کورس udemy – 100 days of code

Day001

میتونیم برای کمک گرفتن در کورس با #100daysofcode توی تویتر مطلب پیدا کنیم.



میخوایم the band name generator بسازیم.

Hello world رو رفت و داره استرینگ توضیح میده .

توی pycharm داره پیش میره ، و جالبیش اینه که توی Pycharm میتونیم اون هایلایتی که زیر چیزای اشتباه میندازه رو هاور کنیم و نکته ش رو ببینیم. فکر کنم این توی vscode نیست.

-نکته جدید: خط آخر توی اکثر زبان ها بهتره که خالی باشه. وگرنه ai که چک میکنه اشکال تشخیص میده و Char آخر رو Highlight میکنه .

-یادآوری: \n برای رفتن به خط بعدی در خروجی

Concat با +

یه نرم افزار معرفی کرد به نام thonny که میتونیم باهاش کدی که نوشتیم رو ببینیم مرحله به مرحله توسط کامپیوتر چطور اجرا میشه. این جالب میتونه باشه و در ادامه استفاده های زیادی ازش بکنیم . شاید Pycharm و vscode هم داشته باشنشون .

تا 012 Variables

ادامه

همونطور که میتونیم باید از underline استفاده کنیم به جای camelCase .

یادآوری: len() برای گرفتن تعداد char ها.

-توی vscode توی اجرای کد هایی که توی هرجایی بجز c ذخیره شده به مشکل خوردیم . توی همون pycharm پیش میریم. یا عشقمون کشید با jupyter میریم .

تا انتهای day001

ادامه

Day002

Len() برای Int عمل نمیکنه و میگه که no length . باید conversion انجام بدیم روی Int ها وعدد ها اگر میخوام Len بگیریم .

با گذاشتن براکت جلوی هر چیزی که Length داره میتونه خروجی Index مربوطه رو داشته باشیم

Index منفی هم میتونیم استفاده کنیم و از آخر میشماره

Concatenation

نکته جالب: میتونیم بین اعداد Underline بزاریم به جای کاما تا خوندشون راحت تر بشه

Type() برای چک کردن data type

Type conversion در یسری حالتا غیر قابل انجامه.



یادآوری: اولی Float میده به صورت پیشفرض ، دومی ceil میکنه

\*\* توان

Round() روی اعداد >> گرد میکنه بسته به مقدار اعشار به بالا و یا به پایین 5 بشه به بالا

Round(x,x) مقدار دوم میتونیم اعلام کنیم تا چند رقم اعشار گرد کن.

Fstring : print(f”string{value}”

تمرین امروز



تا وسطای 009 Day 2 Project Tip Calculator

ادامه

ما برای تمرین روز دوم از number: .2f استفاده کردیم تا اعشارش رو محدود کنیم ولی خودش از round(x, x) استفاده کرده. اولی متغیری هست که میخوایم روند شه ، دومی مقدار روند کردنشه.

Day003 treasue-islad-end

تا 003 Control Flow with if else and Conditional Operators

ادامه از 003 Control Flow with if else and Conditional Operators

Modulo operator یا همون % خودمون. باقیمانده.

تمرین:



Nested if/else

تودرتو









-داخل pycharm یه سیستم تشخیص simplify داریم که میتونیم روش کلیک کنیم و همون کدی که نوشتیم رو ساده ترش رو بهمون بده .



<https://ascii.co.uk/art>

یادآوری: برای اینکه بتونیم یچیزی مثل صندوق ابتدای تمرین نهایی رو print() کنیم باید به جای “” از ‘’’ ‘’’ استفاده کنیم که 3 تا Single quotation میشه. این اجازه میده چنیدن خط رو یکجا پرینت بگیریم.

پروژه نهایی هم در انتها اینه که یه بازی ابتدا بر اساس diagram زیر بسازیم. بعدش اگر حست کشید بزرگتر و بسط داده ترش کن یا یکی دیگه بساز با یه diagram دیگه .



پایان day002

(پروژه day003 بازی سنگ کاغذ قیچیه. که بیشتر با مفهوم randomization و Python Lists قراره کار کنیم.)

ادامه

Day004

پروژه نهایی ساخت یک بازی سنگ کاغذ قیچی

Randomization

برای اینکه بتونیم از مقادیر random استفاده کنیم باید از module اضافه کنیم به برنامه :

Import random

که از.random() و .randint() استفاده میکنیم برای گرفتن یک مقدار random که البته فرق کوچیکی دارن که با یه سرچ در میاد (مثلا random خودِ صفر رو در نظر میگیره تا اون عددی که بهش دادی)

مفهوم module : میتونیم بخش هایی از برنامه رو



میتونیم ضرب هم کنیم که دو parameter رو ضرب میکنه از اونجایی که یکیشون صفره اونیکی فقط اعمال میشه عملا

یه برنامه نوشتیم واسه "شیر یا خط" که یه سکه پرت میکنیم.

Python list

یه data structure عه .



یادآوری: با Index میتونیم یک item رو return کنیم. با مقدار منفی از انتها شمرده میشه.

aList.add() و aList.extend() و...

میتونیم با random.choice() از item های یک List به صورت random returnداشته باشیم.

Nested list : میتونیم داخل یک List ، List تعریف کنیم (داخل index ها)

تا 007 day 004 03:17

ادامه

برای اینکه بتونیم برنامه ای که نوشتیم رو کاری کنیم دائم در حال اجرا باشه و در صورت اینکه کاربر قصد پایان داشت خاتمه پیدا کنه باید از ساختار کلی زیر در پایتون استفاده کنیم (که روی پروژه روز 4 سنگ کاغذ قیچی پیاده ش کردیم) :



حالا باید بریم ببینیم روش خودش چیه

1-ascii art هارو داخل یک List ذخیره کرد که با مقادیر وارد شده توسط user مطابقت داره که باعث میشه فقط بنویسه List\_name [user\_choice] که برای پرینت کردن شکل ها خیلی راحت تره. (کاری که ما کردیم این بود که یه module براش ساختیم و هر دفعه برای پرینت کردن هرکدوم باید این رو مینوشتیم: module\_name.scissors و یدور هم نیاز به Import module\_name داشت. که خب کاری که خودش کرده راحت تره)

ادامه

جلسه قبل در طی ور رفتن با گیت فایل هایی که نوشته بودیم پرید ، ولی از دوباره یه repository ساختیم که اونجا Push کنیم و نیاز نیست از دوباره کد هارو بزنیم مربوط به ابتدا رو ، فقط از اینجا به بعد رو انجام میدیم .

Day005

پروژه امروز PyPassword generator هست .



جالب: یه سایت معرفی کرد برای اینکه چک کنیم email رو و اگر شرکتی مورد هک واقع شده و اطلاعات شما هم اونجا بوده و رفته بهتون بگه. ([link](https://haveibeenpwned.com/))

خب ما دوتا راه داریم این کارو انجام بدیم: easy version و hard version

ابتدا میریم سراغ easy .

در easy به ترتیبِ حروف ، سمبل، عدد رفتیم ولی در hard باید بدون ترتیب مشخص پسوورد ساخته میشد. که ما از خودمون با random درستش کردیم. ولی داخلِ خودِ random یه method هست به نام shuffle() که همین کار رو انجام میده .

Random.shuffle(List\_name) یک Object رو به صورت درهم خروجی میده.

در hard version میتونستیم از روش های مختلفی بریم. از اونجایی که easy version رو رفته بودیم پس یعنی تعداد در اون درسته ، فقط میمونه ترتیب رو به هم بریزیم که با random.shuffle انجام میشه . چیزی که ابتدا تو ذهنم بود این بود که برای هر جایگاه به صورت تصادفی از یکی از لیست ها انتخاب بشه و بعد به صورت اتفاقی از داخل همون لیست انتخاب بشه و این بره تا جایی که ورودی های کاربر تشکیل بشه. یعنی وقتی تعداد Number هایی که کاربر میخواست touch شد دیگه اون لیست رو داخل امکانات نزاره.

