

In the name of the god of wisdom & life
that our knowledge may not pass in life

به نام خداوند جان و خرد
کزین برتر اندیشه برنگذرد

The guide to
M.i.o.p

My idea of programming as an interpreter

V0.1.2

By me

Hossein Mohammadi

Aka : @Shasquell_

راهنمای
M.i.o.p

ایده من از برنامه نویسی به عنوان یک interpreter

نسخه ی ۰/۱/۲

به وسیله ی من

حسین محمدی

در این راهنما به زبان انتخابی و موقعیت مورد نظر خود بروید:

Navigate to your language of choice and the position you desire in this guide:

English:

The goal of this program	3
Base syntax and system	4
Scripting	7
Last words	7

فارسی:

۸	هدف این برنامه
۹	نحو و سیستم پایه
۱۲	اسکرپت نویسی
۱۲	حرف آخر

First of all I'd like to apologize for my bad English grammar, after all English is not my first language and I totally understand someone getting mad over it.

The goal of this program:

I always wanted people to experience what I experienced in my head, well most of the stuff in my head could not be expressed easily, this one ... this idea of mine could !

Thus I began working on this project the purpose or better say 'use' of this program is that it can run commands given directly to the application or run a script file containing code for the program.

Well yes I know many people have done this already and with better results

but firstly they didn't start their project big, but expanded it and didn't abandon it in the long run.

secondly they were experts and had access to everything or at the very least the stuff they needed ,and also they learned how to code properly ,unlike me who barely knows how to code ,they were also mostly college students and some were high-schoolers.

No I am not belittling their work but simply saying that the reason I am bad at coding is that I lack experience and I made this project to gain experience.

The goal is to make this program give people the programming experience they want, but it is indeed far and requires help from anyone who can contribute. This is its goal and goals aren't always reached. but experience is always gained.

Base syntax and system:

The syntax of this program is quite simple and the help command explains it. Though I understand that to some it might seem vague and unclear ...

So in this part of the guide I will explain the syntax of some commands which this program provides you with. Also I will explain a little.

The first command I'd like to explain is the 'make' command. it is pretty basic but it does require some context . it is also a base command because it is the command which defines variables . if you don't know what variables are finish elementary school first and then come back.

Also please keep in mind that every command is in small characters and no character is capital.

The **make** command

The syntax of the make command:

-make [variable name]
=should be read: make [variable name]

It creates/makes a variable with the given variable name which actually can be any string which doesn't contain a space and yes you can make a variable which only contains numbers or like... literally remake the entire meaning of a number

-make [variable name] [variable content]
=make [variable name] containing [variable content]

It's the same as the first for the first parameter but the second parameter or anything written after the first parameter can be any kind of data but should not be the following:

Made of only space characters – will set the variable to nothing
Made of only dots – might cause crashes and problems

Also if you write another name of another variable it WILL NOT take the content of that variable and write them to this one.

The **delete** command

The syntax of the delete command is:

-delete [variable name]

It deletes a variable if it exists or it will write to the console that the variable didn't exist thus could not be deleted.

The **set** command

The syntax of the set command is:

-set [variable name] [content]
=set the content of [variable name] to [variable name or content]

The set commands first parameter is self-explanatory
but the second parameter can be the name of another variable which will obviously be changed to the content of that other variable.

The **add and sub** commands

Their syntax is:

-add/sub [numeric data] [variable name]
=add/sub [variable name or numeric data] to/from [variable name]

Adds the given number to the variable
/subtracts the given number from the variable

The **multiply, divide & remainder** commands

Their syntax is:

-multiply/divide/remain [variable name] [numeric data]
=multiply/divide [variable name] by [variable name or numeric data]

Please note that these all work the opposite way of the add/sub commands
and also remainder is 'remainder of division' or this '%' if you are familiar with it.

The **system** command

The syntax of the system command:

-system "string"

Runs a system command or literally passes the given string to the console entry.

-svstem [variable name]

Passes the given variable to the system.

The **out** command

The syntax of the out command:

-out "string"

Writes the given string to the console.

-outv [variable name]

=out variable [variable name]

This command takes a variable and writes it to the console.

The **in** command

The syntax of the in command:

-in [variable name]

Takes entry from the user and saves it to the given variable.

Please NOTE That these commands were added last minute (in version 0.1.2 the first build that I made public) thus they might contain some bugs!

The **insert** command

The syntax of the insert command is:

-insert [string data] [variable name] [at index]

=insert [variable name or string data] to [variable name] at [index]

The **remove** command

The syntax of the remove command is:

-remove [variable name] [from index] [amount to remove]

Please NOTE That it doesn't say 'to index', but instead says 'amount to remove'

Please NOTE That scripting is also something I just started messing around with thus it will have its problems!

