbols <:> |
| Assignment Operator | << |
| Binary Operators | m = multiplication  d = division (round down)  a = addition  s = subtraction |
| Unary Operators | ugly = add 19123476230 to the number  flipper = flip the number from positive to negative, or vice versa  flock = determine how many full groups of 4 can be created from the number |
| Variables | All variable names must end with a Q, as if they are quacking at the screen. Valid characters are all alphanumeric, and ^ . |

**Variables Types:**

[You want these to relate directly to types in the base language, or at least for it to be possible to check if they are valid based on those types.]

mcduck = integers

daffy = strings

* All strings are enclosed around 3 commas, like this: ,,, Hello World ,,,

**Keywords:**

[print, loop, if-else, variable declaration, function declaration]

quack: everything after this command is printed to the console

duckquack: for up to 100 characters after this command, they are printed to the console as if coming from a speech bubble of a duck

\_\_ \_\_ \_\_ \_\_ \_\_ \_\_ \_\_ \_

| Hello World |

| \_\_ \_\_ \_\_ \_\_ \_\_ \_\_ |

|/

\_ \_ \_

>(.)\_\_ <(.)\_\_ =(.)\_\_

(\_\_\_/ (\_\_\_/ (\_\_\_/

waddle: the beginning of a loop, after this word will need to be a variable declaration, condition, and then stepper

Built-in Functions: