

Time	Narration
00:01	به برنامه آموزشی Testing and debugging خوش آمدید.
00:06	در این برنامه می‌فهمید که software testing چیست
00:11	functions های ساده را برای functionality آن‌ها امتحان کنید.
00:15	Automate tests
00:17	درک نیاز به coding style و یادگیری برخی از standards که Python Community دنبال می‌کند.
00:25	برای ضبط این برنامه من از سیستم عامل
00:32	
00:35	Ubuntu Linux 16.04
	Python 3.4.3
	IPython 5.1.0
	و Gedit text editor استفاده می‌کنم.
00:42	برای تمرین این برنامه باید بدانید که چگونه از functions استفاده کنیم.
00:48	اگر نه برنامه آموزشی Python مربوطه در این وب سایت را ببینید.
00:53	ابتدا در مورد software testing یاد می‌گیریم.
00:57	Software testing فرایند ارزیابی عملکرد نرم افزار و یا یک program است.
01:04	این به پیدا کردن اینکه آیا برنامه الزامات مشخص شده را دارد یا نه کمک می‌کند.
01:10	این یک برنامه بدون نقص را تضمین می‌کند به طوری که ما یک برنامه با کیفیت دریافت کنیم.
01:16	تمام کدهای مورد استفاده در این آموزش در لینک Code Files این آموزش در دسترس هستند.
01:23	شما باید در working directory فعلی دانلود و استفاده کنید.
01:28	text editor را باز کنید و کد زیر را تایپ کنید.
01:33	این یک function ساده برای محاسبه gcd از دو عدد می‌باشد.
01:38	ما به مجموعه‌ای از inputs برای متغیر a و b احتیاج داریم.
01:43	فایل را به صورت gcd.py find underscore در working directory فعلی ذخیره کنید.
01:50	سپس فایل test underscore gcd.py را باز می‌کنیم.
01:56	مورد تست ما 48 و 64 برای a و b می‌باشد.
02:02	برای این مورد تست ما می‌دانیم که GCD 16 است. و خروجی طبق انتظار ماست.
02:11	حالا script را اجرا می‌کنیم و code خود را test می‌کنیم.
02:15	Terminal را باز کنید و python3 test underscore gcd.py را تایپ کنید.
02:24	ما output را Test Passed که یعنی code ما درست است بدست می‌آوریم.

02:30	اما می تواند تعدادی موارد که در آن gcd function ممکن است بشکند, وجود داشته باشد.
02:36	بنابراین ، برای بررسی جایی که کد مامکن است بشکند باید بسیاری از تستها اجرا شود.
02:42	اینجا جایی است که در آن مفهوم automating tests می آید.
02:46	ابتدا تست را در gcd function انجام و automate می کنیم.
02:51	فایل textcases.txt را که در آن پارامترهای مختلف تست داده شده است باز کنید.
02:58	ساختار فایل دو input parameters دارد.
03:03	parameter سوم output result درست می باشد. ما عناصر را با space جدا کرده ایم.
03:11	سپس فایل gcd.py test underscore gcd.py automate underscore test را باز می کنیم.
03:18	ابتدا به import gcd function از find underscore gcd برای استفاده در آزمایش نیاز داریم.
03:26	سپس فایل testcases.txt خط به خط خوانده می شود.
03:32	دو input parameters اولی به متغیرهای x و y اختصاص داده شده اند.
03:38	parameter سوم که مقدار output درست است به متغیر result اختصاص داده شده.
03:44	ما بررسی می کنیم که آیا مقدار بازگردانده شده توسط gcd function برابر است با مقدار در variable result .
03:52	بالاخره این پیام را مطابق پرینت می کند.
03:56	در ترمینال python3 automate underscore test underscore gcd.py را تایپ کنید.
04:05	همانطور که می بینید هر سه test cases در testcases.txt پاس شده اند.
04:12	مقدار محاسبه شده توسط gcd function برابر است با مقدار خروجی در testcases.txt
04:20	ویدیو را متوقف کنید. این تمرین را انجام دهید و به ویدیو برگردید.
04:26	برای همان ورودی بعنوان gcd شما automated tests برای LCM بنویسید.
04:33	از داده ها در فایل lcmtestcases.txt استفاده کنید.
04:38	برای حل کردن به ترمینال بروید.
04:42	کد برای محاسبه lcm از دو عدد را می بینیم. نام فایل find underscore lcm.py می باشد.
04:52	این فایل داده ها برای lcm test می باشد.
04:57	نام فایل lcmtestcases.txt می باشد.
05:02	توجه کنید که هر دو فایلها باید در working directory فعلی باشند.
05:07	حالا script را اجرا می کنیم و code خود را test می کنیم.
05:11	python3 find underscore lcm.py را تایپ کنید.
05:17	اینجا آزمایش سوم رد می شود. چون ورودی در lcmtestcases.txt اشتباه می باشد.
05:26	این برای بررسی رفتار برنامه در شرایط نادرست است.

05:32	یک program خوب باید readable باشد. بنابراین دیگران می توانند آن را گسترش و بهبود دهند.
05:39	Code بیشتر از اینکه نوشته شود، خوانده می شود.
05:43	ما یک نام را انتخاب می کنیم تا درک استفاده از آن آسان تر شود.
05:49	همانطور که در مثال می بینیم این بسیار ساده است که code چه کار می کند را بفهمیم.
05:55	نام مناسب برای فهمیدن کد بسیار کمک می کند.
06:00	همچنین باید موارد زیر را در هنگام نوشتن کد در پایتون به خاطر داشته باشید.
06:06	Four Space Indentation
06:09	محدودیت 79 کاراکتر در خط
06:13	Functions و methods باید با دو خط خالی جدا شوند.
06:18	برای مستند از بخشی از کد از Docstring استفاده کنید.
06:23	از whitespace در اطراف operators و بعد از punctuations استفاده کنید.
06:28	به پایان این برنامه می رسیم. خلاصه می کنیم.
06:34	در این برنامه یاد گرفتیم که tests های ساده برای function ایجاد کنیم.
06:40	Automate tests از بسیاری predefined test cases استفاده می کند و استفاده از python coding standards .
06:49	و تمرین برای شما:
06:53	indentation مناسب برای python code طبق دستورالعمل های style چیست؟
06:59	و پاسخ : چهار Space Indentation برای نوشتن یک کد پایتون با توجه به دستورالعمل های style مورد نیاز است.
07:08	لطفاً سؤالات کلی خود در مورد Python را به این انجمن بفرستید.
07:12	لطفاً سؤالات کلی خود در مورد Python را به این انجمن بفرستید. تیم FOSSEE پروژه TBC را هماهنگ می کند.
07:22	بودجه پروژه Spoken Tutorial توسط NMEICT, MHRD, دولت هند تأمین می شود. به این سایت مراجعه کنید.
07:33	ترجمه و صدا گذاری شبنم اقبال از IIT Bombay. با تشکر از شما