



gotochan v1.0.14 documentation

getting started

welcome user, to everyones least favourite language. gotochan is an [esoteric](#) language so its designed to be as hard to read as possible!

things to know:

- gotochan uses flatcase (variables are typedlikethis)
- gotochan was made in c#
- gotochan converts your program to a list of commands before running

setup

to start coding in gotochan you will need to follow these steps:

1. open notepad
2. click "file" -> "save as"
3. set "save as type" to "all files"
4. enter file name as "my first program.gotochan"
5. navigate to saved file
6. right click on file -> "open with" -> "choose another app"
7. click "choose an app on your pc"
8. navigate to gotochan.exe
9. click "open" and select "always"
10. you can now edit your file by right clicking -> "open with" -> notepad, and run by double-clicking!

creating your first program

here is a simple hello world program in gotochan:

```
param = ~Hello~World!
```

```
goto say
```

lets break this down:

`param` is a variable. all built in methods use it as a parameter.

`~Hello~World!` is a string. strings in gotochan start with `~` and all spaces are marked by `~`.

`goto` is a keyword. it skips to the specified label or line number.

`say` is a label. `goto say` will skip to the line which defines the `say` label. as `say` is a built-in method, there is no need to define it.

data types

variables can have several data types in gotochan:

- string: `~this~is~a~string`
 - stores a sequence of characters
 - `\n` denotes a newline, `\h` denotes a hashtag
- number: 5, 10.2
 - stores a 64-bit number (double)
- bool: yes, no
 - stores a boolean value (true or false)
- null
 - no value

keywords

other languages can have hundreds of keywords. luckily, in gotochan there are only a few keywords you need to know.

- `goto`
 - skips to the specified label or line number.
 - examples:
 - `goto 1` – skips to line 1
 - `goto +2` – skips to the line 2 lines after this one
 - `goto sad` – skips to the line with the label called sad
 - an optional if statement can be used to only run the goto statement if a boolean variable is true.
 - examples:
 - `on = yes`
 - `goto +3 if on`
- `backto`

- skips to the line after the last line in which goto to the specified label was called.
- examples:
 - `backto somelabel`
- label
 - creates a label at this line
 - examples:
 - `label somelabel`

variables

all variables in gotochan are global. they can be used as such:

- `counter = 0`
- `index += 5`
- `number -= 2.3`
- `mech += ~railgun`
- `casualties *= explosions`
- `cookies /= people`

a variable can be set to the result of an equation with comparison operators.

- `iszero = myvariable == 0`
- `isnotzero = myvariable != 0`
- `isgreaterthanzero = myvariable > 0`
- `islessthanzero = myvariable < 0`
- `isgreaterthanorequaltozero = myvariable >= 0`
- `islessthanorequaltozero = myvariable <= 0`

if a variable doesnt exist, it returns null. set a variable to null to dispose of it and free memory.

comments

comments always start with a #. all text after the comment will be ignored.

note that an empty or commented out line still has a line number which will affect `goto` statements.

- `goto say # output to console`
- `# a comment`

built-in methods

you can create your own methods using `label labelname`. however, there are some built-in methods to allow you to access system apis.

you can use these methods using `goto labelname`. all built-in functions have a built-in `backto` statement.

these methods use the variable `param` for input and `result` for output.

- `say`
 - outputs the value in `param` to the console.
- `clear`
 - clears the console.
- `gettime`
 - sets the variable `result` to the amount of seconds since the epoch (1970).
- `wait`
 - waits for the amount of seconds specified in `param`.
- `input`
 - waits for the user to press a key and sets the variable `result` to the character of the key.
- `hasinput`
 - sets the variable `result` to whether a key pressed by the user is waiting to be processed. (multiple keys may be waiting.)
- `random`
 - sets the variable `result` to a random integer between 0 and `param` (inclusive).
- `error`
 - throws an exception at the current line with the message `param`.
- `gettype`
 - sets the variable `result` to a string containing the data type of the variable `param`.
- `truncate`
 - sets the variable `result` to the number `param` without any decimal places.
- `round`
 - sets the variable `result` to the number `param` rounded to the nearest whole number.
- `length`
 - converts `param` to a string and sets `result` to its length.

program examples

countdown:

```
counter = 10
param = ~countdown:~
param += counter
param += ~\n
goto say
counteriszero = counter == 0
param = 0.5
goto wait
counter -= 1
goto +2 if counteriszero
goto -9
param = ~blast~off!
goto say
```

note pad:

```
param = ~note~pad\npress~enter~to~quit\n\n
goto say
```

```
label inputloop
goto input
param = result
resultisenter = result == ~\n
goto +3 if resultisenter
goto say
```

goto inputloop

hope you enjoy coding in gotochan!

—Joyless