ادامه

قبل از اینکه ادامه بدیم برای اینکه این چیزایی که بلدیم رو سریع بزنیم بره بعدی، چطوره همون ابتدا سعی کنیم پروژه پایانی رو بسازیم. اگر تونستیم میتونیم اون روز رو Skip کنیم.

Day006

با یه تمرین باحال روبروییم اینجا:

<https://reeborg.ca/reeborg.html?lang=en&mode=python&menu=worlds%2Fmenus%2Freeborg_intro_en.json&name=Hurdle%201&url=worlds%2Ftutorial_en%2Fhurdle1.json>

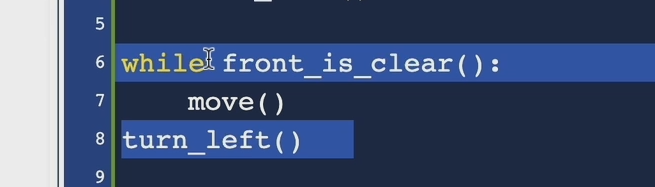
که باید با کد های پایتون این رباته رو ببریم جایی که پرچم هست.



تا مرحله hurdle4 میره و بعدش میره سراغ maze که همه رو حل کردیم کداش هست. (برای مرور خوبه یه دور چک کنی)

3 تا مسئله هم برای debug داره که اونا هم حل کردیم ولی یه راه خودش رفت که برای سه تاشون جواب بود و اون رو نمینویسم اینجا توی کد ها هم نیست. سعی کن بنویسیش توی مرور.

(



این قطعه کد رو میزاره قبل از تابع اصلی تا قبل از اینکه شروع به هرکاری کنه، مطمئن بشه دیوار رو پیدا میکنه چون الگوریتم بر این اساسه که دیوار سمت راستشو بگیره و بره. لازم به ذکره داریم فقط و فقط راجب maze صحبت میکنیم.

)

ادامه

Day007

جلسه قبلی project نداشت و فقط درس بود. اما این جلسه Hangman رو داریم.

طبق چیزی که جلسه قبلی گفتیم سعی میکنیم خودمون بزنیمش اگر نتونستیم میریم با خودش پیش میریم.

خب چیزی که میبینم اینه که به تنهایی از این یکی بر نمیایم. با خودش پیش میریم. (روی این با خودش پیش میریم خیلی تاکید میکنم، چیزی رو skip نکن اگر مردی سرعت ببخش بهش. توی این تمرین ها قراره مسیر رو یادمون بده و راه و روش رو. اگر یه بخشی رو بلدی دلیل نمیشه بقیه رو هم بلد باشی. )

جالب اینه که Flow chart کشیدن رو یه مرحله از برنامه نویسی میدونه و باید قبل از کد زدن انجامش داد. با ابزار draw.io آنلاین میتونیم این کار رو بکنیم.

توضیح بازی:

اینطوریه که یه کلمه باید حدس زده بشه. تعداد حدس ها هم یه تعداد محدودیه (هربار که اشتباه حدس زده بشه یه قسمت از شکل hangman (انسانی که به دار آویزونه) رو میکشه و هروقت شکل کامل بشه یعنی ما باختیم و اگر قبل از اینکه جونامون تموم بشه کلمه رو حدس بزنیم ، بردیم .

حالا باید با این توضیحات و چیزی که دیدیم یه flow chart بکشیم که این پروژه چطور کار میکنه Logically

Enumerate() میتونیم str بفرستیم داخلش و به عنوان خروجی بهمون index و char رو میده میتونیم استفاده کنیم . (که با استفاده از اون داخل یک for میتونیم iterate کنیم و برای هرکدوم داشته باشیمش. )

