لیست کامپرشن ها

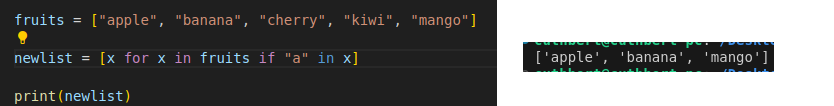
-----------------------------------------------------------------------------------------------------

ما میومدیم و از حلقه ها برای پر کردن لیست ها و انجام عملیات رو ی اونها استفاده میکردیم

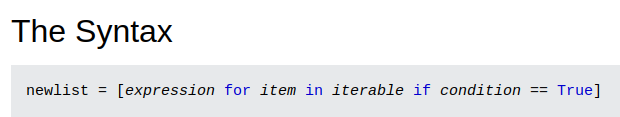
برای مثال :



طبق معمول اینجا اومدیم این مار رو کردیم ولی حال همین کد رو میخاییم معادل لیست کامپرشن شو بنویسیم :



حالا این کد اصلن چه معنی میده اصن؟ به سینتکتس اون توجه کنید :

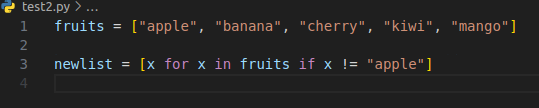


اون ایکس که اولش گذاشتیم یعنی اون چیز رو append کن به لیست مون . بعد حلقه مونو زدیم و گفتیم به اعضای اون ایکس هایی که توی لیست میوه ها هست . و بعدش هم یک شرط گذاشتیم که اگر اون شرط روی ایکس برقرار بود میاد اون رو append میکنه به لیست مون

کد بالایی و پایینی یک کاری را انجام میدهند ولی با کد نویسی کمتر

-----------------------------------------------

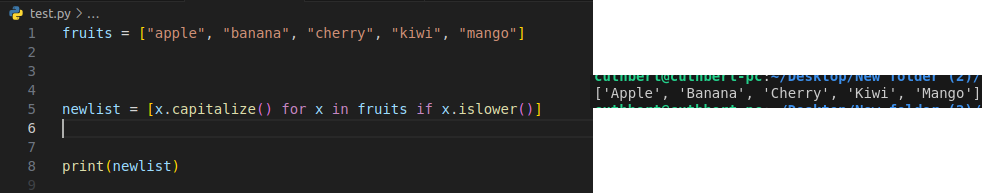
مثال بعدی رو ببینید تا متوجه بشید بیشتر :



توی این کد گفتم که یه ایکس داریم که به اعضای اون در میوه ها است ولی فقط به شرطی اضافه شون کن که apple نباشن اگر پرینت کنیم لیست رو میبینیم که تمام میوه ها بجز سیب در لیست جدید وجود دارد.

---------------------------------------------

مثال دیگه :

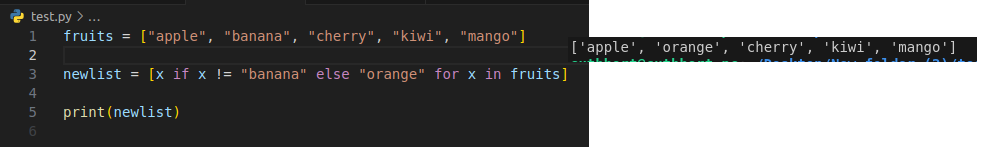


اومدم گفتم که اون ایکس هارو بزرگ شده شو میخام . به اعضای اون ایکسایی که توی میوه ها

هست اگر تابع islower() که روش اعمال میشه True برگردونه. بخش شرطی اگر درست

باشه اون ایکس رو قبول میکنه و میاد اولش که نوشتم x.capitalize() شده اش رو میریزه توی

لیست.



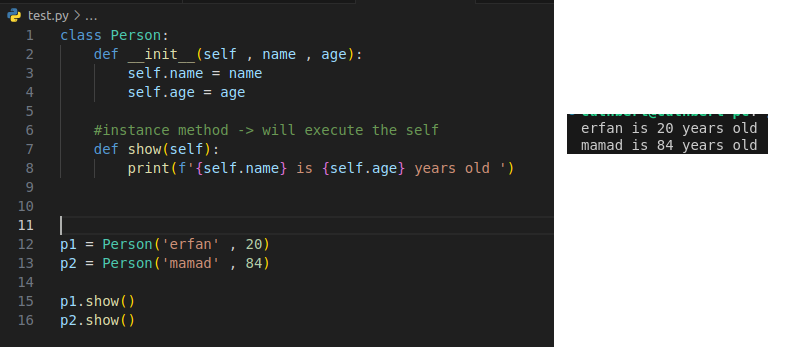
توی این یکی مثال اومدم گفتم که : اون ایکس هایی رو برداری بیار که اول اینکه bannana نباشه ولی اگر بود بجاش orange بزار . اون هایی رو که در میوه ها هستند. میتونیم شرط دیگه ای رو هم بزنیم مثلا مثل بالا بگیم اونایی که islower() اشون درسته و یا حتی اون اول ایکس رو روش بلایی بیاریم مثلا بزنیم x.upper() تا بیاد حروف بزرگ شده شو اضافه کنه

انواع متود ها : استاتیک . کلاس و اینستنس متود ها

-----------------------------------------------------------------------------------------------------

اینستنس متود instance method :

تمام متود هایی که در کلاس ها مینوشتیم و به آن self رو پاس میدادیم به آن اینستنس متود ) متود هایی که روی اینستنس ها ) آبجکت ها ( اجرا میشوند( میگویند :



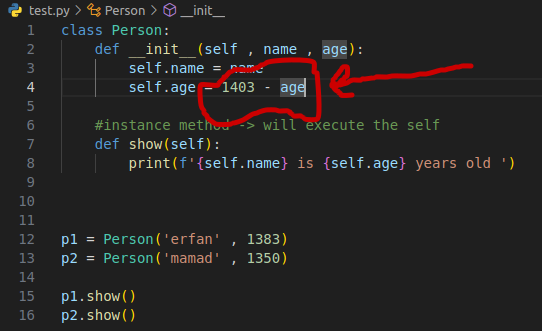
اینجا تابع show یک اینستنس متود هست چون داره با سلف کار میکنه و روی آبجکت هامون اجرا میشه p1.show() p2.show() اینجا ابجکت هامونو صدا زدیم

یا مثلا حتی داندر ها مثل داندر اس تی ار . چون وقتی از ابجکت رشته میگیریم فعال میشه بهش میگیم اینستنس متد

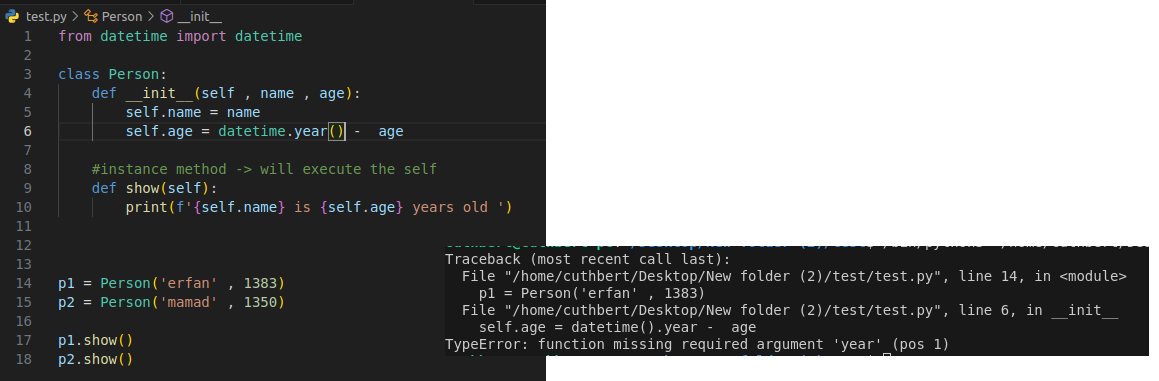
*--------------------------------------------*

مواقعی هست که ما میخاییم روی ورودی های داندر اینیت تغییر ایجاد کنیم و اون تغییر پیدا کرده رو ذخیره منیم مثلا ما از طرف سال تولد میگیریم ولی میخاییم سن رو ذخیره کنیم پس کسی که

پایتون سرش نشه میاد و این کار رو میکنه :



این کد به درستی کار میکنه ولی مانمیتونیم که هر سال جدید بیاییم برای کارفرما برنامه شو اپدیت کنیم ! پس باید از ماژول تاریخ استفاده کنیم ولی وقتی این کارو میکنیم به یه مشکلی میخوریم :



حالا این اروری که میگیریم ربطی به این نداره که این کار رو نمیشه کرد این ارور از ماژول

کتابخانه تاریخ هست ولی در کل این رو میخایم بگم که :

\*\*\*\* زمانی که میخاییم در ویژگی های ورودی ابجکت تغییری ایجاد کنیم و اون رو ذخیره

کنیم از کلاس متود ها استفاده میکنیم \*\*\*\*

حالا چجور ؟ بیایید ص بعد....

این تعریف درست تر هست که بگم که وقتی از این استفاده میکنیم که نیاز داشته باشیم به خود کلاس دسترسی داشته باشیم مثلا اینجا ما به خود کلاس دسترسی پیدا میکنیم تا بیاییم رو ی

مقادیر ورودی تغییری ایجا بکنیم .