Scripting:

When running a script using this program there are some things that change.

One of them is that you don't have to write dash in the script file.

Please use a '\ ' to disable a line or to ignore it.

The jump command

The syntax of the jump command:

-jump [line id]

Jumps to the specified line in the script file

Remember that the ID of the first line is ZERO.

The **jumpif** command

The syntax of the jumpif command:

-jumpif [data] [operator] [data] [line id]

=jump to [line id] if [data] [operator] [data]

If the if statement is true it will jump and if it's false it will not.

The **oute and outve** commands:

Their syntax is:

-oute "string"

Writes everything other than the command and the space after it to the console and adds an newline at the end of it.

-outve [variable name]

Writes the given variable to the console and adds an newline at the end of it.

Last words:

If you encountered a bug please send an email to nowface1221@gmail.com

Also this is my github if you want or need it: <https://github.com/Shasquell/M.i.o.p>.

I apologize if this is uninteresting

فارسی:

هدف این برنامه:

من همیشه دوست داشتم مردم آنچه را که من در ذهنم تجربه کرده ام تجربه کنند، خوب بیشتر چیزهایی که در ذهنم وجود دارد را نمی توان به راحتی بیان کرد، این یکی را ولی ... این ایده من را می شود!

بنابراین من شروع به کار بر روی این پروژه کردم. هدف یا بهتر بگوییم "استفاده" از این برنامه این است که بتواند دستورات داده شده مستقیماً به برنامه را اجرا کند یا یک فایل اسکریپت حاوی کد برای برنامه را اجرا کند.

خوب بله، می دانم که بسیاری از افراد قبلاً این کار را انجام داده اند و نتایج بهتری داشته اند

اما اولاً آنها پروژه خود را بزرگ شروع نکردند، بلکه آن را گسترش دادند و در دراز مدت آن را رها نکردند.

ثانیاً آنها متخصص بودند و به همه چیز یا حداقل چیزهایی که نیاز داشتند دسترسی داشتند، و همچنین یاد گرفتند که چگونه به درستی کدنویسی کنند، برخلاف من که به سختی کدنویسی را بلد هستم، آنها نیز اکثراً دانشجو و برخی دبیرستانی بودند.

نه من کار آنها را کوچک نمی کنم بلکه به سادگی می گویم که دلیل بد بودن من در کدنویسی این است که تجربه ندارم و این پروژه را برای کسب تجربه ساختم.

هدف این است که این برنامه به افراد تجربه برنامه نویسی مورد نظر را بدهد، اما در واقع بسیار دور است و به کمک هر کسی که می تواند کمک کند نیاز دارد. این هدفش است و هر چیز، همیشه به اهدافش نمی رسد. اما تجربه همیشه به دست می آید.

نحو و سیستم پایه:

سینتکس این برنامه کاملاً ساده است و دستور help آن را توضیح می‌دهد. اگرچه می‌دانم که برای برخی ممکن است مبهم و مبهم به نظر برسد...

بنابراین در این قسمت از راهنما، نحو برخی از دستوراتی را که این برنامه در اختیار شما قرار می‌دهد را توضیح خواهم داد. همچنین کمی توضیح خواهم داد.

اولین دستوری که می‌خواهم توضیح دهم دستور «ساختن» است. بسیار ابتدایی است اما به زمینه‌ای نیاز دارد. همچنین یک دستور پایه است زیرا فرمانی است که متغیرها را تعریف می‌کند. اگر نمی‌دانید چه متغیرهایی هستند ابتدا دبستان را تمام کنید و سپس برگردید. همچنین لطفاً به خاطر داشته باشید که هر دستور با حروف کوچک است و هیچ کاراکتری بزرگ نیست.

دستور make

نحو دستور make

[متغیر نام] -make

[متغیر نام] make: شود خوانده باید=

یک متغیر با نام متغیر داده شده ایجاد می‌کند / می‌سازد که در واقع می‌تواند هر رشته‌ای باشد که حاوی فاصله نباشد و بله می‌توانید متغیری بسازید که فقط شامل اعداد یا مانند آن باشد ... به معنای واقعی کلمه کل معنای یک عدد را بازسازی کنید.

[متغیر محتوای] [متغیر نام] -make

[متغیر محتوای] حاوی [متغیر نام] ساخت=

این همان پارامتر اول برای پارامتر اول است ، اما پارامتر دوم یا هر چیزی که بعد از پارامتر اول نوشته می‌شود می‌تواند هر نوع داده‌ای باشد اما نباید به صورت زیر باشد:

ساخته شده از فقط کاراکترهای فاصله - متغیر را خالی تنظیم می‌کند -
فقط از نقطه ها ساخته شده است - ممکن است باعث خرابی و مشکلات شود

همچنین اگر نام دیگری برای متغیر دیگری بنویسید، محتوای آن متغیر را نمی‌گیرد و آنها را در این یکی می‌نویسد.