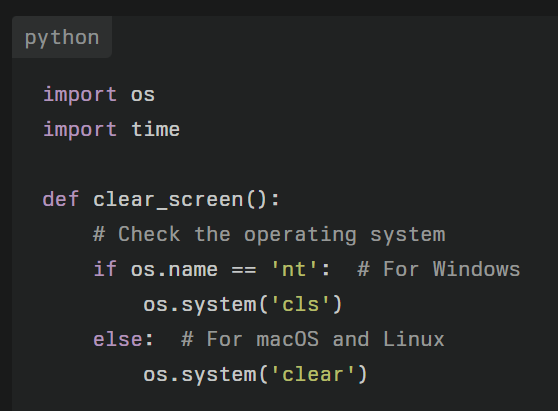
با ذهن خسته نمیشه کار کرد ، از یه مسیر خیلی ساده ای میشد مشکل حل بشه که داخل ویدیو خودش رفت و ما مدتی روش فکر کردیم ، نکته اصلی اینه که نباید سعی کنیم جلو جلو بریم ، وقتی داریم step step میریم باید باهاش پیش بریم و کاری که خواسته رو بکنیم.

باقیش برای فردا

تا انتهای 004 از day007

ادامه

-نکته: میتونیم برای اینکه در مراحل مختلف برنامه cls بگیریم داخل کد از :



استفاده کنیم. البته باید import subprocess هم بکنیم.

پیشنهاد: میتونیم برای تمرین یا مرور سعی کنیم از ابتدا project های هر روزی که هست میخوایم مرورش کنیم رو بزنیم از ابتدا. دفعه بعدی این کار رو بکن ببین efficient هست از لحاظ زمانی یا نه

تا انتهای day007

ادامه

باید بدو بدو بریم تا جلسه 15. در حداقلی ترین و سریع ترین حالت ممکن پیش میریم. بجز نکات مهم چیزی نمینویسیم اینجا .

Day008

Encode decode کردن با shift کردن به تعداد مشخص (Caesar cypher )

یادآوری: میشه در تعریف function به جای Positional arg از حالت parameter=”value” استفاده کرد که دیگه ترتیب مهم نباشه. ولی خب باید Parameter هارو بدونیم.



(عکس بالا برای شیف کردن استفاده شد که به جای اینکه چندین سری از alphabet رو اد کنیم توی List که جا داشته باشه و ادامه بده به میزان اون عدد ، اینکار باعث میشه user هر عددی خواست وارد کنه و معادلش در حالت شیف شده بدست بیاد. )

تا day008 007 12:31

ادامه

برای نوشتن Caesar() یه ترفند جالبی زد که با توجه به فرایند فقط نیاز بود shif \*= -1 بشه تا decrypt انجام بشه. همین. لازم نبود مثل چیزی که خودمون نوشتیم یه func جدا تعریف کنیم.

پایان day008

Day009

Dictionaries and nesting

Silent auction program

یادآوری dictionary\_name = {“key” : “value”, “key2”: “value”, … }

برای return کردن: dictionary\_name[“key”]

برای مقدار اضافه کردن : dictionary\_name[“new\_key\_name”]= “value”

برای تغیر یک key : dictionary\_name[“key\_for\_change”] = “changed value”

-در iteration ها key return میشه

-برای بیشترین مقدار max(dictionary\_name)

تا ابتدای day010

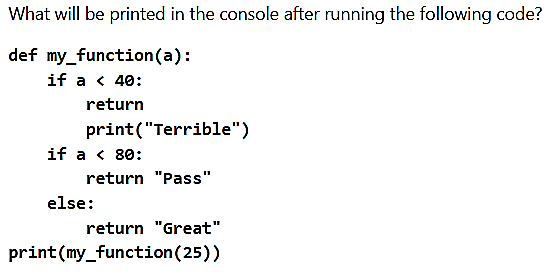
ادامه

‘string’.title() کاری میکنه string مون هرجوری از لحاظ بزرگی و کوچیکی بود اولین char بزرگ بشه بقیش کوچیک

-یادآوری: وقتی return انجام میشه func تموم میشه.

