**Module 1: Hello Python! – Xin chào Python!**

|  |
| --- |
| In this module we’ll introduce you to the Coursera platform and the course format. Then, we’ll dive into the basics of programming languages and syntax, as well as automation using scripting. We’ll also introduce you to the Python programming... |

Trong mô-đun này, chúng tôi sẽ giới thiệu cho bạn nền tảng Coursera và định dạng khóa học. Sau đó, chúng ta sẽ đi sâu vào những kiến ​​thức cơ bản về ngôn ngữ lập trình và cú pháp, cũng như tự động hóa bằng cách sử dụng tập lệnh. Chúng tôi cũng sẽ giới thiệu cho bạn cách lập trình Python...

|  |
| --- |
| **Learning Objectives**   * Define the terms computer program, programming language, script, and automation * Use the print() function to output data to the screen * Explain the difference between the syntax and semantics of a programming language * List some of the characteristics of the Python language * Utilize basic Python arithmetic operators to obtain the results of mathematical expressions |

**Mục tiêu học tập**

* Xác định các thuật ngữ chương trình máy tính, ngôn ngữ lập trình, tập lệnh và tự động hóa
* Sử dụng hàm print() để xuất dữ liệu ra màn hình
* Giải thích sự khác biệt giữa cú pháp và ngữ nghĩa của ngôn ngữ lập trình
* Liệt kê một số đặc điểm của ngôn ngữ Python
* Sử dụng các toán tử số học Python cơ bản để thu được kết quả của các biểu thức toán học

**1. Course Introduction – Giới thiệu khóa học**

***1.1 Specialization Introduction – Giới thiệu chuyên ngành***

|  |
| --- |
| Working in IT is more than just a job. It's a career path. Research shows that the field of IT support is a launchpad for future career growth and better wages. In fact, a study on the subject was recently conducted by the Harvard Business School Accenture and Burning Glass entitled Bridge the Gap. It found that among today's middle-skilled jobs which require training but not a formal college degree, IT support offers clear pathways to prosperity. We saw this phenomenon play out here at Google, in our IT support program. Those who push themselves to learn how to code in Python typically saw strong career growth. They built skills that are critical to accessing higher level positions in the IT field, and after honing those skills through hard work and determination, they advanced into more technical IT support specialists, systems administrators, technical solutions engineers, and even site reliability engineers. The common thread across all of these roles is knowing how to write code to solve problems and automate solutions. |

Làm việc trong lĩnh vực CNTT không chỉ là một công việc.Đó là một con đường sự nghiệp. Nghiên cứu cho thấy lĩnh vực hỗ trợ CNTT là bệ phóngđể phát triển sự nghiệp trong tương lai và mức lương tốt hơn.Trên thực tế, một nghiên cứu về chủ đề này gần đây đã được thực hiện bởi HarvardTrường Kinh doanh Accenture và Burning Glass mang tên Bridge the Gap.Nó phát hiện ra rằng trong số những công việc đòi hỏi trình độ trung bình ngày nay đòi hỏiđào tạo nhưng không phải là bằng đại học chính thức, bộ phận hỗ trợ CNTT đưa ra lộ trình rõ ràngtới sự thịnh vượng. Chúng tôi đã thấy hiện tượng này diễn ra tại Google, trong bộ phận CNTT của chúng tôichương trình hỗ trợ. Những người thúc đẩy bản thân học cách viết mãPython thường chứng kiến ​​sự phát triển nghề nghiệp mạnh mẽ.Họ xây dựng những kỹ năng quan trọng để tiếp cận các vị trí cấp cao hơn tronglĩnh vực CNTT, và sau khi mài giũa những kỹ năng đó bằng sự chăm chỉ và quyết tâm,họ đã nâng cao thành các chuyên gia, hệ thống hỗ trợ CNTT kỹ thuật hơnquản trị viên, kỹ sư giải pháp kỹ thuật và thậm chí cả độ tin cậy của trang webKỹ sư. Điểm chung của tất cả các vai trò này là biết cách viếtmã để giải quyết vấn đề và tự động hóa các giải pháp.

|  |
| --- |
| By expanding your toolbox to include coding skills, you open a window into the world of systems management that can lead you towards more advanced technical roles down the line. Python in particular is having a huge surgence. |

Bằng cách mở rộng hộp công cụ của bạn để bao gồm các kỹ năng viết mã, bạn mở một cửa sổ vàothế giới quản lý hệ thống có thể đưa bạn tới những nơi tiên tiến hơnvai trò kỹ thuật ở tuyến dưới.Python nói riêng đang có sự phát triển vượt bậc.

|  |
| --- |
| According to the 2019 Stack Overflow Developer Survey, Python is the coding language most people want to learn. The second most loved by those who already know it and the fourth most popular overall. So why take this program to learn how to code in Python? Well, first it's geared towards people who are already in or aspiring to be in the field of IT. Maybe you're thinking bigger about your current IT role and want to work towards managing operations at scale, or maybe you're just starting out and looking to break into the IT industry. Perhaps you've taken our IT Support Professional Certificate program on Coursera already, or you have equivalent IT support knowledge with basic computing skills, like working with files and directories, familiarity with networking concepts, and understanding how to install software on your computer. In any case, this program is tailor-made for you. Second, this program offers three hands-on methods of teaching coding and Python and automation, code blocks, Jupyter notebooks, and Qwiklabs. Third, we've assembled an awesome group of Googlers who will serve as your instructors in each course. They all started their careers in IT support then learned programming and moved onto more technical roles like me. |

Theo Khảo sát nhà phát triển Stack Overflow năm 2019, Python là công cụ mã hóangôn ngữ mà hầu hết mọi người muốn học.Thứ hai được yêu thích nhất bởi những người đã biết nó và thứ tưphổ biến tổng thể. Vậy tại sao lại tham gia chương trình này để học cách viết mã bằng Python?Chà, đầu tiên nó hướng tới những người đã tham gia hoặc mong muốn tham gialĩnh vực CNTT. Có thể bạn đang suy nghĩ lớn hơn về vai trò CNTT hiện tại của mình vàmuốn hướng tới việc quản lý hoạt động trên quy mô lớn hoặc có thể bạn đangmới bắt đầu và đang tìm cách thâm nhập vào ngành CNTT.Có lẽ bạn đã tham gia chương trình Chứng chỉ Chuyên gia Hỗ trợ CNTT của chúng tôi trênĐã có Coursera hoặc bạn có kiến ​​thức hỗ trợ CNTT tương đương vớikỹ năng tính toán cơ bản, như làm việc với các tập tin và thư mục, làm quenvới các khái niệm về mạng và hiểu cách cài đặt phần mềm trênmáy tính của bạn. Trong mọi trường hợp, chương trình này được thiết kế riêng cho bạn.Second, this program offers three hands-on methods of teaching coding and Python and automation, code blocks, Jupyter notebooks, and Qwiklabs. Third, we've assembled an awesome group of Googlers who will serve as your instructors in each course. They all started their careers in IT support then learned programming and moved onto more technical roles like me.

|  |
| --- |
| We can't wait to share our stories with you on how we use Python in our day-to-day. Oh and hey, I should probably introduce myself. My name is Christine Rafla, and I'm a Systems Administrator at Google. I will be your instructor in this course. |

Chúng tôi rất nóng lòng được chia sẻ câu chuyện của mình với bạn về cách chúng tôi sử dụng Python trongngày qua ngày. Ồ và này, có lẽ tôi nên giới thiệu bản thân mình.Tên tôi là Christine Rafla và tôi là Quản trị viên hệ thống tại Google.Tôi sẽ là người hướng dẫn bạn trong khóa học này.

|  |
| --- |
| This program has been entirely designed and developed by Google and we even filmed each course at different cool Google spaces. It will introduce you to the Python programming language with a special focus on how this language applies to automating tasks in the world of IT systems support and administration. I'm super excited to be here with you. When I was younger, I had no idea that careers in IT even existed. There are a lot of reasons I wanted to participate in the certificate program. But one of my biggest motivations is that I want to see more women represented in the industry. I remember going to a System Administration Summit where there were hundreds of men and about three women that were sysadmins. A lot has changed since then but there's still so much we can do to bring new ideas and representation into the IT field. That's why I want to share my knowledge with as many people as possible. I love my job and I love the people I work with because they make it easy to ask for help and offer their guidance. This type of support network allows our team and ultimately our industry to be more successful. I understand from experience that it can feel pretty intimidating and maybe even a bit scary to learn a coding language. Just remember, everyone started where you are right now with the first command, the first script, and of course, the first of many errors. When I started out in my career, I strived to get everything perfectly right the first time I tried it. But that actually slowed down my progress. So don't be afraid to make mistakes, it will give you a leg up. So let's get down to it. What's ahead? The program begins with a crash course in Python where you will learn to write simple programs and understand their role in automation. Next, we'll get more hands-on focus on how Python interacts with the operating system. After that, we'll cover how to use Git and GitHub to manage versions of your code. Then we will focus on troubleshooting and debugging techniques to find and solve the root cause of problems in IT infrastructure. The next course covers automating at scale where you will learn to deploy configuration management on a fleet of either physical or virtual machines running in the Cloud. Last up, we will bring all this knowledge together and complete a final project designed to solve tasks that you might encounter in real-world IT settings. Bonus, you can post your projects to GitHub to show off your fancy new skills to employers or friends or both. |

Chương trình này hoàn toàn do Google thiết kế và phát triển và thậm chí chúng tôi cònđã quay từng khóa học tại các không gian thú vị khác nhau của Google.Nó sẽ giới thiệu cho bạn ngôn ngữ lập trình Python với mộttập trung vào cách ngôn ngữ này áp dụng để tự động hóa các tác vụ trong thế giới CNTThỗ trợ và quản trị hệ thống.Tôi rất vui mừng được ở đây với bạn.Khi tôi còn trẻ, tôi không hề biết rằng sự nghiệp trong lĩnh vực CNTT thậm chí còn tồn tại.Có rất nhiều lý do khiến tôi muốn tham gia chương trình cấp chứng chỉ.Nhưng một trong những động lực lớn nhất của tôi là tôi muốn gặp nhiều phụ nữ hơnđại diện trong ngành.Tôi nhớ mình đã tham dự Hội nghị Thượng đỉnh về Quản trị Hệ thống, nơi cóhàng trăm người đàn ông và khoảng ba phụ nữ là quản trị viên hệ thống.Đã có nhiều thay đổi kể từ đó nhưng vẫn còn rất nhiều điều chúng ta có thể làm để mang lạinhững ý tưởng và đại diện mới trong lĩnh vực CNTT.Đó là lý do tại sao tôi muốn chia sẻ kiến ​​thức của mình với càng nhiều người càng tốt.Tôi yêu công việc của mình và tôi yêu những người làm việc cùng tôi vì họ giúp tôi dễ dàng làm việc.yêu cầu giúp đỡ và đưa ra hướng dẫn của họ.Loại mạng lưới hỗ trợ này cho phép nhóm của chúng tôi và cuối cùng là ngành của chúng tôithành công hơn. Tôi hiểu từ kinh nghiệm rằng nó có thể cảm thấy kháđáng sợ và thậm chí có thể hơi đáng sợ khi học một ngôn ngữ mã hóa.Chỉ cần nhớ rằng mọi người đều bắt đầu như bạn hiện tại với lần đầu tiênlệnh, tập lệnh đầu tiên và tất nhiên là lỗi đầu tiên trong số rất nhiều lỗi.Khi mới bắt đầu sự nghiệp, tôi đã cố gắng hoàn thành mọi việc một cách hoàn hảođúng ngay lần đầu tiên tôi thử nó.Nhưng điều đó thực sự đã làm chậm tiến độ của tôi.Vì vậy, đừng sợ mắc sai lầm, nó sẽ giúp bạn tiến bộ hơn.Vì vậy, chúng ta hãy đi xuống nó.Phía trước là gì? Chương trình bắt đầu bằng một khóa học cấp tốc về Python, nơi bạn sẽhọc cách viết các chương trình đơn giản và hiểu vai trò của chúng trong tự động hóa.Tiếp theo, chúng ta sẽ tập trung thực hành hơn vào cách Python tương tác với hệ điều hànhhệ thống. Sau đó, chúng tôi sẽ đề cập đến cách sử dụng Git và GitHub để quản lý các phiên bản củama cua ban. Sau đó chúng ta sẽ tập trung vào các kỹ thuật khắc phục sự cố và gỡ lỗiđể tìm và giải quyết nguyên nhân gốc rễ của các vấn đề trong cơ sở hạ tầng CNTT.Khóa học tiếp theo đề cập đến việc tự động hóa trên quy mô lớn, nơi bạn sẽ học cách triển khaiquản lý cấu hình trên một nhóm máy vật lý hoặc ảochạy trên đám mây. Cuối cùng, chúng ta sẽ tập hợp tất cả những kiến ​​thức này lại với nhau vàhoàn thành dự án cuối cùng được thiết kế để giải quyết các nhiệm vụ mà bạn có thể gặp phải trongcài đặt CNTT trong thế giới thực.Ngoài ra, bạn có thể đăng dự án của mình lên GitHub để thể hiện những kỹ năng mới thú vị của mìnhcho người sử dụng lao động hoặc bạn bè hoặc cả hai.

|  |
| --- |
| That was a lot to rattle off. Are you excited or what? Now I'd like to quickly introduce you to my fellow instructors who you will get to meet along the way. Hey, my name is Roger Martinez. I'm a Linux System Administrator, and I'll be your instructor in the course on using Python's interacts with an operating system. Hi. I'm Kenny Sulma, and I'll be your guide in the course about using Git and GitHub to manage versions of your code. Hi there. I'm Amanda Ballas, and I'll be teaching you about troubleshooting and debugging. Hey, I'm Phelan Vendeville, and in my course, we'll learn about Automation at scale using Configuration Management and the Cloud. Thanks everyone. This all-star team was brought together to guide you on your adventures in coding. You're in very good hands. Okay, I think that's everything. Let's get ready to learn some new skills and maybe even have some laughs along the way. I'll see you in the next video. |

Đó là rất nhiều điều để nói ra.Bạn có hào hứng hay không?Bây giờ tôi muốn nhanh chóng giới thiệu bạn với những người hướng dẫn đồng nghiệp của tôi, những người mà bạn sẽgặp nhau trên đường đi. Xin chào, tên tôi là Roger Martinez.Tôi là Quản trị viên hệ thống Linux và tôi sẽ là người hướng dẫn bạn trong khóa học vềsử dụng Python tương tác với hệ điều hành.CHÀO. Tôi là Kenny Sulma và tôi sẽ là người hướng dẫn bạn trong khóa học về cách sử dụng Git vàGitHub để quản lý các phiên bản mã của bạn.Chào bạn. Tôi là Amanda Ballas và tôi sẽ hướng dẫn bạn cách khắc phục sự cốvà gỡ lỗi. Này, tôi là Phelan Vendeville, và trong khóa học của tôi, chúng ta sẽtìm hiểu về Tự động hóa trên quy mô lớn bằng cách sử dụng Quản lý cấu hình và Đám mây.Cảm ơn mọi người. Đội ngũ toàn sao này đã được tập hợp lại để hướng dẫn bạn trên con đường của mìnhcuộc phiêu lưu trong mã hóa. Bạn đang ở trong tay rất tốt.Được rồi, tôi nghĩ đó là tất cả.Hãy sẵn sàng học một số kỹ năng mới và thậm chí có thể cười sảng khoáidọc đường. Tôi sẽ gặp bạn trong video tiếp theo.

***1.2. Course Introduction – Giới thiệu khóa học***

|  |
| --- |
| If you work in IT, computer programming skills open up an incredible amount of opportunity. Being able to write scripts and programs that tell your computer to perform a task equips you with an invaluable tool. Not only does it make your work easier and more efficient, it can help you grow faster and advance further in your IT career. But how do you even start to learn a programming language like Python? How do you recognize when to tell a computer to perform a task? And how do you then write a program to actually get your computer to complete the task you want it to do? The thought of learning to write a program in Python can make you feel a whole bunch of emotions, excitement, anticipation, that feeling of wanting to dive right in and get going and also fear. You might ask yourself, can I really learn how to code or do I have it in me? I'm here to tell you, yes, you can absolutely do this. Learning how to program can be scary and intimidating, but at the same time it's really fun and really exciting. In coding as in life, if we're going to get philosophical, the most rewarding work is usually a bit challenging, but ultimately well worth the effort. Of course, I'm able to say all this from experience, especially the cheesy parts. My name is Christine Rafla. I'm a systems administrator at Google and I'm going to be your instructor and guide in this course. The role of a sysadmin can vary a lot from company to company and even within different teams in the same company. I happen to work in the corporate identity and access management operations team, which is a really long way of saying that we make sure that everyone is represented correctly and if they need to access certain resources, they can. What I love the most about being a sysadmin is that the role has so many diverse functions. We handle loads of unique problems and edge cases from tinkering with different systems to collaborating with other teams. I am always learning something new, so it's really hard to get bored. It all starts with knowing how to automate, if you're an IT support specialist, a systems administrator, or in a role somewhere in between knowing how to get computers to do the hard work for you will set you apart from others in similar IT roles and make your life much easier. Think about it, would you rather manually deploy 100 computers on your own or tell your computer to do it all for you all at once? No-brainer, right, having a coding skill set can help you grow into more specialized roles like a systems administrator, Cloud Solutions engineer, DevOps specialist, site reliability engineer, or who knows maybe even web developer or data analyst. The point is, being able to write a program is an essential tool in your IT toolkit and more and more employers are looking for these skills in the people they hire. If you've ever learned a new skill, like playing a musical instrument, speaking a foreign language, knitting, or skateboarding. You know that getting good at something new requires a lot of practice. For me, I love to learn new languages and I'm proud to say I speak Spanish, Arabic, French, and I even know ten words in Russian. Our world is shaped by the words and the languages we speak and while some words may be unique to one language you can always find similarities that help you learn and understand. Being able to connect the dots between cultures allows me to see things others might not, kind of sounds like this applies to IT programming, huh? My point is, whether you're learning French or Python, it's never easy. You have to start small, learn the basics and practice those until you master them. Only then can you move on to more complex and impressive stuff and I'm here to help you along that path along with my colleagues who you'll meet in later courses. We'll start slow, master the foundations together and you'll soon be ready for more challenging stuff. So are you wondering why we filmed this course in a cabin on a lake in Canada? The truth is we're actually in a game room at one of the Google offices in Sunnyvale, California. We chose a different themed office space for each course of the program just to mix things up and I think I scored with this one. I should warn my manager that I'll be hanging out in this one well after the course ends because it's super comfy. By the end of this course, you'll understand the benefits of programming in IT roles. You'll be able to write simple programs using Python, figure out how the building blocks of programming fit together, and combine all this knowledge to solve a complex programming problem. That's right, by the end of this course, you're going to write a program in Python that's designed to solve a real-world IT problem, super exciting right? We'll start off by diving into the basics of writing a computer program. You'll get hands-on experience with programming concepts through interactive exercises and real-world examples. |

Nếu bạn làm việc trong lĩnh vực CNTT, kỹ năng lập trình máy tính sẽ mở ra vô số cơ hộicơ hội. Có khả năng viết các kịch bản và chương trình cho bạn biếtmáy tính để thực hiện một nhiệm vụ sẽ trang bị cho bạn một công cụ vô giá.Nó không chỉ giúp công việc của bạn dễ dàng và hiệu quả hơn mà còn có thể giúp bạn phát triểnnhanh hơn và thăng tiến hơn nữa trong sự nghiệp CNTT của bạn.Nhưng làm thế nào để bạn bắt đầu học một ngôn ngữ lập trình như Python?Làm thế nào để bạn nhận biết khi nào cần yêu cầu máy tính thực hiện một tác vụ?Và sau đó bạn viết một chương trình như thế nào để máy tính của bạn thực sự hoàn thànhnhiệm vụ bạn muốn nó làm?Ý nghĩ học viết chương trình bằng Python có thể khiến bạn cảm thấycả đống cảm xúc, phấn khích, mong chờ, cảm giác muốnlao vào và bắt đầu và cũng sợ hãi.Bạn có thể tự hỏi mình, liệu tôi có thể thực sự học cách viết mã hay tôi đã có sẵn nó rồi?Tôi ở đây để nói với bạn rằng, vâng, bạn hoàn toàn có thể làm được điều này.Học cách lập trình có thể đáng sợ và đáng sợ, nhưng đồng thờinó thực sự thú vị và thực sự thú vị.Trong lập trình cũng như trong cuộc sống, nếu chúng ta muốn hiểu triết học, điều bổ ích nhấtcông việc thường có một chút thử thách nhưng cuối cùng cũng đáng nỗ lực.Tất nhiên, tôi có thể nói tất cả những điều này từ kinh nghiệm, đặc biệt là những câu sến súa.các bộ phận. Tên tôi là Christine Rafla.Tôi là quản trị viên hệ thống của Google và tôi sẽ là người hướng dẫn bạn vàhướng dẫn trong khóa học này.Vai trò của quản trị viên hệ thống có thể khác nhau rất nhiều tùy theo từng công ty và thậm chí trong nội bộ công ty.các nhóm khác nhau trong cùng một công ty.Tôi tình cờ làm việc trong lĩnh vực quản lý quyền truy cập và nhận dạng công tynhóm vận hành, đó là một cách thực sự dài để nói rằng chúng tôi đảm bảo rằngmọi người đều được thể hiện chính xác và liệu họ có cần truy cập một số quyền truy cập nhất định hay khôngnguồn lực, họ có thể.Điều tôi thích nhất khi trở thành quản trị viên hệ thống là vai trò này có rất nhiềuchức năng đa dạng.Chúng tôi xử lý vô số vấn đề đặc biệt và các trường hợp khó khăn từ việc mày mò các phương pháp khác nhau.hệ thống để cộng tác với các nhóm khác.Tôi luôn học hỏi điều gì đó mới mẻ nên thực sự rất khó để cảm thấy nhàm chán.Tất cả đều bắt đầu từ việc biết cách tự động hóa, nếu bạn là nhân viên hỗ trợ CNTTchuyên gia, quản trị viên hệ thống hoặc ở một vai trò nào đó ở giữa hiểu biếtcách để máy tính thực hiện công việc khó khăn cho bạn sẽ khiến bạn khác biệt vớinhững người khác có vai trò CNTT tương tự và giúp cuộc sống của bạn dễ dàng hơn nhiều.Hãy suy nghĩ về điều này, bạn có muốn triển khai thủ công 100 máy tính trên máy tính của mình không?sở hữu hay bảo máy tính của bạn làm tất cả những việc đó cho bạn cùng một lúc?Không cần bàn cãi, đúng vậy, có bộ kỹ năng viết mã có thể giúp bạn phát triển hơncác vai trò chuyên biệt như quản trị viên hệ thống, kỹ sư Giải pháp đám mây,Chuyên gia DevOps, kỹ sư về độ tin cậy của trang web hoặc thậm chí có thể biết về webnhà phát triển hoặc nhà phân tích dữ liệu.Vấn đề là, khả năng viết chương trình là một công cụ thiết yếu trong hệ thống CNTT của bạn.bộ công cụ và ngày càng có nhiều nhà tuyển dụng đang tìm kiếm những kỹ năng này ở con ngườihọ thuê. Nếu bạn đã từng học một kỹ năng mới, chẳng hạn như chơi nhạcnhạc cụ, nói ngoại ngữ, đan lát hoặc trượt ván.Bạn biết rằng để giỏi một thứ gì đó mới đòi hỏi phải luyện tập rất nhiều.Đối với tôi, tôi thích học ngôn ngữ mới và tôi tự hào nói rằng tôi nói được tiếng Tây Ban Nha,Tiếng Ả Rập, tiếng Pháp và tôi thậm chí còn biết đến 10 từ tiếng Nga.Thế giới của chúng ta được định hình bởi những từ ngữ và ngôn ngữ chúng ta nói và trong khi một sốcác từ có thể là duy nhất cho một ngôn ngữ, bạn luôn có thể tìm thấy những điểm tương đồng có íchbạn học và hiểu.Khả năng kết nối các điểm giữa các nền văn hóa cho phép tôi nhìn thấy những thứ kháccó thể không, những âm thanh như thế này áp dụng cho lập trình CNTT phải không?Quan điểm của tôi là, cho dù bạn đang học tiếng Pháp hay Python, điều đó không bao giờ dễ dàng.Bạn phải bắt đầu từ việc nhỏ, học những điều cơ bản và thực hành chúng cho đến khi bạnlàm chủ chúng. Chỉ khi đó bạn mới có thể chuyển sang những thứ phức tạp và ấn tượng hơn vàTôi ở đây để giúp bạn trên con đường đó cùng với những đồng nghiệp của tôi mà bạn sẽ gặptrong các khóa học sau này. Chúng ta sẽ bắt đầu chậm rãi, cùng nhau nắm vững nền tảng vàbạn sẽ sớm sẵn sàng cho những điều khó khăn hơn.Vậy bạn có thắc mắc tại sao chúng tôi lại quay khóa học này trong một cabin trên hồ ở Canada không?Sự thật là chúng tôi đang ở trong phòng trò chơi tại một trong những văn phòng của Google ởSunnyvale, California.Chúng tôi đã chọn không gian văn phòng theo chủ đề khác nhau cho mỗi khóa học của chương trìnhchỉ để trộn lẫn mọi thứ và tôi nghĩ tôi đã ghi điểm với điều này.Tôi nên cảnh báo người quản lý của mình rằng tôi sẽ đi chơi ở nơi này saukhóa học kết thúc vì nó siêu thoải mái.Đến cuối khóa học này, bạn sẽ hiểu được lợi ích của việc lập trìnhtrong vai trò CNTT. Bạn sẽ có thể viết các chương trình đơn giản bằng Python, hãy tìm hiểucách các khối xây dựng của chương trình khớp với nhau và kết hợp tất cả những điều nàykiến thức để giải quyết một vấn đề lập trình phức tạp.Đúng vậy, khi kết thúc khóa học này, bạn sẽ viết một chương trìnhbằng Python được thiết kế để giải quyết vấn đề CNTT trong thế giới thực, cực kỳ thú vịPhải? Chúng ta sẽ bắt đầu bằng cách đi sâu vào những kiến ​​thức cơ bản về viết máy tínhchương trình. Bạn sẽ có được trải nghiệm thực tế với các khái niệm lập trình thông quabài tập tương tác và ví dụ thực tế.

|  |
| --- |
| You'll quickly start to see how computers can perform a multitude of tasks. You just have to write code that tells them what to do. Along the way, we'll be talking about automation, which is the process of getting computers to automatically do a task that us humans normally have to do by hand. Now, some of the stuff can get a little complex and confusing. I promise to do my best to make these lessons clear and easy to understand, but if you get stuck at any point, please feel free to re-watch the videos. Practice as much as you like and take the time you really need to understand these topics. The goal of this course isn't to teach you everything there is to know about software engineering because yikes, that would be a long course. Instead, we're going to introduce you to some of the key concepts of programming and scripting that will empower you to spot opportunities for automation in real life. |

Bạn sẽ nhanh chóng thấy được máy tính có thể thực hiện vô số công việc như thế nào.nhiệm vụ. Bạn chỉ cần viết mã cho họ biết phải làm gì.Trong phần này, chúng ta sẽ nói về tự động hóa, đó là quá trìnhkhiến máy tính tự động thực hiện một nhiệm vụ mà con người chúng ta thường phải làmbằng tay. Bây giờ, một số nội dung có thể hơi phức tạp và khó hiểu.Tôi hứa sẽ cố gắng hết sức để làm cho những bài học này trở nên rõ ràng và dễ hiểu,nhưng nếu bạn gặp khó khăn ở bất kỳ điểm nào, vui lòng xem lại video.Thực hành bao nhiêu tùy thích và dành thời gian bạn thực sự cần để hiểunhững chủ đề này. Mục tiêu của khóa học này không phải là dạy cho bạn mọi thứ cóđể biết về công nghệ phần mềm vì rất tiếc, đó sẽ là một chặng đường dàikhóa học. Thay vào đó, chúng tôi sẽ giới thiệu cho bạn một số chìa khóakhái niệm về lập trình và viết kịch bản sẽ cho phép bạn phát hiệncơ hội tự động hóa trong cuộc sống thực.

|  |
| --- |
| You're about to learn a skill that can help you take your career to whole new levels. Are you excited? I'm excited, so let's jump in! |

Bạn sắp học một kỹ năng có thể giúp bạn đưa sự nghiệp của mình lên một tầm cao mớicấp độ. Bạn có phấn khích không?Tôi rất phấn khích, vì vậy hãy nhảy vào!

***1.3. Program Surveys – Khảo sát chương trình***

|  |
| --- |
| **Program Surveys** |

**Khảo sát chương trình**

|  |
| --- |
| During this certificate program, you will be asked to complete a few short surveys. These are part of an important research study to understand how effective the certificate is for participants like you. Make sure to check out below for a rundown of what each survey will cover. |

Trong chương trình chứng chỉ này, bạn sẽ được yêu cầu hoàn thành một số khảo sát ngắn. Đây là một phần của nghiên cứu quan trọng nhằm tìm hiểu mức độ hiệu quả của chứng chỉ đối với những người tham gia như bạn. Hãy nhớ xem bên dưới để biết tóm tắt về nội dung của mỗi cuộc khảo sát.

|  |
| --- |
| Heads up: Your survey participation is optional but strongly encouraged. There are no correct answers, and your responses and personal data:   * will not affect your course experience, scores, or ability to receive a certificate or job in any way * will be kept confidential, with your name separated from your data * will not be shared outside of our research team except where you give permission to share contact information with hiring partners. |

Lưu ý: Việc tham gia khảo sát của bạn là tùy chọn nhưng được khuyến khích mạnh mẽ. Không có câu trả lời chính xác và câu trả lời cũng như dữ liệu cá nhân của bạn:

* sẽ không ảnh hưởng đến trải nghiệm khóa học, điểm số hoặc khả năng nhận chứng chỉ hoặc công việc của bạn dưới bất kỳ hình thức nào
* sẽ được giữ bí mật, tên của bạn được tách biệt khỏi dữ liệu của bạn
* sẽ không được chia sẻ ra bên ngoài nhóm nghiên cứu của chúng tôi trừ khi bạn cho phép chia sẻ thông tin liên hệ với các đối tác tuyển dụng.

|  |
| --- |
| Thanks for your consideration and time! It is only by hearing from learners like you that our programs can be successful. |

Cảm ơn sự quan tâm và thời gian của bạn! Chỉ khi được lắng nghe ý kiến ​​từ những người học như bạn thì chương trình của chúng tôi mới có thể thành công.

|  |
| --- |
| **Individual Course Feedback** |

**Phản hồi khóa học cá nhân**

|  |
| --- |
| When you **complete the last graded assignment** **within an individual course**, you may be asked to answer a survey that revisits earlier questions and also asks what you have learned up to that point in the program. |

Khi bạn **hoàn thành bài tập được chấm điểm cuối cùng trong một khóa học riêng lẻ** , bạn có thể được yêu cầu trả lời một bản khảo sát xem lại các câu hỏi trước đó và cũng hỏi bạn đã học được gì cho đến thời điểm đó trong chương trình.

|  |
| --- |
| **Entry survey** |

**Khảo sát đầu vào**

|  |
| --- |
| In the next course item, you'll have the opportunity to fill out a brief survey to help us understand why you’ve enrolled in this class. If you do not fill it out, you'll receive an invitation to fill out the survey after completing one lecture or assignment. |

Trong mục khóa học tiếp theo, bạn sẽ có cơ hội điền vào một bản khảo sát ngắn để giúp chúng tôi hiểu lý do bạn đăng ký tham gia lớp học này. Nếu không điền, bạn sẽ nhận được lời mời điền vào bản khảo sát sau khi hoàn thành một bài giảng hoặc bài tập.

|  |
| --- |
| We will ask about your experiences leading up to this course, and what you hope to accomplish. This is critical information for making sure we can meet the needs of learners like you, and can offer this program in the future. |

Chúng tôi sẽ hỏi về trải nghiệm của bạn khi tham gia khóa học này và những gì bạn hy vọng đạt được. Đây là thông tin quan trọng để đảm bảo chúng tôi có thể đáp ứng nhu cầu của những người học như bạn và có thể cung cấp chương trình này trong tương lai.

|  |
| --- |
| **Certificate Completion Survey** |

**Khảo sát hoàn thành chứng chỉ**

|  |
| --- |
| After you **complete the last graded assignment in the final course** of the certificate, you will be asked to answer a survey that revisits earlier questions. |

Sau khi **hoàn thành bài tập được chấm điểm cuối cùng trong khóa học cuối cùng** của chứng chỉ, bạn sẽ được yêu cầu trả lời một bản khảo sát xem lại các câu hỏi trước đó.

|  |
| --- |
| It also asks about what you have learned, and if you want to share your contact information with prospective employers. |

Nó cũng hỏi về những gì bạn đã học được và liệu bạn có muốn chia sẻ thông tin liên hệ của mình với các nhà tuyển dụng tiềm năng hay không.

***1.4. Google Cert Participant Entry Survey – Khảo sát về người tham gia tham gia Google Cert***

***1.5. Take a Minute to Set Yourself up for Success – Dành một phút để chuẩn bị cho thành công***

***1.6. Welcome to the Program – Chào mừng đến với Chương trình***

|  |
| --- |
| **Welcome to the Program** |

**Chào mừng đến với Chương trình**

|  |
| --- |
| The Google IT Automation with Python Certificate program is designed to teach you the foundations of programming in Python. We’re excited to join you on this journey as you learn one of the most-in-demand job skills in IT today. In the U.S. alone, according to Indeed data from November 2023, there were ~818K job openings in 2023 asking for Python skills. |

Chương trình Tự động hóa CNTT của Google có Chứng chỉ Python được thiết kế để dạy cho bạn kiến ​​thức cơ bản về lập trình bằng Python. Chúng tôi rất vui được tham gia cùng bạn trong hành trình này khi bạn học một trong những kỹ năng công việc được yêu cầu cao nhất trong lĩnh vực CNTT hiện nay. Chỉ riêng ở Hoa Kỳ, theo dữ liệu của Indeed từ tháng 11 năm 2023, đã có ~ 818 nghìn cơ hội việc làm vào năm 2023 yêu cầu kỹ năng Python.

|  |
| --- |
| **Certificate prerequisites** |

**Điều kiện tiên quyết về chứng chỉ**

|  |
| --- |
| This certificate program requires no previous knowledge of programming. Familiarity with basic IT concepts, like operating systems, files and processes, networking and data management will be required in further courses. For learners with no IT background at all, we recommend taking the [Google IT Support Professional Certificate](https://www.coursera.org/specializations/google-it-support). |

Chương trình chứng chỉ này không yêu cầu kiến ​​thức lập trình trước đó. Các khóa học tiếp theo sẽ yêu cầu làm quen với các khái niệm CNTT cơ bản, như hệ điều hành, tệp và quy trình, mạng và quản lý dữ liệu. Đối với những người học hoàn toàn không có nền tảng về CNTT, chúng tôi khuyên bạn nên tham gia khóa học[Chứng chỉ chuyên môn hỗ trợ CNTT của Google](https://www.coursera.org/specializations/google-it-support).

|  |
| --- |
| **Program overview** |

**Tổng quan chương trình**

|  |
| --- |
| This program is composed of six courses: |

Chương trình này bao gồm sáu khóa học:



|  |
| --- |
| 1. **Crash Course on Python** (*this course)* 2. [Using Python to Interact with the Operating System](https://www.coursera.org/learn/python-operating-system/home/welcome) 3. [Introduction to Git and GitHub](https://www.coursera.org/learn/introduction-git-github/home/welcome) 4. [Troubleshooting and Debugging Techniques](https://www.coursera.org/learn/troubleshooting-debugging-techniques/home/welcome) 5. [Configuration Management and the Cloud](https://www.coursera.org/learn/configuration-management-cloud/home/welcome) 6. [Automating Real-World Tasks with Python](https://www.coursera.org/learn/automating-real-world-tasks-python/home/welcome) |

1. **Khóa học cấp tốc về Python** ( *khóa học này)*
2. [Sử dụng Python để tương tác với hệ điều hành](https://www.coursera.org/learn/python-operating-system/home/welcome)
3. [Giới thiệu về Git và GitHub](https://www.coursera.org/learn/introduction-git-github/home/welcome)
4. [Kỹ thuật khắc phục sự cố và gỡ lỗi](https://www.coursera.org/learn/troubleshooting-debugging-techniques/home/welcome)
5. [Quản lý cấu hình và đám mây](https://www.coursera.org/learn/configuration-management-cloud/home/welcome)
6. [Tự động hóa các tác vụ trong thế giới thực bằng Python](https://www.coursera.org/learn/automating-real-world-tasks-python/home/welcome)

|  |
| --- |
| Crash Course on Python is the first course in this program. |

Khóa học cấp tốc về Python là khóa học đầu tiên trong chương trình này.



|  |
| --- |
| **How to pass the certificate** |

**Làm thế nào để vượt qua chứng chỉ**

|  |
| --- |
| You can review videos, readings, discussion forums, in-video questions, and practice quizzes in the program for free. However, to access graded assignments and be eligible to receive your official Google IT Automation with Python Certificate, you must:   * Pay the [course certificate fee](https://www.coursera.support/s/article/209818963-Payments-on-Coursera?language=en_US), or apply and be approved for [Coursera Financial Aid](https://www.coursera.support/s/article/209819033-Apply-for-Financial-Aid-or-a-Scholarship?language=en_US).   AND   * Pass all graded assignments in the six courses at the minimum passing level, or above. Each graded assignment in a course is part of a cumulative grade for that course. The passing score for each course is 80%. |

Bạn có thể xem lại video, bài đọc, diễn đàn thảo luận, câu hỏi trong video và câu hỏi thực hành trong chương trình miễn phí. Tuy nhiên, để truy cập vào các bài tập đã chấm điểm và đủ điều kiện nhận Chứng chỉ Python chính thức của Google IT Automation, bạn phải:

* Tra[lệ phí cấp chứng chỉ khóa học](https://www.coursera.support/s/article/209818963-Payments-on-Coursera?language=en_US), hoặc nộp đơn và được phê duyệt[Hỗ trợ tài chính Coursera](https://www.coursera.support/s/article/209819033-Apply-for-Financial-Aid-or-a-Scholarship?language=en_US).

VÀ

* Đạt tất cả các bài tập được chấm điểm trong sáu khóa học ở mức đậu tối thiểu hoặc cao hơn. Mỗi bài tập được chấm điểm trong một khóa học là một phần của điểm tích lũy cho khóa học đó. Điểm đậu mỗi môn là 80%.

|  |
| --- |
| **How deadlines work** |

**Thời hạn hoạt động như thế nào**

|  |
| --- |
| When you enroll in the course, the system automatically sets a deadline for when you need to complete each section. Heads up: These deadlines are there to help you organize your time, but you can take the course at your own pace. If you "miss" a deadline, you can just reset it to a new date. There’s no time limit in which you have to finish the course, and you can earn the certificate whenever you finish. |

Khi bạn đăng ký khóa học, hệ thống sẽ tự động đặt ra thời hạn bạn cần hoàn thành từng phần. Lưu ý: Những thời hạn này nhằm giúp bạn sắp xếp thời gian nhưng bạn có thể tham gia khóa học theo tốc độ của riêng mình. Nếu bạn "bỏ lỡ" thời hạn, bạn có thể đặt lại thời hạn đó sang một ngày mới. Không có giới hạn thời gian mà bạn phải hoàn thành khóa học và bạn có thể nhận được chứng chỉ bất cứ khi nào bạn hoàn thành.

|  |
| --- |
| **Getting and giving help** |

**Nhận và giúp đỡ**

|  |
| --- |
| Here are a few ways you can give and get help:   1. Discussion forums: You can share information and ideas with your fellow learners in the discussion forums. These are also great places to find answers to questions you may have. If you're stuck on a concept, are struggling to solve a practice exercise, or you just want more information on a subject, the discussion forums are there to help you move forward. 2. Study together: It can be hard to find the time to progress through a Google Career Certificate with all life’s distractions. [Register for Study Group](https://lu.ma/u/Google-Career-Certificates-Study-Group) – live study sessions hosted Monday through Thursday that offer you dedicated time and space to focus on your coursework alongside a community of your peers. 3. Coursera learner support: Use the [Learner Help Center](https://learner.coursera.help/hc/en-us) to find information on specific technical issues. These include error messages, difficulty submitting assignments, or problems with video playback. If you can’t find an answer in the documentation, you can also report your problem to the Coursera support team by clicking on the Contact Us! link available at the bottom of help center articles. 4. Course content issues: You can also flag problems in course materials by rating them. When you rate course materials, the instructor will see your ratings and feedback; other learners won’t. To rate course materials:  * Open the course material you want to rate. You can only rate videos, readings, and quizzes. * If the item was interesting or helped you learn, click the thumbs-up icon. * If the item was unhelpful or confusing, click the thumbs-down icon. |

Dưới đây là một số cách bạn có thể cho và nhận trợ giúp:

1. Diễn đàn thảo luận: Bạn có thể chia sẻ thông tin và ý tưởng với những người cùng học trong diễn đàn thảo luận. Đây cũng là nơi tuyệt vời để tìm câu trả lời cho các câu hỏi mà bạn có thể có. Nếu bạn bị mắc kẹt ở một khái niệm nào đó, đang gặp khó khăn khi giải một bài tập thực hành hoặc bạn chỉ muốn biết thêm thông tin về một chủ đề thì các diễn đàn thảo luận luôn sẵn sàng giúp bạn tiến về phía trước.
2. Cùng nhau học tập: Thật khó để tìm được thời gian để đạt được Chứng chỉ nghề nghiệp của Google với tất cả những phiền nhiễu trong cuộc sống.[Đăng ký nhóm học tập](https://lu.ma/u/Google-Career-Certificates-Study-Group)- các buổi học trực tiếp được tổ chức từ Thứ Hai đến Thứ Năm mang đến cho bạn thời gian và không gian dành riêng để tập trung vào bài tập của mình cùng với cộng đồng bạn bè cùng trang lứa.
3. Hỗ trợ người học Coursera: Sử dụng[Trung tâm trợ giúp người học](https://learner.coursera.help/hc/en-us)để tìm thông tin về các vấn đề kỹ thuật cụ thể. Chúng bao gồm các thông báo lỗi, khó gửi bài tập hoặc sự cố khi phát lại video. Nếu bạn không thể tìm thấy câu trả lời trong tài liệu, bạn cũng có thể báo cáo vấn đề của mình với nhóm hỗ trợ Coursera bằng cách nhấp vào liên hệ với chúng tôi! liên kết có sẵn ở cuối bài viết trong trung tâm trợ giúp.
4. Các vấn đề về nội dung khóa học: Bạn cũng có thể gắn cờ các vấn đề trong tài liệu khóa học bằng cách xếp hạng chúng. Khi bạn đánh giá tài liệu khóa học, người hướng dẫn sẽ thấy xếp hạng và phản hồi của bạn; những người học khác thì không. Để đánh giá tài liệu khóa học:

* Mở tài liệu khóa học bạn muốn xếp hạng. Bạn chỉ có thể xếp hạng video, bài đọc và câu hỏi.
* Nếu mục này thú vị hoặc giúp bạn tìm hiểu, hãy nhấp vào biểu tượng thích.
* Nếu mục này không hữu ích hoặc gây nhầm lẫn, hãy nhấp vào biểu tượng không thích.

|  |
| --- |
| **Finding out more information** |

**Tìm hiểu thêm thông tin**

|  |
| --- |
| Throughout this course, we'll be teaching you the basics of programming and automation. We'll provide a lot of information through videos and supplemental readings. But sometimes, you may also need to look things up on your own, now and throughout your career. Things change fast in IT, so it’s critical to do your own research so you stay up-to-date on what’s new. We recommend you use your favorite search engine to find additional information— it’s great practice for the real world! |

Trong suốt khóa học này, chúng tôi sẽ dạy bạn những kiến ​​thức cơ bản về lập trình và tự động hóa. Chúng tôi sẽ cung cấp nhiều thông tin thông qua video và bài đọc bổ sung. Nhưng đôi khi, bạn cũng có thể cần phải tự mình tìm kiếm mọi thứ ngay bây giờ và trong suốt sự nghiệp của mình. Mọi thứ thay đổi nhanh chóng trong lĩnh vực CNTT, vì vậy điều quan trọng là bạn phải tự nghiên cứu để luôn cập nhật những thông tin mới. Chúng tôi khuyên bạn nên sử dụng công cụ tìm kiếm yêu thích của mình để tìm thêm thông tin— đó là cách thực hành tuyệt vời cho thế giới thực!

|  |
| --- |
| On top of search results, here are some good programming resources available online:   * The official [Python tutorial](https://docs.python.org/3/tutorial/index.html). This tutorial is designed to help people teach themselves Python. While it goes in a different order than the one we're taking here, it covers a lot of the same subjects that we explore in this course. You can refer to this resource for extra information on these subjects. * The [Think Python](https://greenteapress.com/wp/think-python-2e/) book. This book aims to teach people how to program in Python. It's available online in PDF and browsable forms. Again, you can use this resource to learn more about some of the subjects we cover. |

Ngoài kết quả tìm kiếm, đây là một số tài nguyên lập trình tốt có sẵn trực tuyến:

* Chính thức[Hướng dẫn Python](https://docs.python.org/3/tutorial/index.html). Hướng dẫn này được thiết kế để giúp mọi người tự học Python. Mặc dù nó diễn ra theo thứ tự khác với thứ tự chúng ta đang trình bày ở đây nhưng nó bao gồm rất nhiều chủ đề tương tự mà chúng ta khám phá trong khóa học này. Bạn có thể tham khảo tài nguyên này để biết thêm thông tin về các chủ đề này.
* Các[Nghĩ Python](https://greenteapress.com/wp/think-python-2e/)sách. Cuốn sách này nhằm mục đích dạy mọi người cách lập trình bằng Python. Nó có sẵn trực tuyến ở dạng PDF và có thể duyệt được. Một lần nữa, bạn có thể sử dụng tài nguyên này để tìm hiểu thêm về một số chủ đề chúng tôi đề cập.

|  |
| --- |
| The [official language reference](https://docs.python.org/3/reference/index.html). This is a technical reference of all Python language components. At first, this resource might be a little too complex, but as you learn how Python works and how it’s built, this can be a useful reference to understand the details of these interactions. |

Các[tài liệu tham khảo ngôn ngữ chính thức](https://docs.python.org/3/reference/index.html). Đây là tài liệu tham khảo kỹ thuật của tất cả các thành phần ngôn ngữ Python. Lúc đầu, tài nguyên này có thể hơi phức tạp nhưng khi bạn tìm hiểu cách Python hoạt động và cách nó được xây dựng, đây có thể là tài liệu tham khảo hữu ích để hiểu chi tiết về các tương tác này.

|  |
| --- |
| **Note:** As you progress this course you may see updated or repeated content due to Google’s regular review of the certificate program to provide the most up-to-date skills and tools in the industry. These changes include the addition of new practice material and supportive content. In addition, Google’s certificate updates sometimes include changes to graded items like quizzes and labs. If you have previously completed a graded item in this course, you *may*  need to take a new assessment to complete this course. We hope the new content will support you on your learning journey! |

**Lưu ý:** Khi tiến hành khóa học này, bạn có thể thấy nội dung được cập nhật hoặc lặp lại do chương trình chứng chỉ được Google thường xuyên xem xét nhằm cung cấp các kỹ năng và công cụ cập nhật nhất trong ngành. Những thay đổi này bao gồm việc bổ sung tài liệu thực hành mới và nội dung hỗ trợ. Ngoài ra, các bản cập nhật chứng chỉ của Google đôi khi bao gồm những thay đổi đối với các mục được xếp loại như câu đố và phòng thí nghiệm. Nếu trước đây bạn đã hoàn thành một mục được chấm điểm trong khóa học này, bạn *có thể*  cần thực hiện bài đánh giá mới để hoàn thành khóa học này. Chúng tôi hy vọng nội dung mới sẽ hỗ trợ bạn trên hành trình học tập của mình!