دستور حذف

سینتکس دستور delete به صورت زیر است:

[متغیر نام] -delete

اگر متغیری وجود داشته باشد آن را حذف می‌کند یا به کنسول می‌نویسد که متغیر وجود نداشته است بنابراین نمی‌توان آن را حذف کرد.

دستور set

نحو دستور set به صورت زیر است:

[محتوا] [نام متغیر] -set

کنید تنظیم [متغیر محتوای یا نام] روی را [متغیر نام] محتوای =

پارامتر اول مجموعه دستورات خود توضیحی است
اما پارامتر دوم می تواند نام متغیر دیگری باشد که بدیهی است به محتوای آن متغیر دیگر تغییر می کند.

دستورات add و sub

نحو آنها این است:

[متغیر نام] [عددی های داده] -add/sub

[متغیر نام] از/به [عددی های داده یا متغیر نام] =add/sub

عدد داده شده را به متغیر اضافه می کند
/ عدد داده شده را از متغیر کم می کند

دستورات ضرب، تقسیم و باقیمانده

نحو آنها این است:

[عددی های داده] [متغیر نام] -multiply/divide/remain

[عددی های داده یا متغیر نام] بر [متغیر نام] تقسیم/ضرب =

لطفاً توجه داشته باشید که همه اینها بر خلاف دستورهای add/sub عمل می کنند
و همچنین باقیمانده «Remainder of Division» یا این «%» است اگر با آن آشنایی دارید.

فرمان سیستم

نحو دستور سیستم:

-سیستم "رشته"

یک فرمان سیستمی را اجرا می کند یا به معنای واقعی کلمه رشته داده شده را به ورودی کنسول منتقل می کند.

[متغیر نام] -svstem

متغیر داده شده را به سیستم منتقل می کند.

فرمان بیرون

نحو دستور out

خارج از "رشته"

رشته داده شده را در کنسول می نویسد.

[متغیر نام] outv-

[متغیر نام] خارج متغیر =

این دستور یک متغیر را می گیرد و در کنسول می نویسد.

فرمانده

نحو دستور in

[متغیر نام] in-

ورودی را از کاربر می گیرد و آن را در متغیر داده شده ذخیره می کند.

لطفاً توجه داشته باشید که این دستورات در آخرین لحظه اضافه شدند) در نسخه 0.1.2 اولین ساختی که من عمومی کردم (بنابراین ممکن است حاوی برخی از اشکال باشد!

دستور insert

نحو دستور insert به صورت زیر است:

[در اندیس] [نام متغیر] [داده رشته ای] insert-

[اندیس] در [نام متغیر] به [نام متغیر یا داده رشته ای] insert=

دستور حذف

سینتکس دستور remove به صورت زیر است:

[حذف مقدار] [فهرست از] [متغیر نام] remove-

«حذف مقدار» گویدمی بلکه گوید، نمی «فهرست به» که باشید داشته توجه لطفاً

لطفاً توجه داشته باشید که اسکریپت نویسی نیز چیزی است که من به تازگی شروع به ور رفتن با آن کردم، بنابراین مشکلات خود را خواهد داشت!

اسکریپت نویسی:

هنگام اجرای یک اسکریپت با استفاده از این برنامه مواردی وجود دارد که تغییر می کنند.

یکی از آنها این است که شما نیازی به نوشتن خط تیره در فایل اسکریپت ندارید.

برای اجرا نشدن یک خط کد یک '\' در آغاز آن قرار دهید.

دستور پرش

نحو دستور jump

[خط شناسه] -jump

پرش به خط مشخص شده در فایل اسکریپت
به یاد داشته باشید که شناسه خط اول صفر است.

دستور jumpif

نحو دستور jumpif

[شناسه خط] [داده] [اپراتور] [داده] -jumpif
[داده] [اپراتور] [داده] اگر [خط شناسه] به پرش =

اگر عبارت if درست باشد پرش می کند و اگر نادرست باشد نمی پرد.

دستورات oute و outve

نحو آنها این است:

"رشته" -oute

هر چیزی غیر از دستور و فضای بعد از آن را در کنسول می نویسد و در انتهای آن یک خط پایان اضافه می کند.

[متغیر نام] -outve

متغیر داده شده را در کنسول می نویسد و در انتهای آن خط پایانی اضافه می کند.

کلمات اخر:

اگر با یک اشکال مواجه شدید، لطفاً یک ایمیل به nowface1221@gmail.com ارسال کنید.

گیت هاب من: <https://github.com/Shasquell/M.i.o.p>

چنانچه این برنامه خوب و جالب نبود عذرخواهی می کنم.