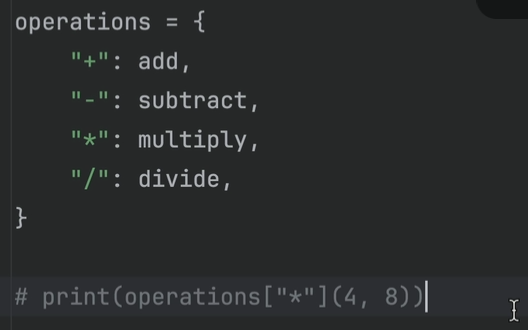
-یادآوری: برای function های داخلی python ما docstring داریم که وقتی روش hover میکنیم اون رو نشون میده و بهمون میگه که این function در واقع چکار میکنه چیا میگیره با چه data type ای و چی پس میده و ... ، ما برای Function هایی که خودمون میسازیم هم میتونیم این رو ایجاد کنیم با تنها اضافه کردن یک string به خط اول function مون. که اگر تعداد خطوط زیاده میتونیم از ‘’’’’’’ استفاده میکنیم. بعد وقتی Hover میکنیم اون چیزی که تو توضیحات نوشتیم نشون داده میشه. توی تمرینات exercism هم داشتیم میتونیم بریم اونجا نکات بیشتری یادمه داریم./

-نکته: در ازای این کد ها None میگیریم:



چرا؟ چون 25 که وارد میشه اولین return که جلوش هیچی نیست و میشه None اجرا میشه چراکه شرطش برقراره و سراغ باقی کد ها نمیره.

یادآوری: تغیر نام function چه داخلی چه چیزی که خودمون نوشتیم فقط باید یک variable بنویسیم و اسم اون function رو بدون پرانتز (چون پرانتز بزاریم call میشه) بزاریم اون داخل. از این به بعد با اسمی که برای اون variable گذاشتیم هم میتونیم function call کنیم که میشه توی ساختاری مثل این ازش استفاده کرد:



(فقط باید مواظب باشیم که نباید در حین این کار Function call انجام بشه ، یعنی نباید پرانتز بزاریم وقتی میخوایم تغیر نام بدیم . بدون پرانتز باید استفاده کنیم .)

تا day010-007-12:13

ادامه

Day011

Black jack game

این بازی رو قبلا ساختیم . ولی سراغ چیزایی که قبلا نوشتیم نمیریم و از ابتدا باید بسازیم