مراحل ساخت یک کلاس متود :

اول : میاییم و بالای تابع مورد نظر یک ادساین میزاریم و مینویسیم classmethod تا این تابع رو به عنوان کلاس متود در نظر بگیره

10

----------------------------------------------------

دوم : دیگه چون اینستنس متود نیست و ما با ابجکت ها کار نداریم نمیاییم self بدیم بهش . چون میخاییم روی خود کلاس تغییر ایجاد کنیم مینویسیم cls که این یک مقدار قرار دادی مثل سلف هست ولی بجای اینکه ابجکت رو بگیره مثل سلف میاد خود کلاس رو میگیره

11

----------------------------------------------------

سوم: میاییم ارگومان هایی که قراره بگیریم رو مینویسم کنار اون سی ال اس . نکته : در آخر موقع بازگردانی این تابع باید مقادیری که در داندر اینیت داده ایم برگردانده شود تا به داندر اینیت برسد ولی حالا میتونید همینجا همه رو بگیرید و یا خودتون بسازید منظورم چیه :

12

موقع آبجکت سازی دیگه خود کلاس رو خالی صدا نمیزنیم تا داندر اینیت اش اجرا بشه اول میاییم اون کلاس متود رو صدا میزنیم تا اول اون اجرا بشه بعد بره سراغ داندر اینیت تا اول روی مقادیر ورودی تغییر ایجاد کرده باشیم بعد بره بشینه توی داندر انیت

اینم که میگم یا خودتون بسازیدش مورد آخره بعد از این بعدیه

چهارم : باید اون تابع ریترن بکنه cls رو یعنی میاد خود اون کلاس رو برمیگردونه داخل اون متغیری که ساختیم یعنی p1 و جلوی پرانتزش انتظار میره که مقادیر داندر اینیت رو بهش پاس بدیم چون داره خودشو برمیگردونه پس کلاس رو اجرا میکنه :



اون قضیه متغیر هارو بگم : ببینید توی تابع from\_birth که من اومدم کلاس متودش کردم اومدم ۲ تا مقدار فقط گرفتم ولی در صورتی که داندر اینیت مقدار ایمیل رو هم میخاد . خب اینجا مشکلی نیست چون من توی p1 مستقیم کلاس رو صدا نزدم که اومدم کلاس متد شو صدا زدم پس فقط ازم نام و تاریخ رو میخاد اومدم ایمیل رو خودم درست کردم با تلفیق اسم و تاریخ تولد و بعد موقع returnکردن که گفتم باید اون س ال اس ریترن بشه بهش مقادیر داندر اینیت رو پاس دادم زیرشم که هر کدوم به کدوم از ویژگی های داندر اینیت میره رو هم نوشتم به صورت کامنت ولی مثلا الان توی p2 اینکارو نکردم و مستقیم ابجکت ساختم و به داندر اینیت وصل شدم و ایمیل رو پاس دادم بهش و در خروجی هم مشکلی نمیبینم .

دوباره ص بعد از اول میگم گه چه اتفاقی افتاد

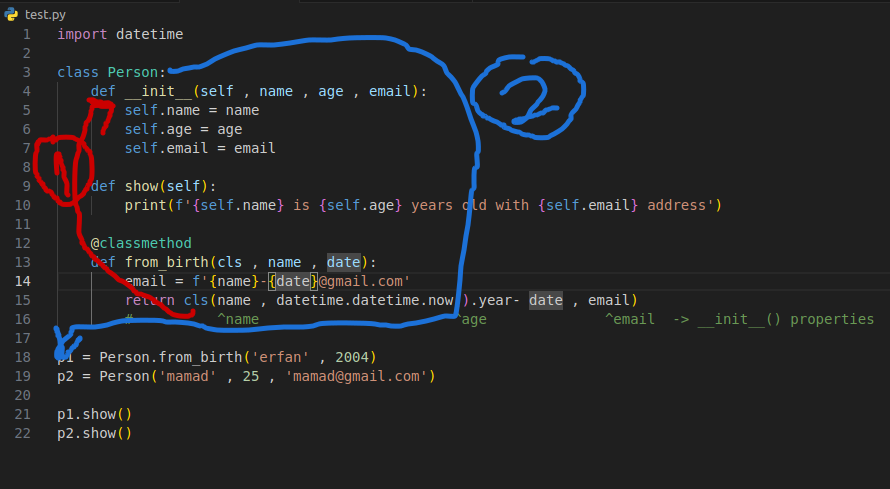
پس شد زمانی از کلاس متد استفاده میکنیم که بخاییم در درفتار کلاس تغییری ایجاد کنیم



توضیح دوباره عکس :