***1.7. Meet & Greet – Gặp gỡ và chào hỏi***

**2. Introduction to Programming – Giới thiệu về lập trình**

***2.1. The Beginning of Your Programming Journey – Sự khởi đầu của hành trình lập trình của bạn***

|  |
| --- |
| As the Chinese proverb says, a journey of a thousand miles begins with a single step. Today's a big day, you're taking your first step in your journey to learn how to write scripts in Python. It's going to be a little challenging at times, but really it's not that scary. We'll go slow and give you everything you need to fully grasp each concept before we move along. In the next few videos, you'll discover the fundamental concepts of computer programming. You'll learn what a programming language is, what scripting is, what languages are out there other than Python, and how this all relates to IT. We'll also have you coding before you know it with small coding exercises we've cooked up to give you hands-on practice with Python. This will include writing your very first Python script. But always keep in mind, if at any point along the way you feel lost or confused, don't panic. You can watch the videos as many times as you need to let the concept sink in, plus you can ask questions in the discussion forums, which is one of the best ways to find extra information and connect with other learners. When I was asked to participate in this program, it made me think about when I first started to code. If I could give that younger version of myself a piece of advice, this is what I would tell her, it never works the first time. Seriously, as a newbie, I expected it all to work like magic. I thought that following the rules and getting it right the first time would prove my value as a coder, but that's just not true, not even the best of the best. If you expect to write perfect code on the first shot you're going to be disappointed. You hear that younger self? |

Như câu ngạn ngữ Trung Hoa có nói, cuộc hành trình ngàn dặm bắt đầu bằng một bước đi duy nhất.bước chân. Hôm nay là một ngày trọng đại, bạn đang bước bước đầu tiên trong hành trình học tập của mìnhcách viết script bằng Python.Đôi khi sẽ có chút khó khăn nhưng thực sự không phải vậyđáng sợ. Chúng tôi sẽ đi chậm và cung cấp cho bạn mọi thứ bạn cần để nắm bắt đầy đủ từng vấn đềkhái niệm trước khi chúng ta di chuyển.Trong một số video tiếp theo, bạn sẽ khám phá các khái niệm cơ bản về máy tínhlập trình. Bạn sẽ tìm hiểu ngôn ngữ lập trình là gì, kịch bản là gìlà, có những ngôn ngữ nào ngoài Python và tất cả những ngôn ngữ này liên quan đếnNÓ. Chúng tôi cũng sẽ giúp bạn viết mã trước khi bạn biết điều đó bằng các bài tập viết mã nhỏchúng tôi đã chuẩn bị sẵn sàng để giúp bạn thực hành Python.Điều này sẽ bao gồm việc viết tập lệnh Python đầu tiên của bạn.Nhưng hãy luôn ghi nhớ, nếu tại bất kỳ thời điểm nào trên đường đi bạn cảm thấy lạc lõng hoặcbối rối, đừng hoảng sợ.Bạn có thể xem video nhiều lần nếu cần để thấm nhuần khái niệm,Ngoài ra, bạn có thể đặt câu hỏi trong các diễn đàn thảo luận, đây là một trong nhữngcách tốt nhất để tìm thêm thông tin và kết nối với những người học khác.Khi tôi được đề nghị tham gia chương trình này, điều đó khiến tôi nghĩ đến thời điểm tôilần đầu tiên bắt đầu viết mã.Nếu tôi có thể cho phiên bản trẻ hơn của mình một lời khuyên thì đây là điều tôisẽ nói với cô ấy, nó không bao giờ có tác dụng trong lần đầu tiên.Nghiêm túc mà nói, là một người mới, tôi đã mong đợi tất cả sẽ hoạt động như một phép thuật.Tôi nghĩ rằng việc tuân theo các quy tắc và làm đúng ngay lần đầu tiên sẽchứng minh giá trị của tôi với tư cách là một lập trình viên, nhưng điều đó không đúng, thậm chí không phải là điều tốt nhấttốt nhất. Nếu bạn muốn viết mã hoàn hảo ngay lần chụp đầu tiên thì bạn sẽ phảithất vọng. Bạn có nghe thấy bản thân trẻ hơn không?

|  |
| --- |
| Try not to feel overwhelmed by the details. Connecting the dots only comes with experience, so the best way to learn is to just jump in. The truth is everyone learns at their own pace. If you already know some of these concepts, feel free to skip ahead to the parts that interests you the most. If you're starting from scratch, take as long as you need for each concept. The assessments will be right there waiting for you when you're done, and if at any point you start doubting yourself, remember, even the most advanced programmers started thinking, Python? |

Cố gắng đừng cảm thấy choáng ngợp bởi các chi tiết.Kết nối các điểm chỉ đi kèm với kinh nghiệm, vì vậy cách học tốt nhất làđể nhảy vào.Sự thật là mọi người đều học theo tốc độ riêng của mình.Nếu bạn đã biết một số khái niệm này, vui lòng bỏ qua phần tiếp theo.những phần mà bạn quan tâm nhất.Nếu bạn đang bắt đầu từ đầu, hãy dành khoảng thời gian bạn cần cho mỗi khái niệm.Các bài đánh giá sẽ ở ngay đó chờ bạn khi bạn hoàn thành và nếubất cứ lúc nào bạn bắt đầu nghi ngờ chính mình, hãy nhớ rằng, thậm chí nhiều nhấtlập trình viên tiên tiến bắt đầu suy nghĩ, Python?

|  |
| --- |
| What's Python? Well, we're about to learn all about it, so let's dive in. Next up, we'll be doing a rundown of what programming is. |

Python là gì? Chà, chúng ta sắp tìm hiểu tất cả về nó, vì vậy hãy đi sâu vào nhé.Tiếp theo, chúng ta sẽ tìm hiểu tóm tắt về lập trình là gì.

***2.2. What is programming? – Lập trình là gì?***

|  |
| --- |
| At a basic level, a computer program is a recipe of instructions that tells your computer what to do. When you write a program, you create a step by step recipe of what needs to be done to complete a task and when your computer executes the program it reads what you wrote and follows your instructions to the letter. How nice is that? |

Ở mức độ cơ bản, một chương trình máy tính là một công thức hướng dẫn cho bạn biếtmáy tính phải làm gì.Khi bạn viết một chương trình, bạn tạo ra một công thức từng bước về những gì cần phải làm.được thực hiện để hoàn thành một nhiệm vụ và khi máy tính của bạn thực thi chương trình, nó sẽ đọcnhững gì bạn đã viết và làm theo hướng dẫn của bạn trong thư.Điều đó thật tuyệt làm sao?

|  |
| --- |
| The recipe is written in a code called programming language. Programming languages are actually similar to humans spoken languages since they have a syntax and semantics. Now if it's been awhile since your last grammar class, here's a quick refresher on syntax and semantics. In a human language, syntax is the rules for how a sentence is constructed while semantics refers to the actual meaning of the statements. In English, sentences generally have both a subject, that's a person, place, or thing and a predicate, usually a verb and a statement that explains what the subject is doing. Let's take the sentence, Paula loves to program in Python as an example. In this sentence, Paula is the subject and loves to program in Python is the predicate. To form a sentence that others can understand, you need to know both the syntax that constructs the sentence and the semantics that gives it meaning. The same applies to programming languages. In a programming language like Python, the syntax is the rules for how each instruction is written and the semantics is the effects the instructions have. Much like spoken languages, there are lots of programming languages to choose from. Each has its own history, features, and applications but they all share the same fundamental ideas. So once you understand the basic concepts in one programming language, it becomes much easier to learn another. Lastly, computers always do exactly what they're told. So when you write a program, it's important to be super clear about what you want the computer to do. Learning the syntax and semantics of the programming language you choose will allow you to do just that. Make sense? Before we continue, let's spend a moment on terminology. In the next few videos you'll hear the term script being used a bunch. So what's the difference between a script and a program? The line between the two can be a bit blurry. In this course, we'll use the terms interchangeably. |

Công thức được viết bằng mã gọi là ngôn ngữ lập trình.Ngôn ngữ lập trình thực sự giống với ngôn ngữ nói của con người vìhọ có cú pháp và ngữ nghĩa.Bây giờ, nếu đã lâu rồi kể từ lớp học ngữ pháp cuối cùng của bạn, thì đây là phần ôn lại nhanhvề cú pháp và ngữ nghĩa.Trong ngôn ngữ của con người, cú pháp là các quy tắc về cách xây dựng một câutrong khi ngữ nghĩa đề cập đến ý nghĩa thực sự của các câu lệnh.Trong tiếng Anh, các câu thường có cả chủ ngữ, đó là người, địa điểm,hoặc sự vật và một vị ngữ, thường là một động từ và một câu giải thích điều gìchủ thể đang làm. Hãy lấy câu nói, Paula thích lập trìnhPython làm ví dụ.Trong câu này, Paula là chủ đề và yêu thích lập trình Python làThuộc tính. Để tạo thành một câu mà người khác có thể hiểu được, bạn cần phải biếtcả cú pháp xây dựng câu và ngữ nghĩa tạo nên câu đónghĩa. Điều tương tự cũng áp dụng cho các ngôn ngữ lập trình.Trong ngôn ngữ lập trình như Python, cú pháp là các quy tắc về cách mỗihướng dẫn được viết và ngữ nghĩa là những tác độnghướng dẫn có. Giống như ngôn ngữ nói, có rất nhiều chương trìnhngôn ngữ để lựa chọn.Mỗi cái đều có lịch sử, tính năng và ứng dụng riêng nhưng chúng đều có chungnhững ý tưởng cơ bản.Vì vậy, một khi bạn hiểu các khái niệm cơ bản trong một ngôn ngữ lập trình, nó sẽviệc học cái khác trở nên dễ dàng hơn nhiều.Cuối cùng, máy tính luôn thực hiện chính xác những gì chúng được yêu cầu.Vì vậy, khi bạn viết một chương trình, điều quan trọng là phải hiểu rõ những gìbạn muốn máy tính thực hiện.Học cú pháp và ngữ nghĩa của ngôn ngữ lập trình bạn chọn sẽcho phép bạn làm điều đó.Có lý? Trước khi tiếp tục, chúng ta hãy dành một chút thời gian về thuật ngữ.Trong một số video tiếp theo, bạn sẽ nghe thấy rất nhiều thuật ngữ script được sử dụng.Vậy sự khác biệt giữa kịch bản và chương trình là gì?Ranh giới giữa hai điều này có thể hơi mờ nhạt.Trong khóa học này, chúng ta sẽ sử dụng các thuật ngữ thay thế cho nhau.

|  |
| --- |
| In general, you can think of scripts as programs with a short development cycle that can be created and deployed rapidly. |

Nói chung, bạn có thể coi tập lệnh là các chương trình có chu kỳ phát triển ngắn.có thể được tạo ra và triển khai nhanh chóng.

|  |
| --- |
| In other words, a script is a program that is short, simple, and can be written very quickly. In this course we'll focus on a specific scripting language called python which we'll use to learn the basics of programming. We'll learn about the python syntax, the rules of how to write a python program, and the semantics or meaning of the different pieces involved. Before we start learning how to code and having you write your first python script, let's talk more about what automation is and why it's useful. |

Nói cách khác, script là một chương trình ngắn gọn, đơn giản và có thểviết rất nhanh. Trong khóa học này, chúng tôi sẽ tập trung vào một kịch bản cụ thểngôn ngữ được gọi là python mà chúng ta sẽ sử dụng để tìm hiểu những điều cơ bản về lập trình.Chúng ta sẽ tìm hiểu về cú pháp của python, các quy tắc viết một pythonchương trình và ngữ nghĩa hoặc ý nghĩa của các phần khác nhau có liên quan.Trước khi chúng ta bắt đầu học cách viết mã và yêu cầu bạn viết con trăn đầu tiên của mìnhscript, hãy nói thêm về tự động hóa là gì và tại sao nó lại hữu ích.

***2.3. What is automation? – Tự động hóa là gì?***

|  |
| --- |
| Although we might not realize it, we reap the benefits of automation all the time in our daily lives. Do you ever pay your bills with scheduled payments or use a self check out at the grocery store? I always set my coffee machine to start brewing before I've even gotten out of bed. The promise of fresh coffee makes early mornings way easier. Automation is the process of replacing a manual step with one that happens automatically. Take a traffic light for example, which continuously regulates the flow of vehicles at an intersection. |

Mặc dù chúng ta có thể không nhận ra điều đó nhưng chúng ta thu được lợi ích từ việc tự động hóa mọi thứ.thời gian trong cuộc sống hàng ngày của chúng ta.Bạn có bao giờ thanh toán các hóa đơn của mình bằng các khoản thanh toán theo lịch trình hoặc sử dụng phương thức tự kiểm traở ngoài cửa hàng tạp hóa à?Tôi luôn đặt máy pha cà phê của mình bắt đầu pha trước khi tôi ra khỏi phòng.giường. Lời hứa về cà phê mới pha sẽ khiến buổi sáng sớm trở nên dễ dàng hơn.Tự động hóa là quá trình thay thế một bước thủ công bằng một bước tự độngtự động. Lấy đèn giao thông làm ví dụ, nó liên tục điều chỉnhluồng xe cộ qua lại tại một ngã tư.

|  |
| --- |
| A traffic light requires human intervention only when it needs repairs or maintenance. The automatic regulation of traffic means that humans don't have to stand at the intersection manually signaling when cars should stop or go. Instead, people can concentrate on more complex, creative, or difficult tasks like focusing on where you're driving. |

Đèn giao thông chỉ cần sự can thiệp của con người khi cần sửa chữahoặc bảo trì. Việc điều tiết giao thông tự động có nghĩa là con ngườikhông phải đứng ở ngã tư ra hiệu bằng tay khi ô tô dừng lạihoặc đi. Thay vào đó, mọi người có thể tập trung vào những việc phức tạp, sáng tạo hoặc khó khăn hơn.những nhiệm vụ như tập trung vào nơi bạn đang lái xe.

|  |
| --- |
| What's more, traffic lights don't get tired, bored, or accidentally display a green light when they mean red. This highlights another benefit of automation consistency. Let's face it, us humans are flawed and sometimes we make mistakes, a human performing the same tasks hundreds of times will never be as consistent as a machine doing the same thing. But for all of its advantages automation isn't a solution for every situation, some tasks just aren't suited for automation. For example, they may require a degree of creativity or flexibility that automatic systems can't provide or for more complicated or less frequently executed tasks creating the automation may actually be more effort or cost than it's worth. Think about when you get a haircut. What would it take to automate the actions of cutting hair with a machine? The client's height, the shape of their head, their current hair length, and desired hairstyle would all need to be taken into account when designing the automatic system. We need to replicate the creativity and skills of a trained specialist along with extensive testing to ensure the clients safety and quality haircut. And if you've ever had a bad experience at a hair salon, you know quality can be subjective. In this case, the cost and effort of automation just isn't worth the benefits of an automatic haircut would provide which is why we don't have robot hairstylists. Not too complex, right? |

Hơn nữa, đèn giao thông không bị mệt mỏi, nhàm chán hoặc vô tình hiển thịđèn xanh khi chúng có nghĩa là màu đỏ.Điều này nêu bật một lợi ích khác của tính nhất quán tự động hóa.Hãy đối mặt với sự thật, con người chúng ta có nhiều khiếm khuyết và đôi khi chúng ta mắc sai lầm, một con ngườithực hiện cùng một nhiệm vụ hàng trăm lần sẽ không bao giờ nhất quán như mộtmáy làm điều tương tự.Nhưng với tất cả những ưu điểm của nó, tự động hóa không phải là giải pháp cho mọiTrong trường hợp này, một số nhiệm vụ không phù hợp với tự động hóa.Ví dụ, họ có thể yêu cầu mức độ sáng tạo hoặc linh hoạthệ thống tự động không thể cung cấp hoặc phức tạp hơn hoặc ít thường xuyên hơncác nhiệm vụ được thực hiện tạo ra sự tự động hóa thực sự có thể tốn nhiều công sức hoặc chi phí hơnnó có giá trị. Hãy nghĩ về thời điểm bạn đi cắt tóc.Cần những gì để tự động hóa các thao tác cắt tóc bằng máy?Chiều cao của khách hàng, hình dạng đầu, độ dài tóc hiện tại của họ vàkiểu tóc mong muốn sẽ cần phải được tính đến khi thiết kếhệ thống tự động. Chúng ta cần nhân rộng sự sáng tạo và kỹ năng của một người được đào tạochuyên môn cùng với thử nghiệm rộng rãi để đảm bảo an toàn và chất lượng cho khách hàngcắt tóc. Và nếu bạn đã từng có trải nghiệm tồi tệ ở tiệm làm tóc, bạn biết đấychất lượng có thể mang tính chủ quan.Trong trường hợp này, chi phí và nỗ lực tự động hóa không mang lại lợi ích gìviệc cắt tóc tự động sẽ mang lại lý do tại sao chúng ta không có robotNhà tạo mẫu tóc. Không quá phức tạp phải không?

|  |
| --- |
| Automation is a powerful tool when used in the right place at the right moment. It can save time, reduce errors, increase consistency, and provide a way to centralize solutions and mistakes, making them easier to fix. Throughout this course, and in upcoming ones we'll be talking about when it makes sense to apply automation and exactly how you do it. Eventually knowing when and where to use automation will become automatic for you. |

Tự động hóa là một công cụ mạnh mẽ khi được sử dụng đúng nơi, đúng thời điểm.Nó có thể tiết kiệm thời gian, giảm sai sót, tăng tính nhất quán và cung cấp một cáchđể tập trung các giải pháp và sai sót, làm cho chúng dễ dàng sửa chữa hơn.Trong suốt khóa học này và trong những khóa học sắp tới, chúng ta sẽ nói về thời điểmviệc áp dụng tự động hóa và chính xác cách bạn thực hiện điều đó là điều hợp lý.Cuối cùng, việc biết khi nào và ở đâu sử dụng tự động hóa sẽ trở nên tự động đối vớiBạn.

***2.4. Uses for Automation – Sử dụng cho Tự động hóa***

|  |
| --- |
| **Uses for Automation** |

**Sử dụng cho Tự động hóa**

|  |
| --- |
| Scripts can be used for automating specific tasks. Automation is used to replace a repetitive manual step with one that happens automatically. Humans are fallible. They can become tired, make mistakes, fail to follow instructions, be inconsistent in their job performance, and more. In contrast, automated processes complete instructions exactly as coded, in a consistent manner. They can run 24 hours a day, everyday, without tiring. For many tasks that are appropriate for automation, it can be more cost effective to use automation than human labor. |

Tập lệnh có thể được sử dụng để tự động hóa các tác vụ cụ thể. Tự động hóa được sử dụng để thay thế một bước thủ công lặp đi lặp lại bằng một bước diễn ra tự động. Con người có thể mắc sai lầm. Họ có thể trở nên mệt mỏi, mắc lỗi, không làm theo hướng dẫn, không nhất quán trong hiệu suất công việc, v.v. Ngược lại, các quy trình tự động hoàn thành các hướng dẫn chính xác như được mã hóa và theo cách nhất quán. Họ có thể chạy 24 giờ một ngày, hàng ngày mà không mệt mỏi. Đối với nhiều nhiệm vụ phù hợp với tự động hóa, việc sử dụng tự động hóa có thể tiết kiệm chi phí hơn so với sử dụng lao động của con người.

|  |
| --- |
| **Appropriate uses for automation include:**   * The automatic timing and regulation of traffic lights * A repetitive task that is at high risk for human error * Sending commands to a computer * Detecting and removing duplicates of data * Sending automated emails that are personalized by pulling individual names from a database and plugging them into the email * Updating a large number of file permissions * Reporting on system data, like disk or memory usage * Installing software * Generating reports * Deploying a file or a computer program to all computers on a company network * Using a configuration management system to deploy software patches, after a human has *designed* the system * Populating an e-commerce site with products * Setting the home directory and access permissions for users |

**Việc sử dụng thích hợp cho tự động hóa bao gồm:**

* Tự động định thời gian và điều chỉnh đèn giao thông
* Một nhiệm vụ lặp đi lặp lại có nguy cơ cao do lỗi của con người
* Gửi lệnh tới máy tính
* Phát hiện và loại bỏ dữ liệu trùng lặp
* Gửi email tự động được cá nhân hóa bằng cách lấy tên riêng lẻ từ cơ sở dữ liệu và cắm chúng vào email
* Cập nhật một số lượng lớn quyền truy cập tập tin
* Báo cáo về dữ liệu hệ thống, như mức sử dụng ổ đĩa hoặc bộ nhớ
* Cài đặt phần mềm
* Tạo báo cáo
* Triển khai một tập tin hoặc một chương trình máy tính tới tất cả các máy tính trong mạng công ty
* Sử dụng hệ thống quản lý cấu hình để triển khai các bản vá phần mềm, sau khi con người *thiết kế* hệ thống
* Đưa sản phẩm vào một trang web thương mại điện tử
* Thiết lập thư mục chính và quyền truy cập cho người dùng

|  |
| --- |
| **Automation is not always an appropriate or complete solution** |

**Tự động hóa không phải lúc nào cũng là giải pháp phù hợp và hoàn chỉnh**

|  |
| --- |
| Automation cannot perform all human work. Tasks that call for human creativity, social connection, psychology, flexibility, ingenuity, evaluation, and/or complex analytic work are not good candidates for full automation. Sometimes automation can be used to perform one or more subtasks of a larger set of tasks – but – human intervention is required to complete the tasks. The following are some examples of tasks that cannot or should not be **fully** automated:   * Items that require human evaluation and analytic skills:   + *Designing* a configuration management system   + Investigating and troubleshooting all end user problems   + Writing a computer program   + Building a new startup business * Items that require human creativity and/or an eye for aesthetic qualities*:*   + Designing an attractive webpage *(AI can do this, but simple automation cannot)*   + Wedding photography   + Haircuts and styling * Items that cannot be automated due to basic physics:   + Troubleshooting or repairing machines that cannot power on or boot up * Items that need human interaction, psychology, and/or evaluation skills:   + Interviewing and hiring new employees   + Customer service *(chat bots cannot address every customer service need)* * Items that should not be fully automated due to costs and safety:   + Grocery store checkout process, including bagging groceries   + Tasks that are less expensive to perform manually |

Tự động hóa không thể thực hiện tất cả công việc của con người. Các nhiệm vụ đòi hỏi sự sáng tạo của con người, kết nối xã hội, tâm lý, tính linh hoạt, sự khéo léo, đánh giá và/hoặc công việc phân tích phức tạp không phải là ứng cử viên sáng giá cho tự động hóa hoàn toàn. Đôi khi tự động hóa có thể được sử dụng để thực hiện một hoặc nhiều nhiệm vụ phụ của một nhóm nhiệm vụ lớn hơn – nhưng – cần có sự can thiệp của con người để hoàn thành các nhiệm vụ. Sau đây là một số ví dụ về các nhiệm vụ không thể hoặc không nên tự động hóa **hoàn toàn** :

* Các hạng mục yêu cầu kỹ năng đánh giá và phân tích của con người:
  + *Thiết kế* hệ thống quản lý cấu hình
  + Điều tra và khắc phục sự cố tất cả các vấn đề của người dùng cuối
  + Viết chương trình máy tính
  + Xây dựng một doanh nghiệp khởi nghiệp mới
* Những đồ vật đòi hỏi sự sáng tạo của con người và/hoặc con mắt thẩm mỹ *:*
  + Thiết kế một trang web hấp dẫn *(AI có thể làm được điều này, nhưng tự động hóa đơn giản thì không)*
  + Chụp ảnh cưới
  + Cắt tóc và tạo kiểu
* Các mục không thể tự động hóa do vật lý cơ bản:
  + Khắc phục sự cố hoặc sửa chữa máy không bật nguồn hoặc không khởi động được
* Các hạng mục cần có kỹ năng tương tác, tâm lý và/hoặc đánh giá của con người:
  + Phỏng vấn và tuyển dụng nhân viên mới
  + Dịch vụ khách hàng *(bot trò chuyện không thể đáp ứng mọi nhu cầu về dịch vụ khách hàng)*
* Các hạng mục không nên tự động hóa hoàn toàn do chi phí và sự an toàn:
  + Quy trình thanh toán tại cửa hàng tạp hóa, bao gồm cả việc đóng gói hàng tạp hóa
  + Các tác vụ ít tốn kém hơn khi thực hiện thủ công

|  |
| --- |
| **Artificial Intelligence** |

**Trí tuệ nhân tạo**

|  |
| --- |
| It is important to understand that basic automation is not the same as artificial intelligence. Automation is used to explicitly instruct a machine on how to perform a task. Artificial intelligence (AI) involves training a computing machine to perform more complex tasks through a process called machine learning. This process prepares the AI software to perform new tasks without a human needing to program explicit instructions for each task. Although AI is often used for automating human tasks, AI automation is much more complex than basic automation. |

Điều quan trọng là phải hiểu rằng tự động hóa cơ bản không giống như trí tuệ nhân tạo. Tự động hóa được sử dụng để hướng dẫn rõ ràng cho máy cách thực hiện một nhiệm vụ. Trí tuệ nhân tạo (AI) liên quan đến việc đào tạo một máy tính để thực hiện các nhiệm vụ phức tạp hơn thông qua một quá trình gọi là học máy. Quá trình này chuẩn bị cho phần mềm AI thực hiện các nhiệm vụ mới mà không cần con người lập trình hướng dẫn rõ ràng cho từng nhiệm vụ. Mặc dù AI thường được sử dụng để tự động hóa các nhiệm vụ của con người nhưng tự động hóa AI phức tạp hơn nhiều so với tự động hóa cơ bản.