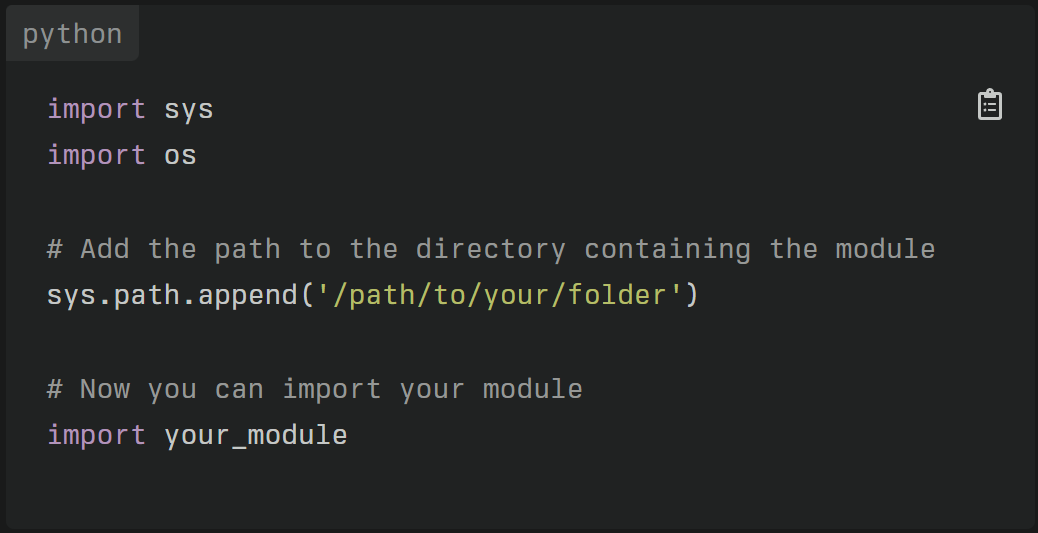
Flow chart رو ساختیم

تا day011 002 end

جلسه بعد از روی flow chart کد بزن

ادامه

الان میخوایم از روی flow chart خودمون کدش رو بزنیم

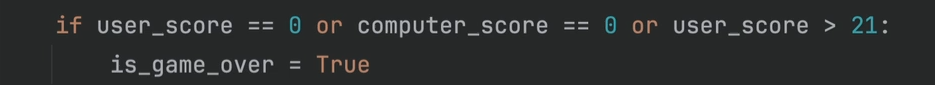
نکته: 

این رو AI داده برای ascii art هر ورق، شاید بعدا ادش کردم توش:



-برای اینکه جمع همۀ عناصر یک list رو داشته باشیم sum()

ادامه



ما به ترتیب به هردو کارت میدادیم ولی اینجا توی دوره اول به user کارت میده بعد شروع میکنه به pc کارت میده تا زمانی که pc < 17 باشه کارت دادن ادامه داره. البته واسه زمانیه که ما گفتیم دیگه کارت نمیخوایم به عنوان user. یعنی کارت دادن به pc بعد از user اتفاق میوفته.

ادامه

Day012

Number guessing project

درمورد Scope و اهمیتش گفت.

نکته یادآوری مهم : برای اینکه از داخل یک function بخوایم یک global var رو Modify کنیم باید یبار داخل Scope ای که برای function داریم ذکر کنیم که یک متغیری هست به این نام و global هست و ما میخوایم اون رو تغیر بدیم. که میشه global variable\_name و این باید داخل scope باشه.

-البته میگه که اینکارو نکنیم بهتره، چون باعث باگ میشه و دیباگ کردنش سخته . اگر خواستیم Modify انجام بدیم روی یچیز global با Function بهتره از return استفاده کنیم و یجوری return کنیم که وقتی برابر با اون متغیر قرارش میدیم اون Functionality ای رو داشته باشه که میخوایم . ولی درمورد const variable فرق میکنه .

یادآوری: Const چیزیه که define میکنیم و نمیخوایم که تغیرش بدیم . در این صورت میتونیم global رو اعلام کنیم ولی باید این رو یادمون باشه که تغیر دادنش ممکنه، یعنی مثل باقی زبان ها نیست که نشه تغیرش داد، عملا همون متغیر عادیه. ولی با حروف بزرگ نوشته میشه که یادِ خودمون بمونه که این const عه . (و اینطوری هم هست که const میسازی و داخل function ها اگر خواستی استفاده کنی global میزاری پشتش و تمام. هر بلایی هم داخل سرش بیاری هرجایی برای ابتدا نوشتی ایجادش کردی اونجا هم در نظر گرفته میشه. )

-علاوه بر اون سایت که برای ascii art معرفی کرد که حاضر آماده بود ، این سایته هم هست:

<https://patorjk.com/software/taag/#p=display&f=Ghost&t=Type%20Something%20>

که میشه بری هرچیزی که مینویسی رو برات ascii art کنه بهت بده

انتهای day012

ادامه

Day013

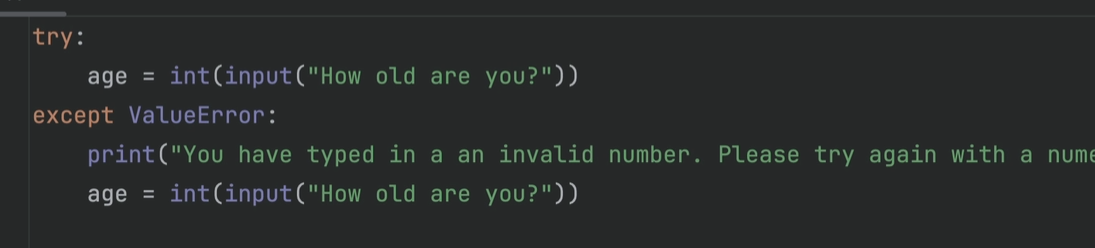
درمورد روش های debug داره صحبت میکنه. چطور مشکل رو حل کنیم.

