



# Math Interpreter

Problem set



امروزه پایتون از جمله زبان های برنامه نویسی پر کاربرد در زمینه ریاضیات است. به طوری که شما می توانید کدی برای جمع کردن، کم کردن، ضرب کردن و یا تقسیم کردن مقادیر و حتی متغیرها بنویسید. بیایید برنامه ای بنویسیم که به کاربران این امکان را بدهد تا فرمول های ریاضی حل کنند، حتی بدون اینکه پایتون را بشناسند.

در فایلی به نام `interpreter.py` برنامه ای بنویسید که یک عدد حسابی را به عنوان ورودی از کاربر دریافت کند و سپس خروجی برنامه را بصورت عدد اعشاری، با یک رقم اعشار برگرداند. فرض کنید ورودی کاربر به صورت  $(z \ y \ x)$  و همراه با یک اسپیس بین  $x$  و  $y$ ، و یک اسپیس بین  $y$  و  $z$  باشد. طبق توضیحات زیر:

- $x$  یک عدد صحیح هست.
- $y$  برابر است با  $+$ ,  $-$ ,  $*$  و  $/$ .
- $z$  یک عدد صحیح هست.



به عنوان مثال اگر کاربر ورودی  $1 + 1$  وارد کند، برنامه‌ی شما باید خروجی 2.0 نمایش دهد.  
توجه کنید اگر کاربر مقدار  $z$  را / وارد کند، آنگاه مقدار  $z$  نباید برابر شود با 0.

### ▼ نکته

- به یاد داشته باشید که string توابع مختلفی دارد. توضیحات بیشتر را در اینجا می‌توانید ببینید: [docs.python.org/3/library/stdtypes.html#string-methods](https://docs.python.org/3/library/stdtypes.html#string-methods) که از جمله‌ی آنها split است، که یک string را بر اساس مقداری که به آن ورودی داده اید به بخش‌های مختلف تقسیم می‌کند و یک لیستی از مقادیر می‌سازد و می‌توان این لیست را به یکباره به تعدادی از متغیر نسبت داد. برای مثال اگر مقدار متغیر expression یک string مانند  $1 + 1$  باشد، به متغیر  $x$  عدد 1 و به متغیر  $y$  علامت  $+$  و به متغیر  $z$  عدد 1 اختصاص داده می‌شود.

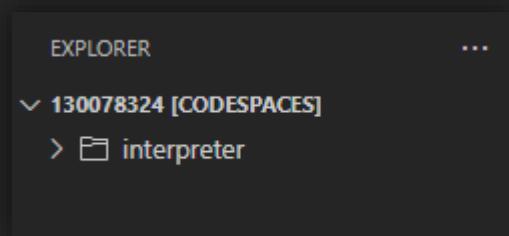
```
expression = ""  
x, y, z = expression.split(" ")
```

وارد [code.cs50.io](https://code.cs50.io) شوید، سپس بر روی پنجرهی Terminal کلیک کنید. توجه داشته باشید که دستور پنجرهی Terminal شما باید به صورت زیر باشد:

\$

سپس کد زیر را اجرا کنید تا یک پوشه به نام interpreter در codespace ایجاد شود:

```
$ mkdir interpreter
```

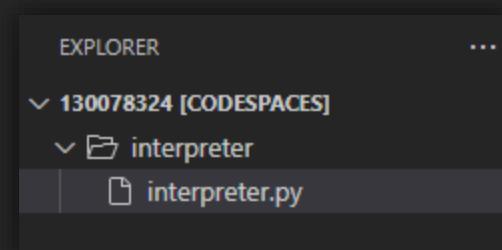


سپس interpreter را اجرا کنید تا به پوشه‌ی interpreter منتقل شوید. اکنون شما باید آدرس \$ interpreter/ را در Terminal مشاهده کنید:

```
$ cd interpreter  
interpreter/ $
```

حالا می‌توانید دستور code interpreter.py را اجرا کنید تا فایلی با نام interpreter.py ایجاد شود:

```
interpreter/ $ code interpreter.py
```



اکنون می‌توانید در این فایل برنامه‌ی خود را بنویسید.

اکنون می‌خواهیم برنامه‌ی نوشته شده‌ی خود را تست کنیم:

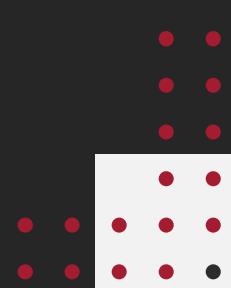
- دستور `python interpreter.py` را اجرا کنید.  $1 + 1$  را تایپ کنید و کلید Enter را فشار دهید. خروجی شما برابر می‌شود با: 2.0
- دستور `python interpreter.py` را اجرا کنید.  $3 - 2$  را تایپ کنید و کلید Enter را فشار دهید. خروجی شما برابر می‌شود با: -1.0
- دستور `python interpreter.py` را اجرا کنید.  $2 * 2$  را تایپ کنید و کلید Enter را فشار دهید. خروجی شما برابر می‌شود با: 4.0
- دستور `python interpreter.py` را اجرا کنید.  $5 / 50$  را تایپ کنید و کلید Enter را فشار دهید. خروجی شما برابر می‌شود با: 10.0



- ❖ شما می‌توانید از آدرس زیر برای بررسی کردن کد خود استفاده کنید. CS50 از این برنامه برای آزمایش کد شما استفاده می‌کند. از این دستور استفاده کنید تا کدهایتان را امتحان کنید.

```
$ check50 cs50/problems/2022/python/interpreter
```

لبخند های سبز به این معنی هستند که برنامه‌ی شما در تست قبول شده و اخم های قرمز یعنی برنامه‌ی شما دارای ایراد هست. با مراجعه به URL check50 می‌توانید خروجی مورد انتظار و خروجی برنامه‌ی خود را بررسی کنید.



با اجرا کردن صورت زیر در Terminal پاسخ خود را ارسال کنید.

```
$ submit50 cs50/problems/2022/python/interpreter
```

# CS50x Iran

Harvard's Computer Science 50x Iran