***2.5. Getting Computers to Work for You – Để máy tính làm việc cho bạn***

|  |
| --- |
| Working in IT, a lot of what we do boils down to using a computer to perform a certain task. In your job you might create user accounts, configure the network, install software, backup existing data, or execute a whole range of other computer based tasks from day to day. Back in my first IT job, I realized that every day I came into work I typed the same three commands to authenticate into systems. Those credentials timed out everyday by design, for security reasons, so I created a script that would automatically run these commands for me every morning to avoid having to type them myself. Funny enough, the team that monitors anomalous activity discovered my little invention and contacted me to remove it, oops. |

Làm việc trong lĩnh vực CNTT, phần lớn công việc chúng ta làm đều xoay quanh việc sử dụng máy tính để thực hiệnnhiệm vụ nhất định. Trong công việc của mình, bạn có thể tạo tài khoản người dùng, định cấu hìnhmạng, cài đặt phần mềm, sao lưu dữ liệu hiện có hoặc thực hiện toàn bộ phạm vicác công việc khác dựa trên máy tính hàng ngày.Trở lại với công việc IT đầu tiên của mình, tôi nhận ra rằng mỗi ngày khi đến chỗ làm tôi đều gõba lệnh tương tự để xác thực vào hệ thống.Những thông tin đăng nhập đó bị hết thời gian hàng ngày theo thiết kế, vì lý do bảo mật, vì vậy tôiđã tạo một tập lệnh tự động chạy các lệnh này cho tôimỗi sáng để tránh phải tự mình gõ chúng.Buồn cười thay, nhóm theo dõi hoạt động bất thường đã phát hiện ra điều nhỏ bé của tôi.phát minh và liên hệ với tôi để loại bỏ nó, rất tiếc.

|  |
| --- |
| Tasks performed by a computer that need to be done multiple times with little variation are really well suited for automation, because when you automate a task you avoid the possibility of human errors, and reduce the time it takes to do it. Imagine this scenario: your company had a booth at a recent conference and has gathered a huge list of emails from people interested in learning more about your products. You want to send these people your monthly email newsletter, but some of the people on the list are already subscribed to receive it. So how do you make sure everyone receives your newsletter, without accidentally sending it to the same person twice? Well, you could manually check each email address one by one to make sure you only add new ones to the list, sounds boring and inefficient, right? It could be, and it's also more error prone, you might accidentally miss new emails, or add emails that were already there, or it might get so boring you fall asleep at your desk. Even your automated coffee machine won't help you out there. |

Các tác vụ được thực hiện bởi máy tính cần được thực hiện nhiều lần với ítbiến thể thực sự rất phù hợp cho tự động hóa, bởi vì khi bạn tự động hóa mộtnhiệm vụ của bạn, bạn tránh được khả năng xảy ra lỗi của con người và giảm thời gian cần thiết để thực hiệnlàm đi. Hãy tưởng tượng kịch bản này: công ty của bạn có một gian hàng tại một sự kiện gần đây.hội nghị và đã thu thập được một danh sách lớn các email từ những người quan tâm đếntìm hiểu thêm về sản phẩm của bạn.Bạn muốn gửi cho những người này bản tin email hàng tháng của bạn, nhưng một sốnhững người trong danh sách đã đăng ký để nhận nó.Vậy làm cách nào để đảm bảo mọi người đều nhận được bản tin của bạn mà không cầnvô tình gửi nó cho cùng một người hai lần?Chà, bạn có thể kiểm tra thủ công từng địa chỉ email một để đảm bảobạn chỉ thêm những cái mới vào danh sách, nghe có vẻ nhàm chán và kém hiệu quả phải không?Điều đó có thể xảy ra và cũng dễ xảy ra lỗi hơn, bạn có thể vô tình bỏ lỡ những thông tin mớiemail hoặc thêm các email đã có sẵn ở đó, nếu không nó có thể khiến bạn nhàm chánngủ quên tại bàn làm việc của bạn.Ngay cả máy pha cà phê tự động của bạn cũng không giúp được gì cho bạn.

|  |
| --- |
| So what could you do instead? You could get the computer to do the work for you. You could write a program that checks for duplicates, and then adds each new email to the list. Your computer will do exactly as it's told no matter how many emails there are in the list, so it won't get tired or make any mistakes. Even better, once you've written the program you can use the same code in future situations, saving you even more time, pretty cool, right? It gets better, think about when you're going to send these emails out, if you send them out manually you'll have to send the same email to everybody, personalizing the emails would be way too much manual work. If instead you use automation to send them, you could have the name and company of each person added to the email automatically. The result? More effective emails, without you spending hours inserting names into the text. Automating tasks allows you to focus on projects that are a better use of your time, letting computers do the boring stuff for you. Learning how to program is the first step to being able to do this. If you want to get computers to do the work for you, you're in the right place. Earlier in this video I told you about the first task I ever automated, now I want to tell you about the coolest thing I ever automated. It was a script that changed a bunch of access permissions for a whole lot of Google Internal Services. The script traversed a large directory tree with tons of different files, checked the file contents, and then updated the permissions to the services based on the conditions that I laid out in the script. Okay, I admit I'm a total nerd, but I still think it's really cool. Next up, it's time to share your ideas. What things would you like to automate using programming? While these discussion prompts are optional, they're really fun. Seriously, they let you get to know your fellow learners a bit, and collaborate on ideas and insights. Make sure you read what others are saying, they may give you ideas that you haven't even thought of. After that, you're ready to take your very first quiz of the course. Don't worry, it's just for practice. |

Vậy bạn có thể làm gì thay thế?Bạn có thể nhờ máy tính làm việc đó cho bạn.Bạn có thể viết một chương trình kiểm tra các bản sao và sau đó thêm từng cái mớiemail vào danh sách. Máy tính của bạn sẽ hoạt động chính xác như được yêu cầu cho dù thế nào đi chăng nữatrong danh sách có rất nhiều email nên sẽ không bị mệt hay mắc lỗi gì.Thậm chí tốt hơn, khi bạn đã viết chương trình, bạn có thể sử dụng cùng một đoạn mã trongcác tình huống trong tương lai, giúp bạn tiết kiệm nhiều thời gian hơn, khá tuyệt phải không?Tốt hơn hết, hãy nghĩ xem khi nào bạn sẽ gửi những email này đi, nếu bạngửi chúng theo cách thủ công, bạn sẽ phải gửi cùng một email cho mọi người,việc cá nhân hóa các email sẽ tốn quá nhiều công sức thủ công.Thay vào đó, nếu bạn sử dụng tính năng tự động hóa để gửi chúng, bạn có thể có tên vàcông ty của mỗi người được thêm vào email tự động.Kết quả? Email hiệu quả hơn mà bạn không phải mất hàng giờ để chèntên vào văn bản. Tự động hóa các tác vụ cho phép bạn tập trung vào các dự ánsử dụng thời gian của bạn tốt hơn, để máy tính làm những công việc nhàm chán cho bạn.Học cách lập trình là bước đầu tiên để có thể làm được điều này.Nếu bạn muốn máy tính làm việc cho mình thì bạn đã đến đúng nơi.Trước đó trong video này, tôi đã kể cho bạn nghe về nhiệm vụ đầu tiên tôi tự động hóa, bây giờ tôimuốn kể cho bạn nghe về điều thú vị nhất mà tôi từng tự động hóa.Đó là một tập lệnh đã thay đổi một loạt quyền truy cập cho rất nhiềuDịch vụ nội bộ của Google.Tập lệnh duyệt qua một cây thư mục lớn với rất nhiều tệp khác nhau,đã kiểm tra nội dung tệp và sau đó cập nhật quyền đối với dịch vụdựa trên các điều kiện mà tôi đặt ra trong kịch bản.Được rồi, tôi thừa nhận mình là một kẻ mọt sách, nhưng tôi vẫn nghĩ nó thực sự rất tuyệt.Tiếp theo, đã đến lúc chia sẻ ý tưởng của bạn.Bạn muốn tự động hóa những việc gì bằng lập trình?Mặc dù những lời nhắc thảo luận này là tùy chọn nhưng chúng thực sự thú vị.Nghiêm túc mà nói, họ cho phép bạn làm quen một chút với những người bạn học của mình vàhợp tác về ý tưởng và hiểu biết sâu sắc.Hãy chắc chắn rằng bạn đọc những gì người khác đang nói, họ có thể cho bạn những ý tưởng mà bạnthậm chí còn chưa nghĩ tới. Sau đó, bạn đã sẵn sàng thực hiện lần đầu tiênbài kiểm tra của khóa học.Đừng lo lắng, nó chỉ để luyện tập thôi.

***2.6. Your Hopes for Automation – Hy vọng của bạn về Tự động hóa***

***2.7. Study Guide: Introduction to Programming – Hướng dẫn học: Nhập môn lập trình***

|  |
| --- |
| **Study Guide: Introduction to Programming** |

**Hướng dẫn học: Nhập môn lập trình**

|  |
| --- |
| Your first practice quiz is coming up soon. This handy study guide should help you prepare for that quiz. The practice quizzes do not count towards your grade in this course. Practice quizzes are opportunities for you to check your understanding of the materials before you take the graded assessments at the end of each module. |

Bài kiểm tra thực hành đầu tiên của bạn sắp ra mắt. Hướng dẫn học tập hữu ích này sẽ giúp bạn chuẩn bị cho bài kiểm tra đó. Các câu hỏi thực hành không được tính vào điểm của bạn trong khóa học này. Các câu hỏi thực hành là cơ hội để bạn kiểm tra mức độ hiểu biết của mình về tài liệu trước khi làm bài đánh giá chấm điểm ở cuối mỗi học phần.

|  |
| --- |
| **Key Terms** |

**Điều khoản quan trọng**

|  |
| --- |
| * **Programming code** - Programming code is a set of written computer instructions, guided by rules, using a computer programming language. It might help to think of the computer instructions as a detailed, step-by-step recipe for performing tasks. The instructions tell computers and machines how to perform an action. Programming code may also be referred to as source code or scripts. |

* **Mã lập trình** - Mã lập trình là tập hợp các hướng dẫn bằng văn bản của máy tính, được hướng dẫn bởi các quy tắc, sử dụng ngôn ngữ lập trình máy tính. Có thể hữu ích nếu coi các hướng dẫn trên máy tính như một công thức chi tiết, từng bước để thực hiện các nhiệm vụ. Các hướng dẫn cho máy tính và máy móc biết cách thực hiện một hành động. Mã lập trình cũng có thể được gọi là mã nguồn hoặc tập lệnh.

|  |
| --- |
| * **Programming languages** - Programming languages are similar to human spoken languages in that they both use syntax and semantics. Programming languages are used to write computer programs.  Some common programming languages include Python, Java, C, C++, C#, and R. |

* **Ngôn ngữ lập trình** - Ngôn ngữ lập trình tương tự như ngôn ngữ nói của con người ở chỗ chúng đều sử dụng cú pháp và ngữ nghĩa. Ngôn ngữ lập trình được sử dụng để viết các chương trình máy tính. Một số ngôn ngữ lập trình phổ biến bao gồm Python, Java, C, C++, C# và R.

|  |
| --- |
| * **Syntax** - Syntax is a set of rules for how statements are constructed in both human and computer languages. Programming syntax includes rules for the order of elements in programming instructions, as well as the use of special characters and their placements in statements. This concept is similar to the syntax rules for grammar and punctuation in human language. |

* **Cú pháp** - Cú pháp là một tập hợp các quy tắc về cách xây dựng các câu lệnh trong cả ngôn ngữ của con người và máy tính. Cú pháp lập trình bao gồm các quy tắc về thứ tự các phần tử trong hướng dẫn lập trình, cũng như việc sử dụng các ký tự đặc biệt và vị trí của chúng trong câu lệnh. Khái niệm này tương tự như các quy tắc cú pháp về ngữ pháp và dấu câu trong ngôn ngữ của con người.

|  |
| --- |
| * **Semantics** - Semantics refers to the intended meaning or effect of statements, or collections of words, in both human and computer languages. Semantic errors are also referred to as logical errors. |

* **Ngữ nghĩa** - Ngữ nghĩa đề cập đến ý nghĩa hoặc tác dụng dự kiến ​​của các câu lệnh hoặc tập hợp các từ, bằng cả ngôn ngữ của con người và máy tính. Lỗi ngữ nghĩa còn được gọi là lỗi logic.

|  |
| --- |
| * **Computer program** - A computer program is a step-by-step list of instructions that a computer follows to reach an intended goal. It is important to be clear and precise about the actions a computer program is supposed to perform because computers will do exactly what they are instructed to do. Computer programs can be long, complex, and accomplish a variety of tasks. They are often developed by computer programmers and software engineers, but anyone can learn to create them. Computer programs may involve a structured development cycle. They can be written in a wide variety of programming languages, such as Python, Java, C++,  R, and more. The completed format of a program is often a single executable file. |

* **Chương trình máy tính** - Chương trình máy tính là danh sách từng bước hướng dẫn mà máy tính tuân theo để đạt được mục tiêu đã định. Điều quan trọng là phải rõ ràng và chính xác về các hành động mà một chương trình máy tính phải thực hiện vì máy tính sẽ thực hiện chính xác những gì chúng được hướng dẫn làm. Các chương trình máy tính có thể dài, phức tạp và thực hiện nhiều nhiệm vụ khác nhau. Chúng thường được phát triển bởi các lập trình viên máy tính và kỹ sư phần mềm, nhưng bất kỳ ai cũng có thể học cách tạo ra chúng. Các chương trình máy tính có thể bao gồm một chu trình phát triển có cấu trúc. Chúng có thể được viết bằng nhiều ngôn ngữ lập trình, chẳng hạn như Python, Java, C++, R, v.v. Định dạng hoàn chỉnh của một chương trình thường là một tệp thực thi duy nhất.

|  |
| --- |
| * **Script** - Scripts are usually shorter and less complex than computer programs. Scripts are often used to automate specific tasks. However, they can be used for complex tasks if needed. Scripts are often written by IT professionals, but anyone can learn to write scripts. Scripts have a shorter, less structured development cycle as compared to the development of complex computer programs and software. Scripts can be written in a variety of programming languages, like Python, Javascript, Ruby, Bash, and more. Some scripting languages are interpreted languages and are only compatible with certain platforms. |

* **Tập lệnh** - Tập lệnh thường ngắn hơn và ít phức tạp hơn chương trình máy tính. Các tập lệnh thường được sử dụng để tự động hóa các tác vụ cụ thể. Tuy nhiên, chúng có thể được sử dụng cho các nhiệm vụ phức tạp nếu cần. Các tập lệnh thường được viết bởi các chuyên gia CNTT, nhưng bất kỳ ai cũng có thể học cách viết tập lệnh. Các tập lệnh có chu kỳ phát triển ngắn hơn, ít cấu trúc hơn so với việc phát triển các chương trình và phần mềm máy tính phức tạp. Tập lệnh có thể được viết bằng nhiều ngôn ngữ lập trình, như Python, Javascript, Ruby, Bash, v.v. Một số ngôn ngữ kịch bản là ngôn ngữ được thông dịch và chỉ tương thích với một số nền tảng nhất định.

|  |
| --- |
| * **Automation** - Automation is used to replace a repetitive manual step with one that happens automatically. |

* **Tự động hóa** - Tự động hóa được sử dụng để thay thế một bước thủ công lặp đi lặp lại bằng một bước diễn ra tự động.

|  |
| --- |
| * **Output** - Output is the end result of a task performed by a function or computer program. Output can include a single value, a report, entries into a database, and more. |

* **Đầu ra** - Đầu ra là kết quả cuối cùng của một tác vụ được thực hiện bởi một chức năng hoặc chương trình máy tính. Đầu ra có thể bao gồm một giá trị, một báo cáo, các mục nhập vào cơ sở dữ liệu, v.v.

|  |
| --- |
| * **Input** - Input is information that is provided to a program by the end user. Input can be text, voice, images, biometrics, and more. |

* **Đầu vào** - Đầu vào là thông tin được người dùng cuối cung cấp cho chương trình. Đầu vào có thể là văn bản, giọng nói, hình ảnh, sinh trắc học, v.v.

|  |
| --- |
| * **Functions** - A function is a reusable block of code that performs a specific task. |

* **Hàm** - Hàm là một khối mã có thể tái sử dụng để thực hiện một tác vụ cụ thể.

|  |
| --- |
| * **Variables** - Variables are used to temporarily store changeable values in programming code. |

* **Biến** - Biến được sử dụng để lưu trữ tạm thời các giá trị có thể thay đổi trong mã lập trình.

***2.8. Practice Quiz: Introduction to Programming – Bài kiểm tra thực hành: Giới thiệu về lập trình***

**3. Introduction to Python – Giới thiệu về Python**

***3.1. What is Python? – Python là gì?***

|  |
| --- |
| Welcome back. How did you do on your first quiz? If you got most of the questions right, great job. If not, no worries it's all part of learning. We'll be here to help you check that you've really got your head around these concepts with regular quizzes like this. If you ever find a question tricky, go back and review the videos and then try the quiz again. You want to feel super comfortable with what you've learned before jumping into the next lesson. Remember, take your time. I will be here whenever you're ready to move on. Okay. Feeling good? Great. Let us dive in. In this course, we will use the Python programming language to demonstrate basic programming concepts and how to apply them to writing scripts. We have mentioned that there are a bunch of programming languages out there. So why pick Python? Well, we chose Python for a few reasons. First off, programming in Python usually feels similar to using a human language. This is because Python makes it easy to express what we want to do with syntax that's easy to read and write. Check out this example. There is a lot to unpack here so don't worry if you don't understand it right away, we'll get into the nitty-gritty details later in the course. But even if you've never seen a line of code before, you might be able to guess what this code does. It defines a list with names of friends and then creates a greeting for each name in the list. Now it is your turn to make friends with Python. Try it out and see what happens. |

Chào mừng trở lại. Bạn đã làm bài kiểm tra đầu tiên như thế nào?Nếu bạn trả lời đúng hầu hết các câu hỏi thì tuyệt vời.Nếu không, đừng lo lắng, đó là một phần của việc học.Chúng tôi sẽ ở đây để giúp bạn kiểm tra xem bạn có thực sự quan tâm đến những vấn đề này khôngkhái niệm với các câu đố thường xuyên như thế này.Nếu bạn thấy câu hỏi nào khó, hãy quay lại xem lại video rồi thửcâu đố nữa. Bạn muốn cảm thấy cực kỳ thoải mái với những gì bạn đã học đượctrước khi chuyển sang bài học tiếp theo.Hãy nhớ, hãy dành thời gian của bạn.Tôi sẽ ở đây bất cứ khi nào bạn sẵn sàng bước tiếp.Được rồi. Cảm thấy tốt? Tuyệt vời.Hãy cùng tìm hiểu sâu hơn. Trong khóa học này, chúng ta sẽ sử dụng ngôn ngữ lập trình Python đểthể hiện các khái niệm lập trình cơ bản và cách áp dụng chúng vào việc viếtkịch bản. Chúng tôi đã đề cập rằng có rất nhiều ngôn ngữ lập trìnhở đó. Vậy tại sao lại chọn Python?Chà, chúng tôi chọn Python vì một vài lý do.Trước hết, lập trình bằng Python thường có cảm giác tương tự như sử dụng con người.ngôn ngữ. Điều này là do Python giúp chúng ta dễ dàng diễn đạt những gì chúng ta muốn làmvới cú pháp dễ đọc và dễ viết.Hãy xem ví dụ này. Có rất nhiều thứ để giải nén ở đây nên đừng lo lắng nếu bạnkhông hiểu ngay, chúng ta sẽ đi vào chi tiết quan trọng sautrong khóa học. Nhưng ngay cả khi bạn chưa bao giờ nhìn thấy một dòng mã nào trước đó, bạn vẫn có thểcó thể đoán được mã này làm gì.Nó xác định một danh sách có tên bạn bè và sau đó tạo lời chào cho mỗi ngườitên trong danh sách. Bây giờ đến lượt bạn kết bạn với Python.Hãy thử nó và xem những gì sẽ xảy ra.

|  |
| --- |
| Throughout this course, you will execute Python code using your web browser. We'll start with some small coding exercises using code blocks just like the one you experimented with. Later on as you develop your skills, you'll work on larger more complex coding exercises using other tools. Getting good at something takes a whole lot of practice and programming and Python is no different. We recommend that you practice every example we share in this course on your own. If you do not have Python installed on your machine, no worries, you can still practice using an online Python interpreter. Check out the next reading for links to the most popular Python interpreters available online. Now I am sure you are wondering what the heck is a Python interpreter. In programming, an interpreter is the program that reads and executes code. Remember how we said a computer program is like a recipe with step-by-step instructions? Well, if your recipe is written in Python, the Python interpreter is the program that reads what is in the recipe and translates it into instructions for your computer to follow. |

Trong suốt khóa học này, bạn sẽ thực thi mã Python bằng cách sử dụng trang web của mìnhbrowser. Chúng ta sẽ bắt đầu với một số bài tập viết mã nhỏ sử dụng các khối mãgiống như cái bạn đã thử nghiệm.Sau này khi bạn phát triển các kỹ năng của mình, bạn sẽ làm việc trên những vấn đề lớn hơn, phức tạp hơn.bài tập mã hóa bằng các công cụ khác.Muốn giỏi một việc gì đó cần phải thực hành và lập trình rất nhiều vàPython cũng không khác.Chúng tôi khuyên bạn nên thực hành mọi ví dụ mà chúng tôi chia sẻ trong khóa học này trênsở hữu. Nếu bạn chưa cài đặt Python trên máy của mình, đừng lo lắng,bạn vẫn có thể thực hành bằng trình thông dịch Python trực tuyến.Hãy xem bài đọc tiếp theo để biết các liên kết đến các trình thông dịch Python phổ biến nhấtcó sẵn trên mạng. Bây giờ tôi chắc chắn bạn đang tự hỏi Python là cái quái gìthông dịch viên. Trong lập trình, trình thông dịch là chương trình đọcvà thực thi mã.Hãy nhớ chúng ta đã nói chương trình máy tính giống như một công thức nấu ăn được thực hiện theo từng bước.hướng dẫn? Chà, nếu công thức của bạn được viết bằng Python, Pythontrình thông dịch là chương trình đọc nội dung trong công thức và dịch nóvào hướng dẫn để máy tính của bạn làm theo.

|  |
| --- |
| Eventually, you'll want to install Python on your computer so you can run it locally and experiment with it as much as you like. We'll guide you through how to install Python in the upcoming course but you don't have to have it installed to get your first taste of Python. You can practice with the quizzes we provide and with the online interpreters and code pads that we'll give you links to in the next reading. We'll provide a whole bunch of exercises for you but feel free to come up with your own and share them in the discussion forums. Feel free to get creative. This is your chance to show off your new skills. |

Cuối cùng, bạn sẽ muốn cài đặt Python trên máy tính của mình để có thể chạynó cục bộ và thử nghiệm nó bao nhiêu tùy thích.Chúng tôi sẽ hướng dẫn bạn cách cài đặt Python trong khóa học sắp tới nhưng bạnkhông cần phải cài đặt nó để có được trải nghiệm Python đầu tiên.Bạn có thể thực hành với các câu đố chúng tôi cung cấp và với các phiên dịch viên trực tuyếnvà các bảng mã mà chúng tôi sẽ cung cấp cho bạn các liên kết trong bài đọc tiếp theo.Chúng tôi sẽ cung cấp rất nhiều bài tập cho bạn nhưng bạn cứ thoải mái đến nhélên ý tưởng của riêng bạn và chia sẻ chúng trong các diễn đàn thảo luận.Hãy thoải mái sáng tạo.Đây là cơ hội để bạn thể hiện những kỹ năng mới của mình.

***3.2. Python Resources – Tài nguyên Python***

|  |
| --- |
| **Python Resources** |

**Tài nguyên Python**

|  |
| --- |
| **More About Python** |

**Thông tin thêm về Python**

|  |
| --- |
| **Using Python on your own** |

**Sử dụng Python một mình**

|  |
| --- |
| The best way to learn any programming language is to practice it on your own as much as you can. If you have Python installed on your computer, you can execute the interpreter by running the python3 command (or just python on Windows), and you can close it by typing exit() or Ctrl-D. |

Cách tốt nhất để học bất kỳ ngôn ngữ lập trình nào là tự mình thực hành nó nhiều nhất có thể. Nếu bạn đã cài đặt Python trên máy tính, bạn có thể thực thi trình thông dịch bằng cách chạy lệnh python3 (hoặc chỉ python trên Windows) và bạn có thể đóng nó bằng cách nhập exit() hoặc Ctrl-D.

|  |
| --- |
| If you don’t already have Python installed on your machine, that’s alright. We’ll explain how to install it in an upcoming course. |

Nếu bạn chưa cài đặt Python trên máy của mình thì không sao. Chúng tôi sẽ giải thích cách cài đặt nó trong khóa học sắp tới.

|  |
| --- |
| **Python practice resources** |

**Tài nguyên thực hành Python**

|  |
| --- |
| In the meantime, you can still practice by using one of the many online Python interpreters or codepads available online. There’s not much difference between an interpreter and a codepad. An interpreter is more interactive than a codepad, but they both let you execute code and see the results. |

Trong thời gian chờ đợi, bạn vẫn có thể thực hành bằng cách sử dụng một trong nhiều trình thông dịch Python hoặc bảng mã trực tuyến có sẵn trên mạng. Không có nhiều khác biệt giữa trình thông dịch và bảng mã. Trình thông dịch có tính tương tác cao hơn bảng mã nhưng cả hai đều cho phép bạn thực thi mã và xem kết quả.

|  |
| --- |
| Below, you’ll find links to some of the most popular online interpreters and codepads. Give them a go to find your favorite. |

Dưới đây, bạn sẽ tìm thấy các liên kết đến một số trình thông dịch và bảng mã trực tuyến phổ biến nhất. Cung cấp cho họ một đi để tìm yêu thích của bạn.

|  |
| --- |
| * <https://www.python.org/shell/> * <https://www.onlinegdb.com/online_python_interpreter> * <https://repl.it/languages/python3> * <https://www.tutorialspoint.com/execute_python3_online.php> * <https://rextester.com/l/python3_online_compiler> * <https://trinket.io/python3> |

* <https://www.python.org/shell/>
* <https://www.onlinegdb.com/online_python_interpreter>
* [https://repl.it/lacular/python3](https://repl.it/languages/python3)
* <https://www.tutorialspoint.com/execute_python3_online.php>
* <https://rextester.com/l/python3_online_compiler>
* <https://trinket.io/python3>

|  |
| --- |
| **Additional Python resources** |

**Tài nguyên Python bổ sung**

|  |
| --- |
| While this course will give you information about how Python works and how to write scripts in Python, you’ll likely want to find out more about specific parts of the language. Here are some great ways to help you find additional info:   * Read the [official Python documentation](https://docs.python.org/3/). * Search for answers or ask a question on [Stack Overflow](https://stackoverflow.com/). * Subscribe to the Python [tutor](https://mail.python.org/mailman/listinfo/tutor) mailing list, where you can ask questions and collaborate with other Python learners. * Subscribe to the [Python-announce](https://mail.python.org/mailman/listinfo/python-announce-list) mailing list to read about the latest updates in the language. |

Mặc dù khóa học này sẽ cung cấp cho bạn thông tin về cách Python hoạt động và cách viết tập lệnh bằng Python, nhưng bạn có thể muốn tìm hiểu thêm về các phần cụ thể của ngôn ngữ. Dưới đây là một số cách tuyệt vời để giúp bạn tìm thêm thông tin:

* Đọc[tài liệu Python chính thức](https://docs.python.org/3/).
* Tìm kiếm câu trả lời hoặc đặt câu hỏi trên[tràn ngăn xếp](https://stackoverflow.com/).
* Đăng ký Python[gia sư](https://mail.python.org/mailman/listinfo/tutor)danh sách gửi thư, nơi bạn có thể đặt câu hỏi và cộng tác với những người học Python khác.
* Đăng ký[Python-thông báo](https://mail.python.org/mailman/listinfo/python-announce-list)danh sách gửi thư để đọc về những cập nhật mới nhất trong ngôn ngữ.