1.اگر مشکل دارید، اول باید سعی کنیم مشکل رو توضیح بدیم برای خودمون و برای خودمون حداقل روشن کنیم که قضیه از چه قراره. جمله ای که گفت : (تقریبا غیر ممکنه وقتی که خودتون از مشکل خبر ندارید بتونید حلش کنید)

2. چیزی که نوشتید ببینید دقیقا چیکار میکنه، چیکار میخواستید بکنه باید چه چیزی رو تغیر بدید

نکته مهمی هم که وجود داره اینه که مشکلی رو میشه حل کرد که بشه از دوباره خلقش کرد. به تعبیری بدونی در ازای چی ارور میگیری یا مشکل داری دراون صورت میتونی حلش کنی. با تغیر دادن random و بدون حساب کتاب گاها شاید مشکل حل بشه ولی توی روند فرسایشی میوفتیم .

یادآوری ساختار try except ، وقتی که یه ارور توی یک بخش مشخص از کد پیش میاد و میدونیم ارور چیه (مثلا ValueError) اون رو بگیریم و تبدیل کنیم به یه پیام مشخص تر یا یه عملیات کاربردی در طی پیش اومدن اون مشکل:



و این هم هست که در طی این try except کد در محل error بلاک نمیشه و روند تا انتها ادامه پیدا میکنه .

نحوه کار با debugger هم اینطوریه که شما ران میکنی و این خط به خط اجرای کد رو بهت نشون میده. یه step over داریم و step in و توی Pycharm یدونه داریم که داخل Module های Inner نمیره و فقط module های خودمون رو برسی میکنه . اینطوری میتوینم خط به خط ببینیم چی میشه و اگر یسری خطوط رو میخوایم بزنیم بره جلو از breakpoint همون نقطه قرمزه که بقل عدد های خطوط کد میشه اضافه ش کرد قرار میشه داد. از اونجایی که breakpoint هست Debug شروع میشه .

اهمیت Debugger از اهمیت print() میاد، که وقتی خروجی میگیری از چیزی که نوشتی بهتر میتونی بفهمی چه اتفاقی داره میوفته. حالا debugger در عمل داره برای همه چی Print() استفاده میکنه و خط به خط نشونت میده چه اتفاقی داره میوفته. با استفاده کردن مکرر از debugger که در IDE های مختلف ساختارش یکیه، میتونیم مهارتمون رو هی بهتر و بهتر کنیم. سر کلاس های c++ , java با استاد عمرانی بهش میگفتیم تحلیلِ برنامه ، که خط به خط میرفتیم و برسی میکردیم چه اتفاقی داره میوفته. جالبه

نکته بعد هم درمورد رفع مشکل و ارور، اینه که اگر هرکاری میکنی نمیشه ، شاید اون روز روزش نیست و باید استراحت کنی و یه مقدار زمان بدی بهش و بعد بیای سراغش. ما این رو میدونیم که بعد از گذروندن زمان و چرت زدن و خوابیدن و ... مغز ما این مهلت رو پیدا میکنه که بین اطلاعات و بخش های مختلف ارتباط برقرار کنه و اینطوریه که ما چیز هارو میفهمیم و به خاطر میسپاریم، این که کد زدنه درمورد ورزش و فعالیت های فیزیکی هم قضیه همینه، پس زمان دادن به مغز برای رفع مشکل خیلی مهمه.

در مرحله بعدی هم اگر نتونستی باید از یه ذهنِ انسانی دیگه بپرسی

نکته بعدی هم این که بعد از اینکه بخشی اضافه شد به کد و نوشتیم و پیاده ش کردیم ، بهترین کار همینه که همون لحظه یه اجرا ازش بگیریم و ببینیم چی پیش میاد.

باگ ها پیش میان، نکته اصلی اینه که حلش کنی و مهمتر اینکه راه های مختلف برای حل کردنش رو بلد باشی، چه مستقیم چه غیر مستقیم .

انتهای Day013