اول اومدیم با نوشتن @classmethod بالای متود from\_birth اون رو تبدیل به یک کلاس متد کردیم تا رفتار کلاس رو عوض کنیم .توی ارگومان های دیگه self ندادیم چون با ابچکت ها کاری نداریم . ما فقط موقع آبجکت سازی میخاییم بجای صدا زدن مسقتیم خود کلاس ایم متد رو صدا بزنیم تا ویژگی شو عوض کنیم . توی ارگومان ها دوست دارید همه ی ارگومان های داندر اینیت رو بگیرید یا دوست ندارید و میخاید بعضی هاشون رو از کاربر نگیرید و خودتون درست کنید مشکلی نداره . اینجا name و date رو میگیرم و بعد یک متغیر میسازم به اسم email و یک ایمیل با ترکیبی از اسم و تاریخ تولدش میسازم . بعد باید cls رو ریترن کنم ) تا کلاس صدا زده بشه و چون داریم ریترن میکنیم ریخته بشه توی p1 تا ابجکت ساخته بشه ( بعد موقع ساختن ابجکت همینجوری نمیایم کلاس رو صدا بزنم میایم بجای صدای مستقیم کلاس ) که موجب میشه داندر اینیتش صدا زده بشه ( اون کلاس متد رو صدا میزنم تا ویژگی های دستی رو کلاس اضافه کنم .

اون تیکش که باید cls ریترن بشه خیلی مهمه دیاگرامش اینشکلیه :



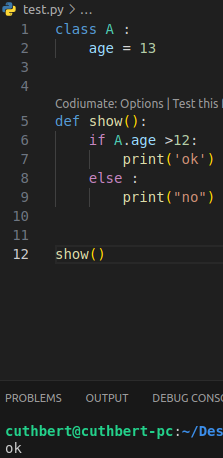
موقع ریترن شدن کلاس صدا زده میشه و داندر اینیتش صدا زده میشه و متغیر هایی که توی پرانتز cls دادم )خط پونزده( ارسال میشن به داندر اینیت و بعد از تشکیل ابجکت ریترن داده میشه به داخل p1بعد حالا هرکاری میخایی بکن با اون ابجکت

--------------------------------------------------

استاتیک متود ها :

این چیز خواصی نداره فقط باید بدونید که :

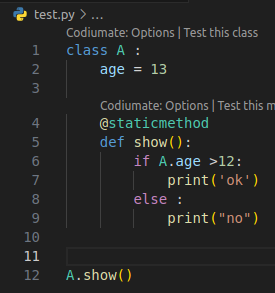
این کد رو ببینید:



این تابع خارج از کلاس هست و داره به خوبی کار میکنه ولی در برنامه نویسی پیشرفته وقتی یک

تابع به مقادیری از کلاس در ارتباطه به صورت غیر مستقیم ما میاییم اون تابع رو به صورت

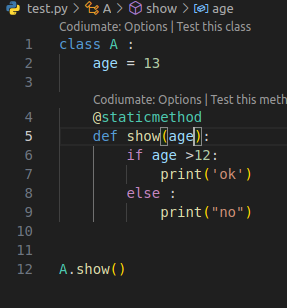
استاتیک توی خود کلاس مینویسیم تا کد مون قشنگ تر بشه یعنی چی?



ببینید فقط کافیه بالاش @staticmethod رو بزنید و بعد باهاش کار کنید

و برای صدا زدنش هم از داخل کلاس صداش میزنیم . وقتی توابع یه جوری نخشون وصله به کلاس

بهتره که داخل کلاس نوشته بشن . این کد رو اینجوری هم میشه نوشت :



چون دیگه داخل کلاس هست نمیخاد بنویسیم A.age همون age رو پاس بدیم به عنوان

ارگومان براش کافیه.

مفهوم پلیموریسمpolymorism چیست .

چیز خواصی نیست از قبل هم میدونستید فقط نه به این اسم

:



یه کلاس vehicle داریم که ۳ تا کلاس دیگه ازش ارث بری میکنن . در کلاس قایق و هواپیما تابع move به قول معروفی overwrite شده و در خروجی رفتاری غیر از رفتار والد خود رو نشون

میده. ولی کلاس ماشین توش تابع move اوررایت نشده و داره رفتار والد رو نشون میده . به این میگن رفتار پولیمورسیم . میتونیم حتی کاری کنیم که علاوه بر رفتار خودش رفتار والدش رو هم

نشون بده . چجور ؟ شما بگین

بیا ص بعد ببینم درست حدث زدی یا نه



اینجا باز اوررایت کردیم و داخل اوررایت شدع گفتیم super().move() یعنی برو از توی بابات تابع move شو بیار و. هم مال خودشو اجرا میکنه هم مال باباشو . البته فقط مال هواپیما و قایق همچین کار ی کردم.