|  |
| --- |
| **Python history and current status** |

**Lịch sử và trạng thái hiện tại của Python**

|  |
| --- |
| Python was released almost 30 years ago and has a rich history. You can read more about it on the [History of Python](https://en.wikipedia.org/wiki/History_of_Python) Wikipedia page or in the section on the [history of the software](https://docs.python.org/3.0/license.html) from the official Python documentation. |

Python đã được phát hành gần 30 năm trước và có một lịch sử phong phú. Bạn có thể đọc thêm về nó trên[Lịch sử của Python](https://en.wikipedia.org/wiki/History_of_Python)trang Wikipedia hoặc trong phần trên[lịch sử của phần mềm](https://docs.python.org/3.0/license.html)từ tài liệu Python chính thức.

|  |
| --- |
| Python has recently been called the fastest growing programming language. If you're interested in why this is and how it’s measured, you can find out more in these articles:   * [The Incredible Growth of Python](https://stackoverflow.blog/2017/09/06/incredible-growth-python/) (Stack Overflow) * [Why is Python Growing So Quickly - Future Trends](https://www.netguru.com/blog/why-python-is-growing-so-quickly-future-trends) (Netguru) * [Python Trends](https://technobrains.io/watch-out-for-python-trends-2023/) (TechnoBrains) * [Developer Survey Results 2018](https://insights.stackoverflow.com/survey/2018#technology) (Stack Overflow) |

Python gần đây được gọi là ngôn ngữ lập trình phát triển nhanh nhất. Nếu bạn quan tâm đến lý do tại sao điều này xảy ra và cách nó được đo lường, bạn có thể tìm hiểu thêm trong các bài viết sau:

* [Sự tăng trưởng đáng kinh ngạc của Python](https://stackoverflow.blog/2017/09/06/incredible-growth-python/)(Chồng tràn)
* [Tại sao Python phát triển nhanh như vậy - Xu hướng tương lai](https://www.netguru.com/blog/why-python-is-growing-so-quickly-future-trends)(Netguru)
* [Xu hướng Python](https://technobrains.io/watch-out-for-python-trends-2023/)(TechnoBrain)
* [Kết quả khảo sát nhà phát triển 2018](https://insights.stackoverflow.com/survey/2018#technology)(Chồng tràn)

***3.3. A Note on Syntax and Code Blocks – Lưu ý về cú pháp và khối mã***

|  |
| --- |
| **A Note on Syntax and Code Blocks** |

**Lưu ý về cú pháp và khối mã**

|  |
| --- |
| When writing code, using correct syntax is critical. Even a small typo, like a missing parenthesis bracket or an extra comma, can cause a syntax error and the code won't execute at all. If your code results in an error or an exception, pay close attention to syntax and watch out for minor mistakes. A single wrong character could take hours to identify in long code so it is important to be mindful of syntax when writing code. |

Khi viết mã, sử dụng cú pháp đúng là rất quan trọng. Ngay cả một lỗi đánh máy nhỏ, như thiếu dấu ngoặc đơn hoặc thừa dấu phẩy, cũng có thể gây ra lỗi cú pháp và mã sẽ không thực thi được. Nếu mã của bạn gây ra lỗi hoặc ngoại lệ, hãy chú ý đến cú pháp và chú ý đến những lỗi nhỏ. Một ký tự sai có thể mất hàng giờ để xác định trong mã dài, vì vậy điều quan trọng là phải chú ý đến cú pháp khi viết mã.

|  |
| --- |
| **Common syntax errors:** |

**Các lỗi cú pháp thường gặp:**

|  |
| --- |
| * Misspellings * Incorrect indentations * Missing or incorrect key characters:   + Bracket types - ( curved ), [ square ], { curly }   + Quote types - "straight-double" or 'straight-single', “curly-double” or ‘curly-single’   + Block introduction characters, like colons - : * Data type mismatches * Missing, incorrectly used, or misplaced Python reserved words * Using the wrong case (uppercase/lowercase) - Python is a case-sensitive language |

* Lỗi chính tả
* Thụt lề không chính xác
* Ký tự chính bị thiếu hoặc không chính xác:
  + Các loại khung - ( cong ), [ vuông ], { xoăn }
  + Các loại trích dẫn - "thẳng đôi" hoặc 'thẳng đơn', "xoăn đôi" hoặc 'xoăn đơn'
  + Chặn các ký tự giới thiệu, như dấu hai chấm - :
* Kiểu dữ liệu không khớp
* Các từ dành riêng của Python bị thiếu, sử dụng không chính xác hoặc đặt sai vị trí
* Sử dụng sai chữ hoa/chữ thường) - Python là ngôn ngữ phân biệt chữ hoa chữ thường

|  |
| --- |
| If your syntax is correct, but the script has unexpected behavior or output, this may be due to a semantic problem. Syntax is like the vocabulary, grammar, spelling, and punctuation of code. Semantics are the meaning and logic of coded statements. It is possible to have syntactically correct code that runs successfully, but doesn't do what we want it to do. |

Nếu cú ​​pháp của bạn đúng nhưng tập lệnh có hành vi hoặc kết quả đầu ra không mong muốn thì điều này có thể do vấn đề về ngữ nghĩa. Cú pháp giống như từ vựng, ngữ pháp, chính tả và dấu câu của mã. Ngữ nghĩa là ý nghĩa và logic của các câu lệnh được mã hóa. Có thể có mã đúng về mặt cú pháp chạy thành công nhưng không thực hiện được những gì chúng ta mong muốn.

|  |
| --- |
| **Common semantic errors:** |

**Các lỗi ngữ nghĩa thường gặp:**

|  |
| --- |
| * Creating functional code, but getting unintentional output * Poor logic structures in the design of the code |

* Tạo mã chức năng nhưng nhận được đầu ra không chủ ý
* Cấu trúc logic kém trong thiết kế mã

|  |
| --- |
| When working with the code blocks in exercises for this course, be mindful of syntax and semantic (logic) errors, along with the overall result of your code. Just because you fixed an error doesn't mean that the code will have the desired effect when it runs! Once you’ve fixed an error in your code, don't forget to click Run to check your work. |

Khi làm việc với các khối mã trong các bài tập của khóa học này, hãy lưu ý đến các lỗi cú pháp và ngữ nghĩa (logic), cùng với kết quả tổng thể của mã của bạn. Chỉ vì bạn đã sửa lỗi không có nghĩa là mã sẽ có tác dụng như mong muốn khi chạy! Khi bạn đã sửa lỗi trong mã của mình, đừng quên nhấp vào Chạy để kiểm tra công việc của bạn.

***3.4. Why is Python relevant to IT? – Tại sao Python lại liên quan đến CNTT?***

|  |
| --- |
| Remember how we mentioned that Python is simple and easy to use? Python makes it easy to express the fundamental concepts of programming like data structures and algorithms with easy to read syntax. This makes Python a great language to use to learn programming. And there are other reasons to pick Python, too. Python is super popular in the IT industry, making it one of the most common programming languages used today. Python isn't new. Its first version was released by Guido van Rossum back in 1991. Since then, the community that develops it has grown and the language has advanced a lot. Whenever there's a significant change to the semantics or syntax of the language, a new major version is released. |

Hãy nhớ rằng chúng tôi đã đề cập rằng Python rất đơn giản và dễ sử dụng?Python giúp bạn dễ dàng diễn đạt các khái niệm cơ bản về lập trình nhưcấu trúc dữ liệu và thuật toán với cú pháp dễ đọc.Điều này làm cho Python trở thành một ngôn ngữ tuyệt vời để học lập trình.Và còn có những lý do khác để chọn Python.Python cực kỳ phổ biến trong ngành CNTT, khiến nó trở thành một trong những ngành phổ biến nhấtngôn ngữ lập trình phổ biến được sử dụng ngày nay.Python không phải là mới. Phiên bản đầu tiên của nó được phát hành bởi Guido van Rossum vào năm1991. Kể từ đó, cộng đồng phát triển nó đã phát triển và ngôn ngữđã tiến bộ rất nhiều.Bất cứ khi nào có sự thay đổi đáng kể về ngữ nghĩa hoặc cú pháp củangôn ngữ, một phiên bản chính mới được phát hành.

|  |
| --- |
| In 2000, Python 2 was released. In 2008, we got Python 3. In this course, we'll use Python 3.7, which came out in 2018. For many years, Python was considered a beginner's language and was mostly used for teaching concepts or writing very small simple scripts, like in this course. But in recent years, the adoption of Python has grown dramatically. One reason for this is that the language has become more powerful. It's also because there's more tools available in Python for a growing range of applications. You can use Python to calculate statistics, run your e-commerce site, process images, interact with web services, and do a whole host of other tasks. Python is perfect for automation. It lets you automate everyday tasks by writing simple scripts that are easy to understand and easy to maintain. That's why Python is the language of choice for lots of people working in IT support, system administration, and web development. |

Năm 2000, Python 2 được phát hành.Năm 2008, chúng tôi có Python 3.Trong khóa học này, chúng tôi sẽ sử dụng Python 3.7, ra mắt vào năm 2018.Trong nhiều năm, Python được coi là ngôn ngữ dành cho người mới bắt đầu và được sử dụng chủ yếuđể dạy các khái niệm hoặc viết những đoạn chữ đơn giản rất nhỏ, như trong nàykhóa học. Nhưng trong những năm gần đây, việc áp dụng Python đã tăng lênđáng kể. Một lý do cho điều này là ngôn ngữ đã trở nênmạnh mẽ. Đó cũng là vì có nhiều công cụ hơn trong Python dành chophạm vi ứng dụng ngày càng tăng.Bạn có thể sử dụng Python để tính toán số liệu thống kê, điều hành trang web thương mại điện tử của mình,xử lý hình ảnh, tương tác với các dịch vụ web và thực hiện nhiều công việc khácnhiệm vụ. Python hoàn hảo cho việc tự động hóa.Nó cho phép bạn tự động hóa các công việc hàng ngày bằng cách viết các tập lệnh đơn giản, dễ thực hiện.hiểu và dễ bảo trì.Đó là lý do tại sao Python là ngôn ngữ được nhiều người làm việc trong lĩnh vực CNTT lựa chọnhỗ trợ, quản trị hệ thống và phát triển web.

|  |
| --- |
| Not only that, but it's also used in fast-growing areas of IT, like machine learning or data analytics. Last but not least, Python is available for download on a wide variety of operating systems, like Windows, Linux, and Mac OS. And what's more, Python is so popular in the workplace that if you are currently working in IT, you've most likely encountered it already. And if you're planning a career in IT, chances are you'll interact with Python quite a bit. So there's a whole lot of reasons for why Python is relevant to today's IT industry. A large part of programming is learning through trial and error and asking questions. So if at any point you get stuck, don't get discouraged. Making mistakes helps you improve. The more you see failure or broken code as an opportunity to learn, the quicker you'll master programming. I remember the first Python script I ever wrote. It took a lot of refactoring, debugging, and testing to get it to work. I relied on a lot of my teammates for help and mentorship and wound up spending more time on Stack Overflow than actually writing the code. Thankfully, you don't have to reinvent the wheel. There's almost always someone on the Internet who's trying to do what you're doing and can help point you in the right direction when you're stuck. Sometimes it takes a village. It's really important to keep in mind that even experienced programmers may need to ask a colleague a question from time to time or look something up on the Internet. Whether you're a programming novice or have some experience in software development, remember, the best programmers overcome challenges by seeking help or using other resources. Once you've completed this program, you'll be well on your way to confidently programming in basic Python. |

Không chỉ vậy, nó còn được sử dụng trong các lĩnh vực CNTT đang phát triển nhanh chóng, như máy móchọc tập hoặc phân tích dữ liệu.Cuối cùng nhưng không kém phần quan trọng, Python có sẵn để tải xuống trên nhiều nền tảng khác nhau.hệ điều hành như Windows, Linux và Mac OS.Và hơn thế nữa, Python rất phổ biến ở nơi làm việc nên nếu bạnhiện đang làm việc trong lĩnh vực CNTT, rất có thể bạn đã gặp phải nó rồi.Và nếu bạn đang dự định theo đuổi sự nghiệp trong lĩnh vực CNTT, rất có thể bạn sẽ tiếp xúc với Pythonkhá nhiều. Vì vậy, có rất nhiều lý do giải thích tại sao Python lại phù hợp vớingành CNTT ngày nay.Phần lớn việc lập trình là học thông qua việc thử, sai và hỏicâu hỏi. Vì vậy, nếu bất cứ lúc nào bạn gặp khó khăn, đừng nản lòng.Phạm sai lầm giúp bạn tiến bộ.Bạn càng coi thất bại hoặc mã bị hỏng là cơ hội để học hỏi thì bạn càng nhanh chóngbạn sẽ thành thạo lập trình.Tôi nhớ tập lệnh Python đầu tiên tôi từng viết.Phải mất rất nhiều lần tái cấu trúc, gỡ lỗi và thử nghiệm để có được nócông việc. Tôi đã dựa vào rất nhiều đồng đội của mình để được giúp đỡ và hướng dẫn và đã kết thúcdành nhiều thời gian cho Stack Overflow hơn là thực sự viết mã.Rất may, bạn không cần phải phát minh lại bánh xe.Hầu như luôn có ai đó trên Internet đang cố gắng làm điều bạn đang làmđang làm và có thể giúp bạn đi đúng hướng khi bạn gặp khó khăn.Đôi khi phải mất một ngôi làng.Điều thực sự quan trọng cần ghi nhớ là ngay cả những lập trình viên có kinh nghiệm cũng có thểthỉnh thoảng cần hỏi đồng nghiệp một câu hỏi hoặc tìm kiếm điều gì đó trên mạngInternet. Cho dù bạn là người mới lập trình hay đã có kinh nghiệm vềphát triển phần mềm, hãy nhớ rằng, những lập trình viên giỏi nhất vượt qua thử thách bằng cáchtìm kiếm sự giúp đỡ hoặc sử dụng các nguồn lực khác.Sau khi hoàn thành chương trình này, bạn sẽ sẵn sàng tiến tớitự tin lập trình Python cơ bản.

|  |
| --- |
| There's lots of information online that will help you continue to develop your programming skills. For example, there are lots of online courses for specific programming languages. You'll find answers to your Python coding questions in the official Python documentation. You can use sites like Stack Overflow to discuss and share with other developers. And you can ask questions in our discussion forums. You can even subscribe to some of the Python mailing lists to keep in the know on the latest updates to the language. You're opening the door to the whole world of programming, and it's super exciting to be joining the development community. The most important thing to remember is that you're never alone. Any questions you may have, any time in your career, there are resources out there to help you find the answers you need. Wow, that was a lot of information. Feel free to take a quick break, grab something to drink, and then head on over to the supplemental reading to learn more about Python and the resources out there to help you learn. |

Có rất nhiều thông tin trực tuyến sẽ giúp bạn tiếp tục phát triển khả năng của mình.Kỹ năng lập trình. Ví dụ: có rất nhiều khóa học trực tuyến dành riêng chongôn ngữ lập trình.Bạn sẽ tìm thấy câu trả lời cho các câu hỏi về mã hóa Python của mình trong Python chính thứctài liệu. Bạn có thể sử dụng các trang như Stack Overflow để thảo luận và chia sẻ vớicác nhà phát triển khác. Và bạn có thể đặt câu hỏi trong diễn đàn thảo luận của chúng tôi.Bạn thậm chí có thể đăng ký một số danh sách gửi thư Python để cập nhật thông tinvề các bản cập nhật mới nhất cho ngôn ngữ.Bạn đang mở ra cánh cửa cho toàn bộ thế giới lập trình và điều đó thật tuyệt vời.rất vui được tham gia cộng đồng phát triển.Điều quan trọng nhất cần nhớ là bạn không bao giờ cô đơn.Bất kỳ câu hỏi nào bạn có thể có, bất cứ lúc nào trong sự nghiệp của bạn, đều có sẵn nguồn lựcở đó để giúp bạn tìm thấy câu trả lời bạn cần.Wow, đó là rất nhiều thông tin.Hãy thoải mái nghỉ ngơi nhanh chóng, lấy thứ gì đó để uống và sau đó tiếp tụcsang phần đọc bổ sung để tìm hiểu thêm về Python vànguồn lực hiện có để giúp bạn học tập.

***3.5. Other Languages – Những ngôn ngữ khác***

|  |
| --- |
| Although we picked Python for this course, it's important to note that it's just one of the many coding languages out there. Think of a given programming language as just one of the many powerful tools in your IT toolbox. Each language has its unique set of pros and cons. Some run faster than others. Some are better suited for enterprise applications. |

Mặc dù chúng tôi đã chọn Python cho khóa học này nhưng điều quan trọng cần lưu ý là nóchỉ là một trong nhiều ngôn ngữ mã hóa hiện có.Hãy coi một ngôn ngữ lập trình nhất định chỉ là một trong nhiều công cụ mạnh mẽtrong hộp công cụ CNTT của bạn.Mỗi ngôn ngữ đều có những ưu và nhược điểm riêng.Một số chạy nhanh hơn những người khác.Một số phù hợp hơn cho các ứng dụng doanh nghiệp.

|  |
| --- |
| Others are particularly good at crunching numbers. There are platform-specific scripting languages like PowerShell which is used on Windows, and Bash which is used on Linux. Both are widely used by system administrators on those platforms. There are also general-purpose scripting languages similar to Python, like Perl or Ruby, which are also widely used for scripting and automation. JavaScript, which was originally developed as a client-side scripting language for the web, is increasingly used server-side for a broader set of tasks. And the list doesn't stop there. There's a vast array of traditional languages to explore like C, C++, Java, or Go. As you progress in your career in IT, you'll probably encounter a number of different languages and learn when to use each of them. But let's not get ahead of ourselves. First, we have Python to get our heads around. A nice feature of learning the basics of programming in one language is that you can generally apply the same concepts you learn to other languages. This means that once you're familiar with Python, you'll find it easier to pick up new coding languages as you'll spot and understand similarities and differences between them. After all, every language needs to do some common things like create variables, control the flow of a program, read input, and display output, even if they do these tasks using different approaches. As we called out earlier, learning a programming language is somewhat similar to learning a foreign language. You'll need to grasp the syntax and semantics for that language. Luckily for us, once you know the fundamentals of programming, learning another language is much easier than learning a second foreign language. There are a lot more similarities between programming languages than differences. To explore some of the similarities and differences between various scripting languages, let's take a look at a simple program that prints the words hello world ten times in three different languages, Python, Bash, and PowerShell. As you can see, each language uses a different approach to printing hello world. But look closer and you'll see similarities too. Each language must somehow put text onto the screen. The command for Python is print, for Bash it's Echo, and for PowerShell it's Write-Host. Also notice that each language has to count to ten in some way. While Python does this by specifying range(10), Bash uses a sequence notation to count from 1 to 10. PowerShell has the most complex syntax in this example, but it also boils down to starting at 1 and counting up to 10. So as we've just seen there's a whole lot of programming languages out there, but don't let that scare you. In this course, you will only need to focus on learning Python. Once you can speak Python you can go on to learn any other language you want. Up next, we've got another quiz to help you practice what you've just learned. |

Những người khác đặc biệt giỏi trong việc tính toán các con số.Có các ngôn ngữ kịch bản dành riêng cho nền tảng như PowerShell được sử dụngtrên Windows và Bash được sử dụng trên Linux.Cả hai đều được quản trị viên hệ thống sử dụng rộng rãi trên các nền tảng đó.Ngoài ra còn có các ngôn ngữ kịch bản có mục đích chung tương tự như Python,như Perl hay Ruby, cũng được sử dụng rộng rãi cho việc viết kịch bản vàtự động hóa. JavaScript, ban đầu được phát triển ở phía máy kháchngôn ngữ kịch bản cho web, ngày càng được sử dụng phía máy chủ chonhóm nhiệm vụ rộng hơn.Và danh sách không dừng lại ở đó.Có rất nhiều ngôn ngữ truyền thống để khám phá như C, C++, Java,hoặc đi. Khi bạn thăng tiến trong sự nghiệp CNTT, có thể bạn sẽ gặp phải mộtsố ngôn ngữ khác nhau và tìm hiểu khi nào nên sử dụng từng ngôn ngữ đó.Nhưng chúng ta đừng vượt quá chính mình.Đầu tiên, chúng ta có Python để xử lý.Một tính năng hay của việc học những điều cơ bản về lập trình bằng một ngôn ngữ lànhìn chung bạn có thể áp dụng những khái niệm tương tự mà bạn học được vào các ngôn ngữ khác.Điều này có nghĩa là khi bạn đã quen với Python, bạn sẽ thấy dễ dàng hơn trong việc sử dụng Python.chọn các ngôn ngữ mã hóa mới vì bạn sẽ phát hiện và hiểu những điểm tương đồng vàsự khác biệt giữa chúng.Suy cho cùng, mọi ngôn ngữ đều cần thực hiện một số việc chung như tạocác biến, điều khiển luồng chương trình, đọc đầu vào và hiển thị đầu ra,ngay cả khi họ thực hiện những nhiệm vụ này bằng các cách tiếp cận khác nhau.Như chúng tôi đã đề cập trước đó, việc học một ngôn ngữ lập trình cũng có phần tương tự như vậy.đến việc học ngoại ngữ.Bạn sẽ cần nắm được cú pháp và ngữ nghĩa của ngôn ngữ đó.Thật may mắn cho chúng tôi, một khi bạn biết các nguyên tắc cơ bản về lập trình, việc họcmột ngôn ngữ khác dễ dàng hơn nhiều so với việc học ngoại ngữ thứ hai.Có nhiều điểm tương đồng giữa các ngôn ngữ lập trình hơn làsự khác biệt. Để tìm hiểu một số điểm giống và khác nhau giữanhiều ngôn ngữ kịch bản khác nhau, chúng ta hãy xem một chương trình đơn giản innhững từ xin chào thế giới mười lần bằng ba ngôn ngữ khác nhau, Python, Bash,và PowerShell. Như bạn có thể thấy, mỗi ngôn ngữ sử dụng một cách tiếp cận khác nhau đểin xin chào thế giới.Nhưng nhìn kỹ hơn và bạn cũng sẽ thấy những điểm tương đồng.Mỗi ngôn ngữ bằng cách nào đó phải đưa văn bản lên màn hình.Lệnh cho Python là in, đối với Bash là Echo và đối với PowerShell làViết-máy chủ. Cũng lưu ý rằng mỗi ngôn ngữ phải đếm đến mười trong một sốđường. Trong khi Python thực hiện điều này bằng cách chỉ định phạm vi (10), Bash sử dụng mộtký hiệu thứ tự để đếm từ 1 đến 10.PowerShell có cú pháp phức tạp nhất trong ví dụ này, nhưng nó cũng tóm tắtbắt đầu từ 1 và đếm đến 10.Như chúng ta vừa thấy có rất nhiều ngôn ngữ lập trình,nhưng đừng để điều đó làm bạn sợ.Trong khóa học này, bạn sẽ chỉ cần tập trung vào việc học Python.Khi bạn có thể nói được Python, bạn có thể tiếp tục học bất kỳ ngôn ngữ nào khác mà bạn muốn.Tiếp theo, chúng tôi có một bài kiểm tra khác để giúp bạn thực hành những gì bạn vừa học.

