Intro to scripting in turtel

Syntax

Syntax of turtel is quite easy. Every line should consist of:

```
Keyword, SEPARATOR, options, ..., SEPARATOR

or

Variable Name, SEPARATOR, Variable Type,
 SEPARATOR, Variable Value, SEPARATOR
```

Every keyword must be separated with **SEPARATOR** (defined in `stdturtel.h`).

Data types

Turtel has 3 data types and no way of adding more.

- `num` - A basic number. Stores a value of a long long int.
- `str` - A basic string. Stores a string. Can be changed and turtel(.c) will manage the memory
- `tof` - A basic boolean. Stores 0 or 1. There is actually no need in using it. It won't improve speed or anything. It's just there.

Declaring variables

To declare a variable, you have to know its type and name.
Syntax: `Variable_name:data_type:value:`
Examples:

```
msg:str:Hello, World!:

# Will declare a string msg with value "Hello, World!".
```

```
x:num::

Will declare a number x with the value of 0.
```

This syntax allows 3 variables of the same name with different types.

Turtel has a builtin variable with type str:

- `__newline`:
  is literally a newline (\n)

Builtin functions

- print
  - syntax: `print:type:var:`
  - info: Prints variable ´`var`´ with type ´`type`´.
- read
  - syntax: `read:type:var:`
  - info: Reads user input to variable ´`var`´ with type ´`type`´.
- if
  - syntax: `if:num_1:operator:num_2:gototag1:gototag2:`
  - info: If (num_1 operator num_2) is true goto `gototag1,` else goto `gototag2.` Operator can be:
    - ´`eq`´ for equal
    - ´`lt`´ for a less than b
    - ´`gt`´ for a greater than b
    - ´`ne`´ for not equal
- srun
  - syntax: `srun:var:`
  - info: Runs ´`var`´ content with /bin/sh. ´`var`´ must be a str.
- exit
  - syntax: `exit:`
  - info: exits.
- goto
  - syntax: `goto:gototag_name:`
  - info: Jumps to ´`gototag_name`´.
- gototag
  - syntax: `gototag:gototag_name:`
  - info: Place to jump from `goto:gototag_name:`

- add, sub, mul, div, mod
  - syntax: `command:num1:num2:`
  - info: Runs arithmetic operations on ´`num1`´ op. ´`num2`´ and stores the result in num1.
- nowequ
  - syntax: `nowequ:type1:var1:type2:var2:`
  - info: Converts ´`var2`´ with type ´`type2`´ to ´`type1`´ and stores the result in ´`var1`´.

Full-line comments should start with a '#' symbol. Everything after the last SEPARATOR in the line is not interpreted, so it can also be used as a way to comment code.