***3.6. Study Guide: Introduction to Python – Hướng dẫn học: Giới thiệu về Python***

|  |
| --- |
| **Study Guide: Introduction to Python** |

**Hướng dẫn học: Giới thiệu về Python**

|  |
| --- |
| This study guide provides a quick-reference summary of what you learned in this lesson and serves as a guide for the upcoming practice quiz. |

Hướng dẫn ôn tập này cung cấp bản tóm tắt tham khảo nhanh về những gì bạn đã học trong bài học này và dùng làm hướng dẫn cho bài kiểm tra thực hành sắp tới.

|  |
| --- |
| In this segment, you learned that Python is a general purpose programming language that is commonly used for scripting and automation, as well as to develop a wide variety of applications. Python is compatible with most operating systems, including Windows, Linux, and Mac OS, and is updated every few years. Python can also run on a variety of machines, such as servers, workstations, PCs, mobile devices, IoT, and more. |

Trong phân đoạn này, bạn đã biết rằng Python là ngôn ngữ lập trình có mục đích chung thường được sử dụng để viết kịch bản và tự động hóa cũng như để phát triển nhiều ứng dụng khác nhau. Python tương thích với hầu hết các hệ điều hành, bao gồm Windows, Linux và Mac OS và được cập nhật vài năm một lần. Python cũng có thể chạy trên nhiều loại máy, chẳng hạn như máy chủ, máy trạm, PC, thiết bị di động, IoT, v.v.

|  |
| --- |
| Python is widely used in the IT field, including IT support, system administration, web development, machine learning, data analytics, and more. Python can be used to calculate statistics, run your e-commerce site, process images, interact with web services, and do a whole host of other tasks. Python instructions resemble the English language, which is what makes it easier to learn and understand when compared to other programming languages. |

Python được sử dụng rộng rãi trong lĩnh vực CNTT, bao gồm hỗ trợ CNTT, quản trị hệ thống, phát triển web, học máy, phân tích dữ liệu, v.v. Python có thể được sử dụng để tính toán số liệu thống kê, vận hành trang web thương mại điện tử của bạn, xử lý hình ảnh, tương tác với các dịch vụ web và thực hiện nhiều tác vụ khác. Các hướng dẫn của Python giống với ngôn ngữ tiếng Anh, đây là điều khiến nó dễ học và dễ hiểu hơn khi so sánh với các ngôn ngữ lập trình khác.

|  |
| --- |
| **Python is:** |

**Python là:**

|  |
| --- |
| * a general purpose scripting language; * a popular language used to code a variety of applications; * a frequently used tool for automation; * a cross-platform compatible language; * a beginner-friendly language. |

* một ngôn ngữ kịch bản có mục đích chung;
* một ngôn ngữ phổ biến được sử dụng để mã hóa nhiều loại ứng dụng;
* một công cụ được sử dụng thường xuyên để tự động hóa;
* một ngôn ngữ tương thích đa nền tảng;
* một ngôn ngữ thân thiện với người mới bắt đầu.

|  |
| --- |
| **Python is not:** |

**Python không phải là:**

|  |
| --- |
| * a platform-specific / OS-specific scripting language; * a client-side scripting language; * a purely object-oriented programming language. |

* ngôn ngữ kịch bản dành riêng cho nền tảng/hệ điều hành;
* một ngôn ngữ kịch bản phía máy khách;
* một ngôn ngữ lập trình hướng đối tượng thuần túy.

|  |
| --- |
| **Code comparison with Python** |

**So sánh mã với Python**

|  |
| --- |
| You will be learning about both Python and Bash scripting in this program. The following code illustrates a syntax difference between the two languages: |

Bạn sẽ học về cả kịch bản Python và Bash trong chương trình này. Đoạn mã sau minh họa sự khác biệt về cú pháp giữa hai ngôn ngữ:

|  |  |
| --- | --- |
| **Print to screen in Python** | **Print to screen in Bash** |
| >> print("Hello, how are you?")  Hello, how are you? | >> echo Hello, how are you?  Hello, how are you? |

|  |  |
| --- | --- |
| **In ra màn hình bằng Python** | **In ra màn hình trong Bash** |
| >> print("Hello, how are you?")  Hello, how are you? | >> echo Hello, how are you?  Hello, how are you? |

|  |
| --- |
| **Key Terms** |

**Điều khoản quan trọng**

|  |
| --- |
| * **Platform-specific / OS specific scripting language** - Platform-specific scripting languages, like PowerShell (for Windows) and Bash (for Linux), are used by system administrators on those platforms. * **Client-side scripting language** - Client-side scripting languages, like JavaScript, are used mostly for web programming. The scripts are transferred from a web server to the end-user’s internet browser, then executed in the browser. * **Machine language** - Machine language is the lowest-level computer language. It communicates directly with computing machines in binary code (ones and zeros). In binary code, one equals a pulse of electricity and zero equals no electrical pulse. Machine language instructions are made from translating languages like Python into complex patterns of ones and zeros. * **Cross-platform** **language** - Programming language that is compatible with one or more platforms / operating systems (e.g., Windows, Linux, Mac, iOS, Android). * **Object-oriented programming language** - In object-oriented programming languages, most coding elements are considered to be objects with configurable properties. For example, a form field is an object that can be configured to accept only dates as input in the mm/dd/yy format, and can be configured to read from and write to a specific database. * **Python interpreter -** An interpreter is the program that reads and executes Python code by translating Python code into computer instructions. |

* **Ngôn ngữ kịch bản dành riêng cho nền tảng/hệ điều hành** - Ngôn ngữ kịch bản dành riêng cho nền tảng, như PowerShell (dành cho Windows) và Bash (dành cho Linux), được quản trị viên hệ thống sử dụng trên các nền tảng đó.
* **Ngôn ngữ kịch bản phía máy khách** - Ngôn ngữ kịch bản phía máy khách, như JavaScript, được sử dụng chủ yếu để lập trình web. Các tập lệnh được chuyển từ máy chủ web sang trình duyệt internet của người dùng cuối, sau đó được thực thi trong trình duyệt.
* **Ngôn ngữ máy** - Ngôn ngữ máy là ngôn ngữ máy tính cấp thấp nhất. Nó giao tiếp trực tiếp với các máy tính bằng mã nhị phân (số một và số không). Trong mã nhị phân, một bằng một xung điện và 0 bằng không có xung điện. Hướng dẫn ngôn ngữ máy được tạo ra từ việc dịch các ngôn ngữ như Python thành các mẫu phức tạp gồm số 1 và số 0.
* **Ngôn ngữ đa nền tảng** - Ngôn ngữ lập trình tương thích với một hoặc nhiều nền tảng/hệ điều hành (ví dụ: Windows, Linux, Mac, iOS, Android).
* **Ngôn ngữ lập trình hướng đối tượng** - Trong các ngôn ngữ lập trình hướng đối tượng, hầu hết các phần tử mã hóa được coi là đối tượng có các thuộc tính có thể định cấu hình được. Ví dụ: trường biểu mẫu là một đối tượng có thể được định cấu hình để chỉ chấp nhận ngày làm dữ liệu đầu vào ở định dạng mm/dd/yy và có thể được định cấu hình để đọc và ghi vào cơ sở dữ liệu cụ thể.
* **Trình thông dịch Python -** Trình thông dịch là chương trình đọc và thực thi mã Python bằng cách dịch mã Python thành các lệnh máy tính.

|  |
| --- |
| **Resources** |

**Tài nguyên**

|  |
| --- |
| For additional Python practice, the following links will take you to several popular online interpreters and codepads:   * [Welcome to Python](https://www.python.org/shell/) * [Online Python Interpreter](https://www.onlinegdb.com/online_python_interpreter) * [Create a new Repl](https://repl.it/languages/python3) * [Online Python-3 Compiler (Interpreter)](https://www.tutorialspoint.com/execute_python3_online.php) * [Compile Python 3 Online](https://rextester.com/l/python3_online_compiler) * [Your Python Trinket](https://trinket.io/python3) |

Để thực hành Python bổ sung, các liên kết sau sẽ đưa bạn đến một số trình thông dịch và bảng mã trực tuyến phổ biến:

* [Chào mừng đến với Python](https://www.python.org/shell/)
* [Phiên dịch Python trực tuyến](https://www.onlinegdb.com/online_python_interpreter)
* [Tạo một bản thay thế mới](https://repl.it/languages/python3)
* [Trình biên dịch Python-3 trực tuyến (Thông dịch viên)](https://www.tutorialspoint.com/execute_python3_online.php)
* [Biên dịch Python 3 trực tuyến](https://rextester.com/l/python3_online_compiler)
* [Đồ trang sức Python của bạn](https://trinket.io/python3)

***3.7. Practice Quiz: Introduction to Python – Bài kiểm tra thực hành: Giới thiệu về Python***

**4. Hello World – Hello World**

***4.1. Review: Hello, World! – Đánh giá: Xin chào Thế giới!***

***4.2. Hello, World! – Chào thế giới!***

|  |
| --- |
| Now that you've got an idea of what Python code looks like, let's check out one of the most basic examples and dive deeper into what's going on. Get ready. We're going to use the Python interpreter to make our computer say hello to the world. |

Bây giờ bạn đã có ý tưởng về mã Python trông như thế nào, hãy cùng xemmột trong những ví dụ cơ bản nhất và đi sâu hơn vào những gì đang diễn ra.Chuẩn bị. Chúng ta sẽ sử dụng trình thông dịch Python để làm cho máy tính của chúng tanói xin chào với thế giới.

|  |
| --- |
| When we run this code either locally on our machine or on a web interpreter, the words hello world appear on the screen, just like magic. Actually it's not magic. It's because Print is a Python function that writes what we tell it to on the screen. Like the statement hello world for example, the print function is part of the basic Python language, whenever we use keywords or functions that are part of the language, we're using the programming language's syntax to tell the computer what to do. So what are functions and keywords? Functions are pieces of code that perform a unit of work. |

Khi chúng tôi chạy mã này cục bộ trên máy của mình hoặc trên trình thông dịch web,dòng chữ xin chào thế giới xuất hiện trên màn hình, giống như một phép thuật.Thực ra nó không phải là phép thuật.Đó là bởi vì Print là một hàm Python ghi những gì chúng ta yêu cầu lên trênmàn hình. Giống như câu lệnh hello world chẳng hạn, hàm print là một phầncủa ngôn ngữ Python cơ bản, bất cứ khi nào chúng ta sử dụng các từ khóa hoặc hàmmột phần của ngôn ngữ, chúng tôi đang sử dụng cú pháp của ngôn ngữ lập trình để nóimáy tính phải làm gì. Vậy chức năng và từ khóa là gì?Hàm là những đoạn mã thực hiện một đơn vị công việc.

|  |
| --- |
| We'll talk a lot more about functions later on, and you'll even learn how to write your own. Keywords are reserved words that are used to construct instructions. These words are the core part of the language and can only be used in specific ways. Some examples include if, while, and for. |

Chúng ta sẽ nói nhiều hơn về các hàm sau này và thậm chí bạn sẽ học cáchviết của riêng bạn. Từ khóa là những từ dành riêng được sử dụng để xây dựnghướng dẫn. Những từ này là phần cốt lõi của ngôn ngữ và chỉ có thể đượcđược sử dụng theo những cách cụ thể.Một số ví dụ bao gồm if, while và for.

|  |
| --- |
| We'll explain all of those and a bunch more later in the course. As we called out, the keywords and functions used in Python are what makes up the syntax of the language. Once we understand how they work, we can use them to construct more complex expressions that get the computer to do what we want it to do. Last off, notice how hello world is written between double quotation marks. Wrapping text in quotation marks indicates that the text is considered a string, which means it's text that will be manipulated by our script. In programming, any text that isn't inside quotation marks is considered part of the code. Now, for a bit of trivia, do you know why we printed the whole world in our example? Well, printing hello world has been the traditional way to start learning a programming language since way back in the '70s when it was used as the first example in a famous programming book called the C programming language. That example looked like this. |

Chúng tôi sẽ giải thích tất cả những điều đó và nhiều điều khác sau trong khóa học.Như chúng tôi đã nêu, các từ khóa và hàm được sử dụng trong Python là thứ tạo nênlên cú pháp của ngôn ngữ.Khi chúng ta hiểu cách chúng hoạt động, chúng ta có thể sử dụng chúng để xây dựng các cấu trúc phức tạp hơnnhững biểu thức khiến máy tính làm những gì chúng ta muốn.Cuối cùng, hãy chú ý cách hello world được viết giữa các dấu ngoặc kép.Việc bao văn bản trong dấu ngoặc kép biểu thị rằng văn bản đó được coi là mộtchuỗi, có nghĩa là văn bản đó sẽ được tập lệnh của chúng tôi xử lý.Trong lập trình, bất kỳ văn bản nào không nằm trong dấu ngoặc kép đều được coi làmột phần của mã. Bây giờ, để có một chút thông tin nhỏ, bạn có biết tại sao chúng tôi incả thế giới trong ví dụ của chúng tôi?Chà, in hello world là cách truyền thống để bắt đầu học mộtngôn ngữ lập trình kể từ những năm 70 khi nó được sử dụng làm ngôn ngữ đầu tiênví dụ trong cuốn sách lập trình nổi tiếng có tên là ngôn ngữ lập trình C.Ví dụ đó trông như thế này.

|  |
| --- |
| In Python, the hello world example is just one line, in C, it's three lines, in other languages, it can be even more. While learning to write hello world won't teach you the whole language, it gives you a first impression of how functions are used, and how a program written in that language looks. All right, now that we've written our first piece of Python code, I think you're ready for something a bit more challenging than hello world. Ready? Let's do it. |

Trong Python, ví dụ hello world chỉ có một dòng, trong C là ba dòng,trong các ngôn ngữ khác, nó có thể còn nhiều hơn nữa.Mặc dù việc học viết hello world sẽ không dạy bạn toàn bộ ngôn ngữ, nhưng nómang lại cho bạn ấn tượng đầu tiên về cách sử dụng các chức năng và cách một chương trìnhđược viết bằng ngôn ngữ đó trông.Được rồi, bây giờ chúng ta đã viết xong đoạn mã Python đầu tiên, tôi nghĩbạn đã sẵn sàng cho điều gì đó khó khăn hơn một chút so với xin chào thế giới.Sẵn sàng? Hãy làm nó.

***4.3. Review: Getting information from the user – Đánh giá: Lấy thông tin từ người dùng***

|  |
| --- |
| **Review: Getting information from the user** |

**Đánh giá: Lấy thông tin từ người dùng**

|  |
| --- |
| This reading contains the code used in the instructional videos from [**Getting Information from the User**](https://www.coursera.org/learn/python-crash-course/lecture/2yv1R/getting-information-from-the-user) |

Bài đọc này chứa mã được sử dụng trong các video hướng dẫn từ[**Lấy thông tin từ người dùng**](https://www.coursera.org/learn/python-crash-course/lecture/2yv1R/getting-information-from-the-user)

|  |
| --- |
| **Introduction** |

**Giới thiệu**

|  |
| --- |
| This follow-along reading is organized to match the content in the video that follows. It contains the same code shown in the next video. These code blocks will provide you with the opportunity to see how the code is written, allow you to practice running it, and can be used as a reference to refer back to. |

Phần đọc tiếp theo này được sắp xếp để phù hợp với nội dung trong video tiếp theo. Nó chứa cùng một mã được hiển thị trong video tiếp theo. Các khối mã này sẽ cung cấp cho bạn cơ hội xem mã được viết như thế nào, cho phép bạn thực hành chạy mã và có thể được sử dụng làm tài liệu tham khảo để tham khảo lại.

|  |
| --- |
| You can follow along in the reading as the instructor discusses the code or review the code after watching the video. |

Bạn có thể theo dõi bài đọc khi người hướng dẫn thảo luận về mã hoặc xem lại mã sau khi xem video.

|  |
| --- |
| name = "Brook"  print("Hello " + name)  #Hello Brook |

|  |
| --- |
| name = "Alex"  print("Hello " + name)  #Hello Alex |

***4.4. Getting Information from the User – Lấy thông tin từ người dùng***

|  |
| --- |
| On the whole, for a program to be useful it needs to get at least some information from the user. With this data, the program can take actions that are relevant to the user, instead of generic actions, like printing hello world. Data can be provided to a computer in a bunch of different ways. |

Nhìn chung, để một chương trình trở nên hữu ích, nó cần có ít nhất một sốthông tin từ người dùng.Với dữ liệu này, chương trình có thể thực hiện các hành động phù hợp với người dùng,thay vì các hành động chung chung, như in hello world.Dữ liệu có thể được cung cấp cho máy tính theo nhiều cách khác nhau.

|  |
| --- |
| For example, on a website you might input data by entering text into text fields or clicking links. If you're using a mobile application, maybe you'll click on buttons or select preferences from a drop-down menu. In a command line program, you might provide additional data by passing strings as parameters to the program, or you could have the program ask you for data interactively. All of these various platforms, programs, and apps process data differently. Some might take the contents of a file as data to be processed, others gather data from other sources and process it in the background. Remember our earlier example, when we automated the process of identifying and removing duplicate emails? |

Ví dụ: trên một trang web, bạn có thể nhập dữ liệu bằng cách nhập văn bản vào văn bảncác trường hoặc nhấp vào liên kết.Nếu bạn đang sử dụng ứng dụng di động, có thể bạn sẽ nhấp vào nút hoặc chọntùy chọn từ menu thả xuống.Trong chương trình dòng lệnh, bạn có thể cung cấp dữ liệu bổ sung bằng cách chuyểnchuỗi làm tham số cho chương trình hoặc bạn có thể yêu cầu chương trình hỏi bạncho dữ liệu có tính tương tác.Tất cả các nền tảng, chương trình và ứng dụng khác nhau đều xử lý dữ liệukhác nhau. Một số có thể lấy nội dung của một tập tin làm dữ liệuđược xử lý, những người khác thu thập dữ liệu từ các nguồn khác và xử lý nó tronglý lịch. Hãy nhớ ví dụ trước đó của chúng tôi, khi chúng tôi tự động hóa quy trìnhxác định và loại bỏ các email trùng lặp?

|  |
| --- |
| There, the data provided to the program was the list of emails, which would usually be given in a file that lists the emails one per line. |

Ở đó, dữ liệu được cung cấp cho chương trình là danh sách các email, sẽthường được cung cấp trong một tệp liệt kê các email trên mỗi dòng.

|  |
| --- |
| Whichever way your application gets the data, it will need to come from somewhere. For our first examples in this course, we'll just have the data as its own line in our block of code. This is limited, but straightforward. Later in this course, and in upcoming courses, we'll introduce you to better ways of feeding data into your code. For now though, let's see this idea in action in a very simple example. |

Cho dù ứng dụng của bạn lấy dữ liệu theo cách nào thì dữ liệu đó sẽ phải đến từmột vài nơi. Đối với các ví dụ đầu tiên của chúng tôi trong khóa học này, chúng tôi sẽ chỉ có dữ liệu dưới dạngdòng riêng của nó trong khối mã của chúng tôi.Điều này là hạn chế, nhưng đơn giản.Ở phần sau của khóa học này và trong các khóa học sắp tới, chúng tôi sẽ giới thiệu cho bạn những cách tốt hơncách đưa dữ liệu vào mã của bạn.Tuy nhiên, bây giờ chúng ta hãy xem ý tưởng này thực tế bằng một ví dụ rất đơn giản.

|  |
| --- |
| By having the name separate from the call to the print function, we're making the line of code that calls the print function generic, while still personalizing the greeting. If we then want to say hello to a different person, we only need to change the name, but the call to the print function will remain the same. Pretty simple, right? Next up, we'll learn a few other easy things that you can get Python to do for you. |

Bằng cách tách biệt tên khỏi lệnh gọi hàm in, chúng tôi đang thực hiệndòng mã gọi hàm in chung, trong khi vẫncá nhân hóa lời chào.Nếu sau đó chúng ta muốn chào một người khác, chúng ta chỉ cần thay đổitên, nhưng lệnh gọi hàm print sẽ giữ nguyên.Khá đơn giản phải không? Tiếp theo, chúng ta sẽ học một vài điều dễ dàng khác mà bạncó thể nhờ Python làm việc đó cho bạn.

***4.5. Review: Python can be your calculator – Đánh giá: Python có thể là máy tính của bạn***

|  |
| --- |
| **Review: Python can be your calculator** |

**Đánh giá: Python có thể là máy tính của bạn**

|  |
| --- |
| This reading contains the code used in the instructional videos from [**Python Can Be Your Calculator.**](https://www.coursera.org/learn/python-crash-course/lecture/y3tjE/python-can-be-your-calculator) |

Bài đọc này chứa mã được sử dụng trong các video hướng dẫn từ[**Python có thể là máy tính của bạn**](https://www.coursera.org/learn/python-crash-course/lecture/y3tjE/python-can-be-your-calculator)

|  |
| --- |
| **Introduction** |

**Giới thiệu**

|  |
| --- |
| This follow-along reading is organized to match the content in the video that follows. It contains the same code shown in the next video.  These code blocks will provide you with the opportunity to see how the code is written, allow you to practice running it, and can be used as a reference to refer back to. |

Phần đọc tiếp theo này được sắp xếp để phù hợp với nội dung trong video tiếp theo. Nó chứa cùng một mã được hiển thị trong video tiếp theo. Các khối mã này sẽ cung cấp cho bạn cơ hội xem mã được viết như thế nào, cho phép bạn thực hành chạy mã và có thể được sử dụng làm tài liệu tham khảo để tham khảo lại.

|  |
| --- |
| You can follow along in the reading as the instructor discusses the code or review the code after watching the video. |

Bạn có thể theo dõi bài đọc khi người hướng dẫn thảo luận về mã hoặc xem lại mã sau khi xem video.

|  |
| --- |
| print(4+5)  #9  print(9\*7)  #63  print(-1/4)  #-0.25 |

|  |
| --- |
| print(1/3)  #0.3333333333333333 |

|  |
| --- |
| print(((2050/5)-32)/9)  #42.0 |

|  |
| --- |
| print(2\*\*10)  #1024 |

***4.6. Python Can Be Your Calculator – Python có thể là máy tính của bạn***

|  |
| --- |
| There's a ton of things that you can do with Python and you'll learn many of them in this course. But before we dive into complex subjects, let's have some fun with another simple task that you can do with Python. We are going to make Python our calculator. |

Có rất nhiều thứ bạn có thể làm với Python và bạn sẽ học được nhiều điềuchúng trong khóa học này. Nhưng trước khi chúng ta đi sâu vào các chủ đề phức tạp, chúng ta hãy có một sốthú vị với một tác vụ đơn giản khác mà bạn có thể thực hiện với Python.Chúng tôi sẽ biến Python thành máy tính của chúng tôi.

|  |
| --- |
| Let's start with something easy. So 4 plus 5 is 9, 9 times 7 is 63, minus 1 divided by 4 is minus 0.25. Easy. Repeating or periodic numbers are printed in a longer format. Let's try 1 divided by 3. |

Hãy bắt đầu với một cái gì đó dễ dàng.Vậy 4 cộng 5 là 9, 9 nhân 7 là 63, trừ 1 chia cho 4 là trừ 0,25.Dễ. Các số lặp lại hoặc định kỳ được in ở định dạng dài hơn.Hãy thử 1 chia cho 3.

|  |
| --- |
| In math theory, when 1 is divided by 3, the digit 3 repeats forever after the decimal point. Of course, it's hard to display something that repeats forever. So instead, we have a representation showing lots of decimal places. Not too hard, right? Let's get the computer something a bit trickier. Let's say we want to divide 2050 by 5, then subtract 32 and then divide the result by 9. To do this, we'll need to use parentheses, just as we do in typical math problems. You can also use Python to get squares, cubes, or any power of n of a number. For example, let's say we want to find out what 2 to the power of 10 is. To get Python to give us the answer, we use the double star operator. If you're starting to worry that this is becoming an algebra course, relax. We're not going to do anything more complex than what we've just seen. If you're thinking, "Why would I use Python instead of just a normal calculator?" That's a valid question. Experimenting in this way, you get familiar with the language's math capabilities. In IT jobs, there are many tasks that require you to use math calculations. You might need to count how many times a certain word appears in a text, or work out the average time it takes for an operation to complete, or how much you have to compress an image to fit in certain size constraints. Whatever you need to calculate, writing a script can help you do it faster and with more accuracy. So you need to know what mathematical operations are available to you. Python actually has a lot more advanced numeric capabilities that are used for data analysis, statistics, machine learning, and other scientific applications. We won't get into these in this course. But if you want to learn more about them on your own, there's a wealth of online resources available. Next up, a cheat sheet to help you with programming concepts that we've just covered. After that, it's time for another quiz. This time with a few small coding exercises. |

Trong lý thuyết toán học, khi chia 1 cho 3 thì chữ số 3 lặp lại mãi mãi sau sốdấu thập phân. Tất nhiên, thật khó để thể hiện điều gì đó lặp đi lặp lại mãi mãi.Vì vậy, thay vào đó, chúng ta có một biểu diễn hiển thị nhiều chữ số thập phân.Không quá khó phải không? Hãy lấy máy tính thứ gì đó phức tạp hơn một chút.Giả sử chúng ta muốn chia 2050 cho 5, sau đó trừ 32 rồi chiakết quả là 9. Để làm điều này, chúng ta cần sử dụng dấu ngoặc đơn, giống như cách chúng ta làm trongcác bài toán điển hình.Bạn cũng có thể sử dụng Python để lấy hình vuông, hình khối hoặc lũy thừa n của một số.Ví dụ: giả sử chúng ta muốn tìm hiểu 2 mũ 10 là bao nhiêu.Để Python đưa ra câu trả lời, chúng tôi sử dụng toán tử sao đôi.Nếu bạn bắt đầu lo lắng rằng đây đang trở thành một khóa học đại số, hãy thư giãn.Chúng ta sẽ không làm bất cứ điều gì phức tạp hơn những gì chúng ta vừa thấy.Nếu bạn đang nghĩ, "Tại sao tôi lại sử dụng Python thay vì chỉ sử dụng một ngôn ngữ bình thường?máy tính?" Đó là một câu hỏi hợp lệ.Thử nghiệm theo cách này, bạn sẽ làm quen với phép toán của ngôn ngữkhả năng. Trong công việc IT có rất nhiều công việc yêu cầu bạn phải sử dụng đến toán họctính toán. Bạn có thể cần đếm số lần một từ nhất định xuất hiện trongmột văn bản hoặc tính ra thời gian trung bình cần thiết để hoàn thành một thao tác hoặcbạn phải nén hình ảnh bao nhiêu để phù hợp với các giới hạn kích thước nhất định.Bất cứ điều gì bạn cần tính toán, việc viết một đoạn script có thể giúp bạn thực hiện nó nhanh hơn vàvới độ chính xác cao hơn. Vậy bạn cần biết các phép toán là gìcó sẵn cho bạn. Python thực sự có nhiều khả năng số nâng cao hơnđược sử dụng để phân tích dữ liệu, thống kê, học máy và các hoạt động khácứng dụng khoa học.Chúng ta sẽ không đề cập đến những điều này trong khóa học này.Nhưng nếu bạn muốn tự mình tìm hiểu thêm về chúng, có rất nhiềunguồn tài nguyên trực tuyến có sẵn.Tiếp theo, một bảng tóm tắt để giúp bạn về các khái niệm lập trình mà chúng ta vừa mớiđề cập. Sau đó là lúc cho một bài kiểm tra khác.Lần này với một vài bài tập viết mã nhỏ.

|  |
| --- |
| Remember, if something is unclear, you can re-watch the videos as many times as you need. Ready? You've got this. |

Hãy nhớ rằng, nếu có điều gì chưa rõ, bạn có thể xem lại video nhiều lần tùy thích.bạn cần. Sẵn sàng?Bạn đã có cái này.

***4.7. Study Guide: First Programming Concepts – Hướng dẫn học: Các khái niệm lập trình đầu tiên***

|  |
| --- |
| **Study Guide: First Programming Concepts** |

**Hướng dẫn học: Các khái niệm lập trình đầu tiên**

|  |
| --- |
| This study guide provides a quick-reference summary of what you learned in this lesson and serves as a guide for the upcoming practice quiz. |

Hướng dẫn ôn tập này cung cấp bản tóm tắt tham khảo nhanh về những gì bạn đã học trong bài học này và dùng làm hướng dẫn cho bài kiểm tra thực hành sắp tới.

|  |
| --- |
| **Functions** |

**Chức năng**

|  |
| --- |
| A function is a piece of code that performs a unit of work. In the examples you've seen so far, you have only encountered the **print()** function, which outputs a message to the screen. You will use this function frequently in this course to check the results of your code. The syntax of the print() function is modeled in the example below. |

Hàm là một đoạn mã thực hiện một đơn vị công việc. Trong các ví dụ bạn đã thấy cho đến nay, bạn chỉ gặp hàm **print()** , hàm này xuất thông báo ra màn hình. Bạn sẽ sử dụng hàm này thường xuyên trong khóa học này để kiểm tra kết quả mã của mình. Cú pháp của hàm print() được mô phỏng trong ví dụ dưới đây.

|  |
| --- |
| # Syntax for printing a string of text. Click Run to check the result.  print("Hello world!")  # Hello world! |

|  |
| --- |
| **Keywords** |

**Từ khóa**

|  |
| --- |
| A keyword is a reserved word in a programming language that performs a specific purpose. In your first Python example, you briefly encountered the keywords **for** and **in**. Note that keywords will often appear in **bold** in this course. |

Từ khóa là một từ dành riêng trong ngôn ngữ lập trình nhằm thực hiện một mục đích cụ thể. Trong ví dụ Python đầu tiên, bạn đã gặp nhanh các từ khóa **for** và **in** . Lưu ý rằng các từ khóa thường sẽ được in **đậm** trong khóa học này.

|  |
| --- |
| In the next few weeks, you will also learn the following keywords: |

Trong vài tuần tới, bạn cũng sẽ học các từ khóa sau:

|  |
| --- |
| Values: **True**, **False**, **None**  Conditions: **if**, **elif**, **else**  Logical operators: **and**, **or**, **not**  Loops: **for**, **in**, **while**, **break**, **continue**  Functions: **def**, **return** |

Giá trị: **Đúng** , **Sai** , **Không có** Điều kiện: **if** , **elif** , **else** Toán tử logic: **and** , **or** , **not** Vòng lặp: **for** , **in** , **while** , **break** , **continue** Hàm: **def** , **return**

|  |
| --- |
| You don't need to learn this whole list now. We'll dive into each keyword as we encounter them. There are additional reserved keywords in Python. If you would like to read about them, please visit the linked “Python Keywords” article in the Resources section at the end of this study guide. |

Bạn không cần phải học toàn bộ danh sách này ngay bây giờ. Chúng tôi sẽ đi sâu vào từng từ khóa khi chúng tôi gặp chúng. Có thêm các từ khóa dành riêng trong Python. Nếu bạn muốn đọc về chúng, vui lòng truy cập bài viết “Từ khóa Python” được liên kết trong phần Tài nguyên ở cuối hướng dẫn nghiên cứu này.

|  |
| --- |
| **Arithmetic operators** |

**Toán tử số học**

|  |
| --- |
| Python can calculate numbers using common mathematical operators, along with some special operators, too: |

Python có thể tính toán các số bằng cách sử dụng các toán tử toán học thông thường, cùng với một số toán tử đặc biệt:

|  |
| --- |
| **x + y**            Addition + operator returns the sum of x plus y  **x - y**             Subtraction - operator returns the difference of x minus y  **x \* y**            Multiplication \* operator returns the product of x times y  **x / y**             Division / operator returns the quotient of x divided by y  **x\*\*y**            Exponent \*\* operator returns the result of raising x to the power of y  **x\*\*2**            Square expression returns x squared  **x\*\*3**            Cube expression returns x cubed  **x\*\*(1/2)**   Square root (½) or (0.5) fractional exponent operator returns the square root of x  **x // y**           Floor division operator returns the integer part of the integer division of x by y  **x % y**          Modulo operator returns the remainder part of the integer division of x by y |

**x + y**            Toán tử cộng + trả về tổng của x cộng y

**x - y**             Toán tử trừ - trả về hiệu của x trừ y

**x \* y**             Toán tử nhân \* trả về tích của x nhân y

**x / y**             Toán tử chia/y trả về thương của x chia cho y

**x\*\*y**             Toán tử mũ \*\* trả về kết quả của việc nâng x lên lũy thừa của y

**x\*\*2**             Biểu thức bình phương trả về x bình phương

**x\*\*3**             Biểu thức lập phương trả về x lập phương

**x\*\*(1/2)** Căn bậc hai (½) hoặc (0,5) toán tử số mũ phân số trả về căn bậc hai của x

**x // y**           Toán tử chia tầng trả về phần nguyên của phép chia số nguyên của x cho y

**x % y**           Toán tử Modulo trả về phần còn lại của phép chia số nguyên của x bởi y

|  |
| --- |
| **Order of operations** |

**Trình tự thao tác**

|  |
| --- |
| The order of operations are to be calculated from left to right in the following order: |

Thứ tự thực hiện các phép tính từ trái qua phải theo thứ tự sau:

|  |
| --- |
| 1. **P**arentheses ( ), { }, [ ] 2. **E**xponents xy   (x\*\*y) 3. **M**ultiplication \* and **D**ivision / 4. **A**ddition + and **S**ubtraction - |

1. Các mệnh đề ( ), { }, [ ]
2. E số mũ x y   (x\*\*y)
3. Phép nhân \* và Phép chia /
4. Phép cộng + và trừ -

|  |
| --- |
| You might find the **PEMDAS** mnemonic device to be helpful in remembering the order. |

Bạn có thể thấy thiết bị ghi nhớ **PEMDAS** hữu ích trong việc ghi nhớ đơn hàng.

|  |
| --- |
| **Resources for more information** |

**Tài nguyên để biết thêm thông tin**

|  |
| --- |
| For more information about the concepts covered in this reading, please visit:   * [Built-in Functions](https://docs.python.org/3/library/functions.html) - Lists and summarizes Python’s built-in functions. * [Python Keywords](https://www.w3schools.com/python/python_ref_keywords.asp) - Lists Python’s reserved keywords and a brief description of what each keyword does. * [Different Arithmetic operators in Python](https://flexiple.com/python/arithmetic-operators-in-python/) - Provides more examples of the proper syntax for using arithmetic operators in Python. |

Để biết thêm thông tin về các khái niệm được đề cập trong bài đọc này, vui lòng truy cập:

* [Chức năng tích hợp sẵn](https://docs.python.org/3/library/functions.html)- Liệt kê và tóm tắt các hàm có sẵn của Python.
* [Từ khóa Python](https://www.w3schools.com/python/python_ref_keywords.asp)- Liệt kê các từ khóa dành riêng của Python và mô tả ngắn gọn về chức năng của từng từ khóa.
* [Các toán tử số học khác nhau trong Python](https://flexiple.com/python/arithmetic-operators-in-python/)- Cung cấp thêm ví dụ về cú pháp thích hợp khi sử dụng toán tử số học trong Python.

|  |
| --- |
| For additional Python practice, the following links will take you to several popular online interpreters and codepads:   * [Welcome to Python](https://www.python.org/shell/) * [Online Python Interpreter](https://www.onlinegdb.com/online_python_interpreter) * [Create a new Repl](https://repl.it/languages/python3) * [Online Python-3 Compiler (Interpreter)](https://www.tutorialspoint.com/execute_python3_online.php) * [Compile Python 3 Online](https://rextester.com/l/python3_online_compiler) * [Your Python Trinket](https://trinket.io/python3) |

Để thực hành Python bổ sung, các liên kết sau sẽ đưa bạn đến một số trình thông dịch và bảng mã trực tuyến phổ biến:

* [Chào mừng đến với Python](https://www.python.org/shell/)
* [Phiên dịch Python trực tuyến](https://www.onlinegdb.com/online_python_interpreter)
* [Tạo một bản thay thế mới](https://repl.it/languages/python3)
* [Trình biên dịch Python-3 trực tuyến (Thông dịch viên)](https://www.tutorialspoint.com/execute_python3_online.php)
* [Biên dịch Python 3 trực tuyến](https://rextester.com/l/python3_online_compiler)
* [Đồ trang sức Python của bạn](https://trinket.io/python3)

***4.8. Practice Quiz: Hello World – Bài kiểm tra thực hành: Xin chào thế giới***

***4.9. Code editors and IDEs overview – Tổng quan về trình soạn thảo mã và IDE***

|  |
| --- |
| - Now that we've covered some Python coding basics, we'll introduce you to some of the coding tools. Coding tools can help you develop, write, debug and visualize your code. In this video, we'll cover the basics of code editors and Integrated Development Environments or IDEs. |

- Bây giờ chúng ta đã trình bày một số kiến ​​thức cơ bản về mã hóa Python,chúng tôi sẽ giới thiệu cho bạn một số công cụ mã hóa.Các công cụ mã hóa có thể giúp bạn phát triển, viết, gỡ lỗivà trực quan hóa mã của bạn.Trong video này, chúng tôi sẽ đề cập đến những điều cơ bản về trình soạn thảo mãvà Môi trường phát triển tích hợp hoặc IDE.

|  |
| --- |
| Code editors are tools used to make writing code easier. With a code editor, you can write, debug and execute programs in Python. Code editors provide features, including syntax highlighting, automatic indentation, error checking, and auto-completion. Overall, code editors help define the code structure and function. This allows you to write code more efficiently. They also make it easier to understand variables, commands, functions, and loops. VS Code, Jupyter Notebooks and CoLab are some of the code editors we will cover in the next few videos. IDEs have more features than code editors. The difference between a code editor and an IDE is that a code editor is like a text editor. IDEs provide direction in writing code and allow you to see how the code executes. You can think of a code editor like a landline phone. You can dial numbers and make phone calls with it. An IDE is more like a smartphone. You can still make phone calls, but it has additional features that allow you to make video calls, send text messages, browse the Internet and use lots of other applications. An IDE is a software tool that simplifies the process of creating a new software application. An IDE is a software application that provides comprehensive facilities for software development. IDEs always include a code editor. IDEs allow you to develop code more efficiently by bringing multiple tools into one environment. You'll be able to edit, build, test and package all in one application. They also allow you to program applications more quickly without having to add manual integration and configuration. Coming up, we will cover IDLE. IDLE is an IDE that comes with Python automatically. It's an excellent beginner level IDE because it's very simple to use. It may not be great for larger projects, but it's excellent for getting to know how to use an IDE. Now, let's dive into some of the code editors and IDEs that you can use as you continue in your Python learning journey. |

Trình chỉnh sửa mã là công cụ được sử dụng để giúp việc viết mã dễ dàng hơn.Với trình soạn thảo mã,bạn có thể viết, gỡ lỗi và thực thi các chương trình bằng Python.Trình chỉnh sửa mã cung cấp các tính năng, bao gồm tô sáng cú pháp,tự động thụt lề, kiểm tra lỗi,và tự động hoàn thành.Nhìn chung, trình soạn thảo mã giúp xác định cấu trúc mãvà chức năng.Điều này cho phép bạn viết mã hiệu quả hơn.Chúng cũng làm cho việc hiểu các biến trở nên dễ dàng hơn,lệnh, hàm và vòng lặp.Mã VS, Sổ ghi chép Jupyter và CoLablà một số trình soạn thảo mã mà chúng tôi sẽ đề cập đếntrong vài video tiếp theo.IDE có nhiều tính năng hơn trình soạn thảo mã.Sự khác biệt giữa trình soạn thảo mãvà IDE là trình soạn thảo mã giống như trình soạn thảo văn bản.IDE cung cấp hướng dẫn viết mãvà cho phép bạn xem mã thực thi như thế nào.Bạn có thể coi trình soạn thảo mã giống như điện thoại cố định.Bạn có thể quay số và gọi điện thoại với nó.IDE giống một chiếc điện thoại thông minh hơn.Bạn vẫn có thể gọi điện thoại,nhưng nó có các tính năng bổ sungcho phép bạn thực hiện cuộc gọi video, gửi tin nhắn văn bản,duyệt Internet và sử dụng nhiều ứng dụng khác.IDE là một công cụ phần mềm giúp đơn giản hóa quy trìnhviệc tạo ra một ứng dụng phần mềm mới.IDE là một ứng dụng phần mềmcung cấp cơ sở vật chất toàn diệncho việc phát triển phần mềm.IDE luôn bao gồm trình soạn thảo mã.IDE cho phép bạn phát triển mã hiệu quả hơnbằng cách đưa nhiều công cụ vào một môi trường.Bạn sẽ có thể chỉnh sửa, xây dựng, kiểm travà đóng gói tất cả trong một ứng dụng.Chúng cũng cho phép bạn lập trình ứng dụng nhanh hơnmà không cần phải thêm tích hợp và cấu hình thủ công.Sắp tới, chúng tôi sẽ đề cập đến IDLE.IDLE là một IDE tự động đi kèm với Python.Đó là một IDE cấp độ mới bắt đầu tuyệt vờibởi vì nó rất đơn giản để sử dụng.Nó có thể không tuyệt vời cho các dự án lớn hơn,nhưng thật tuyệt vời khi biết cách sử dụng IDE.Bây giờ, hãy đi sâu vào một số trình soạn thảo mãvà IDE mà bạn có thể sử dụngkhi bạn tiếp tục hành trình học Python của mình.

***4.10. Review: Use the command-line – Đánh giá: Sử dụng dòng lệnh***

|  |
| --- |
| **Review: Use the command-line** |

**Đánh giá: Sử dụng dòng lệnh**

|  |
| --- |
| The command-line is used to tell your computer what to do. You can use it to access servers, move files, change directories, and write scripts. In this reading, you will learn how to write Python scripts at the command-line along with the Python GUI IDLE. We will also cover the different ways you can access the command-line based on your operating system. |

Dòng lệnh được sử dụng để báo cho máy tính của bạn biết phải làm gì. Bạn có thể sử dụng nó để truy cập máy chủ, di chuyển tệp, thay đổi thư mục và viết tập lệnh. Trong bài đọc này, bạn sẽ tìm hiểu cách viết tập lệnh Python bằng dòng lệnh cùng với Python GUI IDLE. Chúng tôi cũng sẽ đề cập đến các cách khác nhau để bạn có thể truy cập dòng lệnh dựa trên hệ điều hành của mình.

|  |
| --- |
| **Using the command line on MacOS** |

**Sử dụng dòng lệnh trên MacOS**

|  |
| --- |
| Using the spotlight search, type in “terminal.” Select the terminal application. You should see your username followed by the $ sign. MacOS comes with Python 2.7 pre-installed. You can install Python 3 from [python.org](http://python.org/). Just remember, this means that you will have 2 versions of Python installed on your Mac, and you will need to pay special attention to your paths. |

Sử dụng tìm kiếm tiêu điểm, nhập “thiết bị đầu cuối”. Chọn ứng dụng đầu cuối. Bạn sẽ thấy tên người dùng của mình theo sau là dấu $. MacOS được cài đặt sẵn Python 2.7. Bạn có thể cài đặt Python 3 từ[python.org](http://python.org/). Chỉ cần nhớ, điều này có nghĩa là bạn sẽ cài đặt 2 phiên bản Python trên máy Mac và bạn sẽ cần đặc biệt chú ý đến đường dẫn của mình.

|  |
| --- |
| To check which version of Python you have installed on your Mac, use the following command.  **python --version** |

Để kiểm tra phiên bản Python nào bạn đã cài đặt trên máy Mac, hãy sử dụng lệnh sau.

**python --version**

|  |
| --- |
| To check for Python3, use the following command.  **python3 --version** |

Để kiểm tra Python3, hãy sử dụng lệnh sau.

**python3 --version**

|  |
| --- |
| **Using the command line on Windows OS** |

**Sử dụng dòng lệnh trên hệ điều hành Windows**

|  |
| --- |
| In Windows, open the start menu. In the search box type cmd. Right-click on cmd and select Run as administrator. This will open up the command-line. Windows OS does not come with Python installed. Visit the official Python [download page](https://www.python.org/downloads/windows/) for Windows. Select the Windows installer (64-bit) or (32-bit). After the installer is downloaded, double–click the file. Select **Install launcher for all users**. Follow the prompts during installation. Make sure to select the **Add python.exe PATH** checkbox. This will allow you to launch Python from the command line. Once installation is complete, you can check for Python from the command line. |

Trong Windows, mở menu bắt đầu. Trong hộp tìm kiếm gõ cmd. Nhấp chuột phải vào cmd và chọn Chạy với tư cách quản trị viên. Điều này sẽ mở ra dòng lệnh. Hệ điều hành Windows không được cài đặt sẵn Python. Truy cập Python chính thức[trang tải xuống](https://www.python.org/downloads/windows/)cho cửa sổ. Chọn trình cài đặt Windows (64-bit) hoặc (32-bit). Sau khi tải xuống trình cài đặt, bấm đúp vào tệp. Chọn **Cài đặt trình khởi chạy cho tất cả người dùng** . Làm theo lời nhắc trong khi cài đặt. Đảm bảo chọn hộp kiểm **Thêm python.exe PATH** . Điều này sẽ cho phép bạn khởi chạy Python từ dòng lệnh. Sau khi cài đặt hoàn tất, bạn có thể kiểm tra Python từ dòng lệnh.

|  |
| --- |
| To check for Python, use the following command. The version of Python you installed will appear.  **python --version** |

Để kiểm tra Python, hãy sử dụng lệnh sau. Phiên bản Python bạn đã cài đặt sẽ xuất hiện.

**python --version**

|  |
| --- |
| **Using the command line on Linux OS** |

**Sử dụng dòng lệnh trên hệ điều hành Linux**

|  |
| --- |
| Access the Linux terminal using Ctrl + Alt + T. This will allow you to check for Python. Type python. Python comes preinstalled on most Linux systems. If the command is not found, you can install Python by writing sudo apt install python3. |

Truy cập thiết bị đầu cuối Linux bằng Ctrl + Alt + T. Điều này sẽ cho phép bạn kiểm tra Python. Gõ trăn. Python được cài đặt sẵn trên hầu hết các hệ thống Linux. Nếu không tìm thấy lệnh, bạn có thể cài đặt Python bằng cách viết sudo apt install python3.

|  |
| --- |
| You can begin writing Python code from the terminal. Simply type python to use the interactive mode. You can also write Python scripts using Linux with IDLE which we will cover next. |

Bạn có thể bắt đầu viết mã Python từ thiết bị đầu cuối. Chỉ cần gõ python để sử dụng chế độ tương tác. Bạn cũng có thể viết tập lệnh Python bằng Linux bằng IDLE mà chúng tôi sẽ đề cập tiếp theo.

|  |
| --- |
| **Using IDLE** |

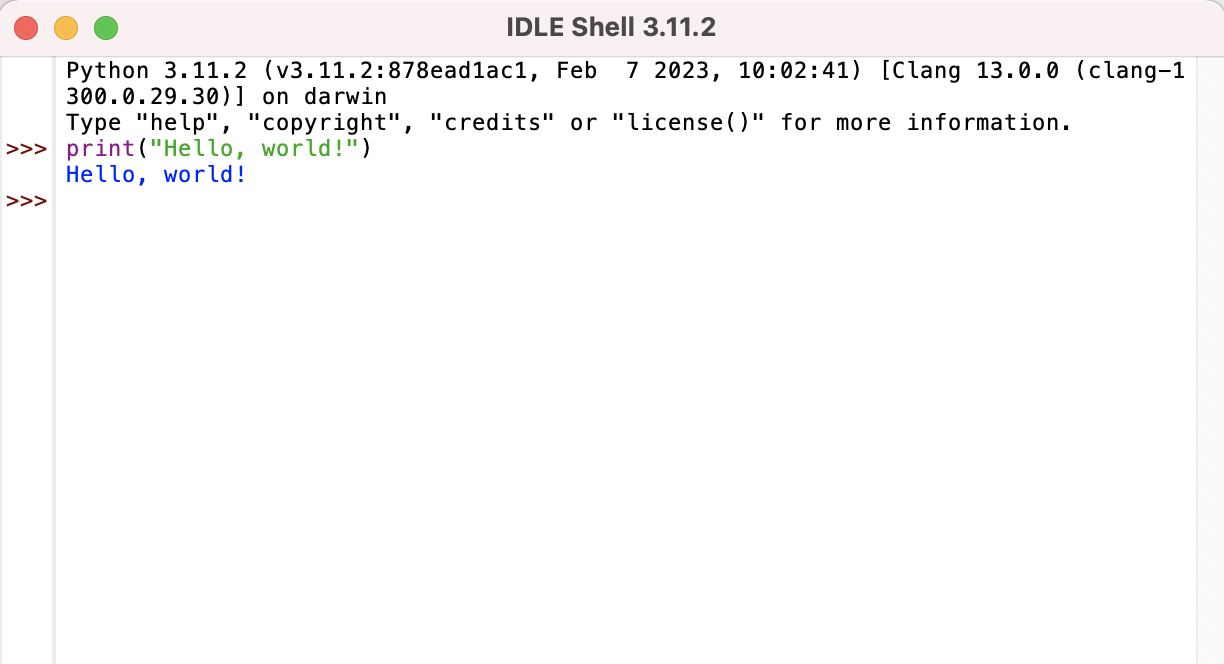
**Sử dụng IDLE**

|  |
| --- |
| Python IDLE is included with Python installations on Windows and MacOS. You can download IDLE using your package manager on Linux. Python IDLE is an interactive interpreter or file editor that allows you to easily write Python scripts and programs. IDLE provides syntax highlighting, code completion, and automatic indentation. |

Python IDLE được bao gồm trong bản cài đặt Python trên Windows và MacOS. Bạn có thể tải xuống IDLE bằng trình quản lý gói trên Linux. Python IDLE là trình thông dịch hoặc trình chỉnh sửa tệp tương tác cho phép bạn dễ dàng viết các tập lệnh và chương trình Python. IDLE cung cấp tính năng tô sáng cú pháp, hoàn thiện mã và thụt lề tự động.

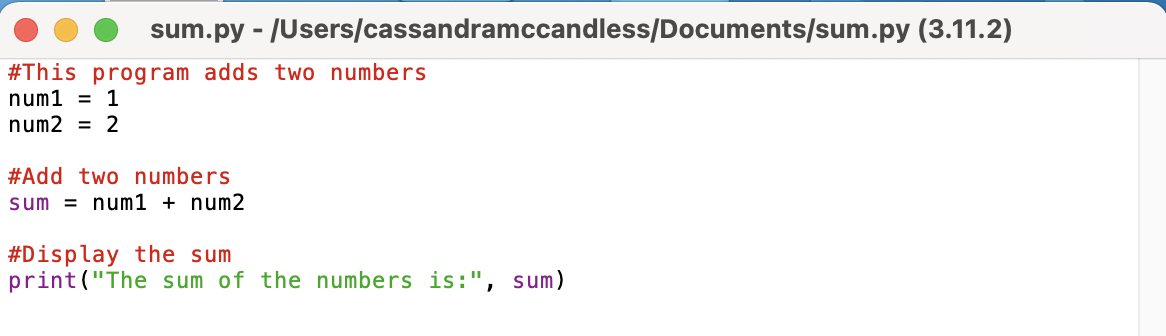
|  |
| --- |
| Double click on the IDLE icon to open it on your computer. This will open a blank Python interpreter window. You can begin writing code right away. |

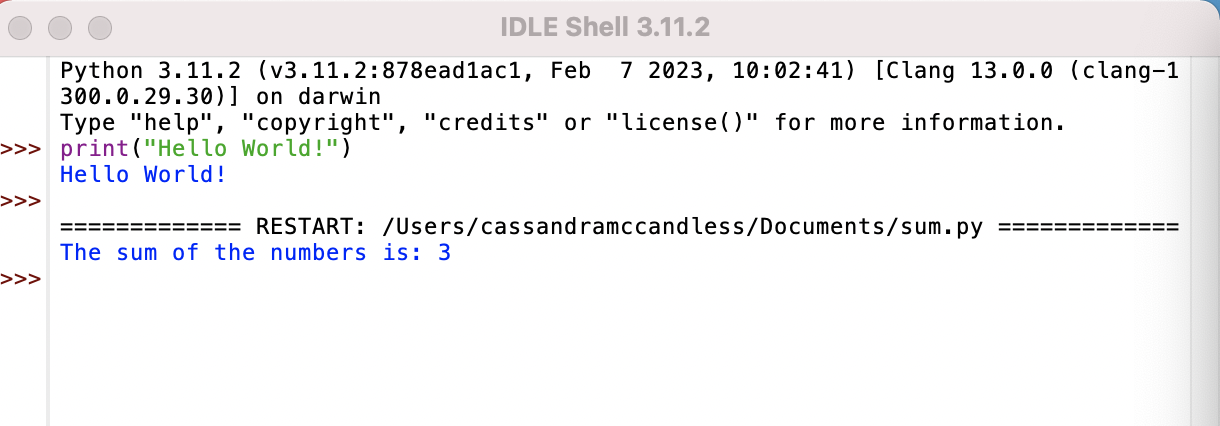
Nhấp đúp chuột vào biểu tượng IDLE để mở nó trên máy tính của bạn. Thao tác này sẽ mở một cửa sổ trình thông dịch Python trống. Bạn có thể bắt đầu viết mã ngay lập tức.



|  |
| --- |
| You can also open a new file. Go to File → Open → New File from the menu bar. Here you can write a Python file. Once you have completed writing your Python code in the file, go to File → Save As. Give your Python file a name. Hit Save. To run the Python code in the file you saved click Run → Run Module from the menu. |

Bạn cũng có thể mở một tập tin mới. Đi tới File → Open → New File từ thanh menu. Tại đây bạn có thể viết một tệp Python. Khi bạn đã viết xong mã Python vào tệp, hãy đi tới Tệp → Lưu dưới dạng. Đặt tên cho tệp Python của bạn. Nhấn Lưu. Để chạy mã Python trong tệp bạn đã lưu, hãy nhấp vào Chạy → Chạy Mô-đun từ menu.





|  |
| --- |
| **Key takeaways** |

**Bài học chính**

|  |
| --- |
| Whichever operating system you are using, you will be able to run Python from the command-line.  Using a text editor like IDLE and running python from the command-line  is best for executing and debugging individual scripts or .py files. |

Cho dù bạn đang sử dụng hệ điều hành nào, bạn đều có thể chạy Python từ dòng lệnh. Sử dụng trình soạn thảo văn bản như IDLE và chạy python từ dòng lệnh là cách tốt nhất để thực thi và gỡ lỗi các tập lệnh hoặc tệp .py riêng lẻ.

|  |
| --- |
| **Resources for more information** |

**Tài nguyên để biết thêm thông tin**

|  |
| --- |
| Here is a list of additional resources for writing and running Python on your local machine.   * A [guide](https://towardsdatascience.com/a-quick-guide-to-using-command-line-terminal-96815b97b955) to using terminal (command-line) on Mac, Windows, and Linux operating systems. * [IDLE documentation and instructions](https://docs.python.org/3/library/idle.html) * [Python Scripts vs. Jupyter Notebooks](https://learnpython.com/blog/python-scripts-vs-jupyter-notebooks/) |

Đây là danh sách các tài nguyên bổ sung để viết và chạy Python trên máy cục bộ của bạn.

* MỘT[hướng dẫn](https://towardsdatascience.com/a-quick-guide-to-using-command-line-terminal-96815b97b955)sang sử dụng terminal (dòng lệnh) trên hệ điều hành Mac, Windows và Linux.
* [Tài liệu và hướng dẫn IDLE](https://docs.python.org/3/library/idle.html)
* [Tập lệnh Python so với sổ ghi chép Jupyter](https://learnpython.com/blog/python-scripts-vs-jupyter-notebooks/)

***4.11. Use the command-line – Sử dụng dòng lệnh***

|  |
| --- |
| - Let's get started with our first coding tool. In this video, you'll learn how to write Python scripts at the command line, along with the Python GUI, Idle. The command line is used to tell your computer what to do. You can use it to access servers, move files, change directories, and write scripts. In this video, we'll show you how to use the command line and Idle on the Linux operating system. If you're using another operating system, you can refer to the reading titled, "Use the command line." Now, let's go over the command line on Linux OS. Access the Linux terminal by clicking on it or using CTRL, ALT, T. This will allow you to check for Python. Type Python. Python comes pre-installed on most Linux systems. If the command is not found, you can install Python by writing sudo apt install python3. Next, it will ask you for your password. Type your password and press enter. Python version three will begin installation. Once installation is complete, you can begin writing Python code from the terminal. Simply type python3 to use the interactive mode. Let's try out interactive mode using a print statement. Type print Hello World, and press enter. You will see the output, "Hello World." To exit interactive mode, we will type exit(). |

- Hãy bắt đầu với công cụ mã hóa đầu tiên của chúng ta.Trong video này, bạn sẽ học cách viết tập lệnh Pythontại dòng lệnh, cùng với GUI Python, Idle.Dòng lệnh được sử dụng để báo cho máy tính của bạn biết phải làm gì.Bạn có thể sử dụng nó để truy cập máy chủ, di chuyển tập tin,thay đổi thư mục và viết tập lệnh.Trong video này, chúng tôi sẽ hướng dẫn bạn cách sử dụng dòng lệnhvà Nhàn rỗi trên hệ điều hành Linux.Nếu bạn đang sử dụng hệ điều hành khác,bạn có thể tham khảo bài đọc có tiêu đề "Sử dụng dòng lệnh."Bây giờ chúng ta hãy xem qua dòng lệnh trên hệ điều hành Linux.Truy cập thiết bị đầu cuối Linux bằng cách nhấp vào nóhoặc sử dụng CTRL, ALT, T.Điều này sẽ cho phép bạn kiểm tra Python.Gõ Python.Python được cài đặt sẵn trên hầu hết các hệ thống Linux.Nếu không tìm thấy lệnh, bạn có thể cài đặt Pythonbằng cách viết Sudo apt install python3.Tiếp theo, nó sẽ hỏi bạn mật khẩu của bạn.Nhập mật khẩu của bạn và nhấn enter.Python phiên bản ba sẽ bắt đầu cài đặt.Sau khi cài đặt hoàn tất,bạn có thể bắt đầu viết mã Python từ thiết bị đầu cuối.Chỉ cần gõ python3 để sử dụng chế độ tương tác.Hãy thử chế độ tương tác bằng câu lệnh in.Gõ print Hello World và nhấn enter.Bạn sẽ thấy kết quả là "Xin chào thế giới".Để thoát khỏi chế độ tương tác, chúng ta sẽ gõ exit().

|  |
| --- |
| You can also write Python scripts using Linux with Idle. Idle allows you to write Python scripts with ease on your local machine. Idle is an integrated development and learning environment for the Python programming language. Python Idle is included with Python installations on Windows and Mac OS. You can download Idle using your package manager on Linux. Python Idle is an interactive interpreter, or file editor, that allows you to easily write Python scripts and programs. Idle provides syntax highlighting, code completion, and automatic indentation. To install Idle on Linux, type sudo apt install idle. Once installation is complete, type idle to open the interpreter and begin coding. |

Bạn cũng có thể viết tập lệnh Python bằng Linux với Idle.Nhàn rỗi cho phép bạn viết các tập lệnh Pythonmột cách dễ dàng trên máy cục bộ của bạn.Idle là một môi trường học tập và phát triển tích hợpcho ngôn ngữ lập trình Python.Python Idle được bao gồm trong bản cài đặt Pythontrên Windows và Mac OS.Bạn có thể tải xuống chế độ Chờ bằng trình quản lý gói trên Linux.Python Idle là một trình thông dịch tương tác,hoặc trình soạn thảo tập tin, điều đó cho phép bạnđể dễ dàng viết các tập lệnh và chương trình Python.Nhàn rỗi cung cấp tính năng tô sáng cú pháp, hoàn thành mã,và thụt lề tự động.Để cài đặt Idle trên Linux, hãy nhập sudo apt install Idle.Sau khi cài đặt hoàn tất,gõ nhàn rỗi để mở trình thông dịch và bắt đầu viết mã.

|  |
| --- |
| Now we can write a Python file! |

Bây giờ chúng ta có thể viết một tệp Python!

|  |
| --- |
| Once you've completed writing your Python code in the file, go to file save as. Give your Python file a name. In this example, we will name our file sum.py. Hit save. To run the Python code in the file you saved, click run, run module from the menu, or press enter. |

Sau khi bạn viết xong mã Python vào tệp,đi đến tập tin lưu dưới dạng.Đặt tên cho tệp Python của bạn.Trong ví dụ này, chúng tôi sẽ đặt tên tệp là sum.py. Nhấn lưu.Để chạy mã Python trong tệp bạn đã lưu, hãy nhấp vào chạy,chạy mô-đun từ menu hoặc nhấn enter.

|  |
| --- |
| Now you know how to use the command line to begin writing Python code. Remember, whichever operating system you're using, you'll be able to run Python from the command line. Using a text editor like Idle and running Python from the command line is best for executing and debugging individual scripts or .py files. Next, let's explore some other code editors and IDEs. I'll see you over in the next video. |

Bây giờ bạn đã biết cách sử dụng dòng lệnhđể bắt đầu viết mã Python.Hãy nhớ rằng, cho dù bạn đang sử dụng hệ điều hành nào,bạn sẽ có thể chạy Python từ dòng lệnh.Sử dụng trình soạn thảo văn bản như Idle và chạy Pythontừ dòng lệnh là tốt nhất để thực thivà gỡ lỗi các tập lệnh hoặc tệp .py riêng lẻ.Tiếp theo, hãy khám phá một số trình soạn thảo mã và IDE khác.Tôi sẽ gặp lại bạn trong video tiếp theo.

***4.12. Review: Use JupyterLab and Jupyter Notebooks – Đánh giá: Sử dụng Notebook JupyterLab và Jupyter***

|  |
| --- |
| **Review: Use JupyterLab and Jupyter Notebooks** |

**Đánh giá: Sử dụng Notebook JupyterLab và Jupyter**

|  |
| --- |
| There are many different places and environments that you can use to write and run code.  In this video, you will learn more about JupyterLab and Jupyter Notebooks to write, run, and debug Python code. JupyterLab and Jupyter Notebooks are part of an open source project called [Project Jupyter](https://jupyter.org/) and are free to use. |

Có nhiều nơi và môi trường khác nhau mà bạn có thể sử dụng để viết và chạy mã. Trong video này, bạn sẽ tìm hiểu thêm về JupyterLab và Jupyter Notebooks để viết, chạy và gỡ lỗi mã Python. JupyterLab và Jupyter Notebooks là một phần của dự án nguồn mở có tên[Dự án Jupyter](https://jupyter.org/)và được sử dụng miễn phí.

|  |
| --- |
| **JupyterLab** |

**JupyterLab**

|  |
| --- |
| JupyterLab is a web-based interface that allows you to use Jupyter Notebooks to write, run, and debug Python code. JupyterLab is an online environment that allows you to run your code in the cloud. |

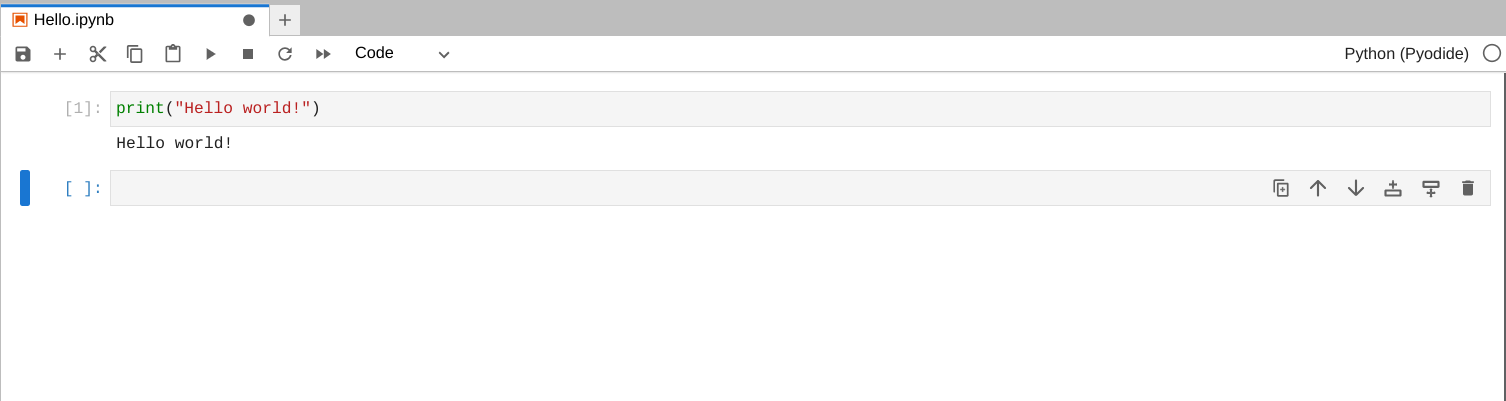
JupyterLab là giao diện dựa trên web cho phép bạn sử dụng Notebook Jupyter để viết, chạy và gỡ lỗi mã Python. JupyterLab là môi trường trực tuyến cho phép bạn chạy mã của mình trên đám mây.

|  |
| --- |
| **Jupyter Notebook** |

**Jupyter Notebook**

|  |
| --- |
| Jupyter Notebook can be used and run in the web-based interface through [JupyterLab](https://jupyter.org/try-jupyter/lab/) or on your local machine. Jupyter Notebook allows you to create documents that contain live code. You can write Python programs and scripts using Jupyter Notebooks and see how they execute all in one place. It is a great tool for creating and understanding the code you are writing because you can see your input and output all in one spot. |

Jupyter Notebook có thể được sử dụng và chạy trong giao diện dựa trên web thông qua[JupyterLab](https://jupyter.org/try-jupyter/lab/)hoặc trên máy cục bộ của bạn. Jupyter Notebook cho phép bạn tạo tài liệu chứa mã trực tiếp. Bạn có thể viết các chương trình và tập lệnh Python bằng Jupyter Notebooks và xem cách chúng thực thi tất cả ở một nơi. Đây là một công cụ tuyệt vời để tạo và hiểu mã bạn đang viết vì bạn có thể thấy tất cả đầu vào và đầu ra của mình ở cùng một nơi.



|  |
| --- |
| **Using Jupyter Notebook on JupyterLab** |

**Sử dụng Notebook Jupyter trên JupyterLab**

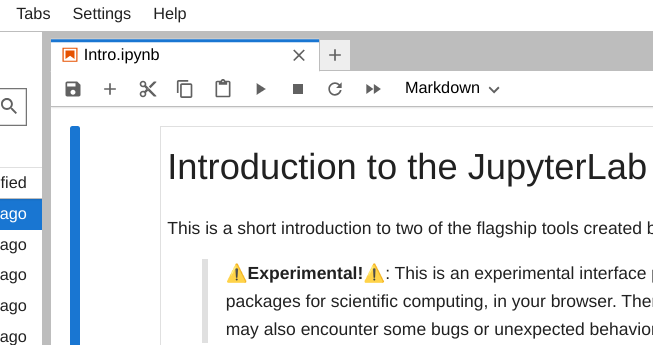
|  |
| --- |
| To use Jupyter Notebook in the cloud-based environment click [here](https://jupyter.org/try-jupyter/lab/?path=notebooks%2FIntro.ipynb). To create a new notebook in the lab environment, + a new tab, and select Python under |

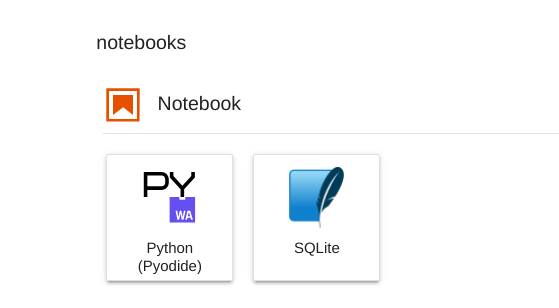
Để sử dụng Jupyter Notebook trong môi trường dựa trên đám mây, hãy nhấp vào[đây](https://jupyter.org/try-jupyter/lab/?path=notebooks%2FIntro.ipynb). Để tạo sổ ghi chép mới trong môi trường phòng thí nghiệm, + tab mới và chọn Python bên dưới



|  |
| --- |
| Notebook. A new Notebook will open up, and you can start writing and running Python code. |

Sổ tay. Một Notebook mới sẽ mở ra và bạn có thể bắt đầu viết và chạy mã Python.





|  |
| --- |
| **Installing Jupyter** |

**Cài đặt Jupyter**

|  |
| --- |
| If you would like to install JupyterLab and Jupyter Notebooks on your local machine you can do this by using the pip command from the Python terminal command-line. |

Nếu bạn muốn cài đặt JupyterLab và Jupyter Notebook trên máy cục bộ của mình, bạn có thể thực hiện việc này bằng cách sử dụng lệnh pip từ dòng lệnh đầu cuối Python.

|  |
| --- |
| **JupyterLab** |

**JupyterLab**

|  |
| --- |
| 1. Install JupyterLab **pip install jupyterlab** 2. Once installed, launch JupyterLab **jupyter-lab** |

1. Cài đặt JupyterLab **pip install jupyterlab**
2. Sau khi cài đặt, hãy khởi chạy JupyterLab **jupyter-lab**

|  |
| --- |
| **Jupyter Notebook** |

**Jupyter Notebook**

|  |
| --- |
| 1. Install the classic Jupyter Notebook **pip install notebook** 2. Run the notebook **jupyter notebook** |

1. Cài đặt Jupyter Notebook cổ điển **pip install notebook**
2. Chạy sổ ghi chép **jupyter notebook**

|  |
| --- |
| **Sharing Jupyter Notebooks** |

**Chia sẻ sổ tay Jupyter**

|  |
| --- |
| JupyterLab and Jupyter Notebooks is a great resource for writing Python code. Notebooks can be shared and saved easily using email, GitHub, and [Jupyter Notebook Viewer](https://nbviewer.org/). |

JupyterLab và Jupyter Notebooks là nguồn tài nguyên tuyệt vời để viết mã Python. Sổ ghi chép có thể được chia sẻ và lưu dễ dàng bằng email, GitHub và[Trình xem sổ tay Jupyter](https://nbviewer.org/).

|  |
| --- |
| **Jupyter Resources** |

**Tài nguyên Jupyter**

|  |
| --- |
| There are tons of Jupyter resources available. Here is a list of some great ones to help you get started! |

Có rất nhiều tài nguyên Jupyter có sẵn. Dưới đây là danh sách một số điều tuyệt vời để giúp bạn bắt đầu!

|  |
| --- |
| * Try [Jupyter Notebooks](https://docs.jupyter.org/en/latest/start/index.html) in your browser. * Install and use [JupyterLab and Jupyter Notebooks](https://docs.jupyter.org/en/latest/install.html). * More information on [getting started](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html) and using Jupyter Notebooks. * [Official documentation](https://docs.jupyter.org/en/latest/) for Project Jupyter. |

* Thử[Máy tính xách tay Jupyter](https://docs.jupyter.org/en/latest/start/index.html)trong trình duyệt của bạn.
* Cài đặt và sử dụng[Sổ tay JupyterLab và Jupyter](https://docs.jupyter.org/en/latest/install.html).
* Thông tin thêm về[bắt đầu](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html)và sử dụng Notebook Jupyter.
* [Tài liệu chính thức](https://docs.jupyter.org/en/latest/)cho Dự án Jupyter.

***4.13. Use JupyterLab and Jupyter Notebooks – Sử dụng sổ ghi chép JupyterLab và Jupyter***

|  |
| --- |
| - There are many different environments that you can use to write and run code. In this video, you'll learn more about Jupyter Lab and Jupyter Notebooks for writing, running and debugging Python code. Jupyter Lab and Jupyter Notebooks are part of an open source project called Project Jupyter and are free to use. You can learn more about Project Jupyter at jupyter.org. Jupyter Lab is a web-based interface that allows you to use Jupyter Notebooks to write, run and debug Python code. Jupyter Lab provides an online environment that allows you to run your code in the cloud. Jupyter Notebook can be used in the web-based interface through Jupyter Lab or on your local machine. Jupyter Notebook allows you to create text-based documents that contain live code blocks. You can write Python programs and scripts using Jupyter Notebooks and see how they execute all in one place. It's a great tool for creating and understanding the code you're writing because you can see your input and output all in one spot. This is what a basic Jupyter Notebook looks like. Next, we will cover how to use Jupyter Notebooks. One of the most popular ways to use Jupyter Notebooks is in the cloud-based environment. You can use this environment by visiting jupyter.org. Once you're at jupyter.org, you can get started. First, click try. Then select Jupyter Lab. To create a new notebook in the lab environment, add a new tab and select Python under notebook. A new notebook will open up and you can start writing and running Python code. Let's give it a try. You will write your code in the cell with the blue boarder. Let's use a print statement. Type print "Hello world" in the cell. Next, we will run the code in the cell. To do this, we'll use the run button located at the top of the toolbar. Click on run and your cell will execute. Your output should be Hello World. |

- Có nhiều môi trường khác nhaumà bạn có thể sử dụng để viết và chạy mã.Trong video này, bạn sẽ tìm hiểu thêm về Jupyter Labvà Notebook Jupyter để viết, chạyvà gỡ lỗi mã Python.Phòng thí nghiệm Jupyter và sổ ghi chép Jupyterlà một phần của dự án nguồn mở có tên Project Jupytervà được sử dụng miễn phí.Bạn có thể tìm hiểu thêm về Project Jupyter tại jupyter.org.Jupyter Lab là một giao diện dựa trên webcho phép bạn sử dụng Notebook Jupyter để viết, chạyvà gỡ lỗi mã Python.Jupyter Lab cung cấp một môi trường trực tuyếncho phép bạn chạy mã của mình trên đám mây.Jupyter Notebook có thể được sử dụng trong giao diện dựa trên webthông qua Jupyter Lab hoặc trên máy cục bộ của bạn.Jupyter Notebook cho phép bạn tạo tài liệu dựa trên văn bảncó chứa các khối mã trực tiếp.Bạn có thể viết các chương trình và tập lệnh Pythonsử dụng sổ ghi chép Jupytervà xem cách họ thực thi tất cả ở một nơi.Đó là một công cụ tuyệt vời để tạo và hiểuđoạn mã bạn đang viếtbởi vì bạn có thể thấy tất cả đầu vào và đầu ra của mình ở cùng một nơi.Đây là hình dáng cơ bản của một Notebook Jupyter.Tiếp theo, chúng tôi sẽ đề cập đến cách sử dụng Notebook Jupyter.Một trong những cách phổ biến nhất để sử dụng Notebook Jupyterlà trong môi trường dựa trên đám mây.Bạn có thể sử dụng môi trường này bằng cách truy cập jupyter.org.Khi đã truy cập jupyter.org, bạn có thể bắt đầu.Đầu tiên hãy bấm thử.Sau đó chọn Phòng thí nghiệm Jupyter.Để tạo một sổ ghi chép mới trong môi trường phòng thí nghiệm,thêm một tab mới và chọn Python trong sổ ghi chép.Một cuốn sổ mới sẽ mở ravà bạn có thể bắt đầu viết và chạy mã Python.Thử một lần đi.Bạn sẽ viết mã của mình vào ô có bảng nội trú màu xanh lam.Hãy sử dụng một câu lệnh in.Gõ chữ "Xin chào thế giới" vào ô.Tiếp theo, chúng ta sẽ chạy mã trong ô.Để làm điều này, chúng tôi sẽ sử dụng nút chạynằm ở phía trên cùng của thanh công cụ.Bấm vào chạy và ô của bạn sẽ thực thi.Đầu ra của bạn phải là Hello World.

|  |
| --- |
| You can save your Jupyter Notebook by using the save button that's also located in the top toolbar. Click save and give your notebook file a name. For this example, I will name my file Hello.ipynb. Also, if you would like to download your notebook, you can do that using the file menu located in the top left-hand corner. |

Bạn có thể lưu Notebook Jupyter của mìnhbằng cách sử dụng nút lưu cũng nằm ở đóở thanh công cụ trên cùng.Nhấp vào lưu và đặt tên cho tệp sổ ghi chép của bạn.Trong ví dụ này, tôi sẽ đặt tên tệp của mình là Hello.ipynb.Ngoài ra, nếu bạn muốn tải sổ ghi chép của mình xuống,bạn có thể làm điều đó bằng cách sử dụng menu tập tinnằm ở góc trên bên trái.

|  |
| --- |
| You can also use Jupyter Lab and Jupyter Notebooks on your local machine. You can do this by using the pip command from the Python terminal command line. To install Jupyter Lab, type pip install Jupyter Lab at the command line. Once it's installed, you can launch it by typing Jupyter-Lab. To install Jupyter Notebooks, type pip install notebook at the command line. Once it's installed, you can launch it by typing Jupyter Notebook. Jupyter Lab and Jupyter Notebooks are great resources for writing Python code. To note, notebooks can also be shared and saved easily using email, GitHub and Jupyter Notebook Viewer. There are also tons of resources on Jupyter Notebooks out there, so if you need more support the Jupyter website is a great resource to dive deeper into. In the next video, we will go over CoLab. This is a coding tool that is very similar to Jupyter Notebooks. Head over to the next video to learn more. |

Bạn cũng có thể sử dụng Jupyter Lab và Jupyter Notebookstrên máy cục bộ của bạn.Bạn có thể làm điều này bằng cách sử dụng lệnh piptừ dòng lệnh đầu cuối Python.Để cài đặt Jupyter Lab, gõ pip install Jupyter Labtại dòng lệnh.Sau khi cài đặt xong, bạn có thể khởi chạy nóbằng cách gõ Jupyter-Lab.Để cài đặt Notebook Jupyter, hãy nhập pip install notebooktại dòng lệnh.Một khi nó được cài đặt,bạn có thể khởi chạy nó bằng cách gõ Jupyter Notebook.Jupyter Lab và Jupyter Notebooks là những tài nguyên tuyệt vờiđể viết mã Python.Cần lưu ý, sổ ghi chép cũng có thể được chia sẻ và lưu lại dễ dàngsử dụng email, GitHub và Jupyter Notebook Viewer.Ngoài ra còn có rất nhiều tài nguyêntrên Jupyter Notebooks hiện có, vì vậy nếu bạn cần hỗ trợ thêmtrang web Jupyter là một nguồn tài nguyên tuyệt vời để tìm hiểu sâu hơn.Trong video tiếp theo, chúng ta sẽ tìm hiểu về CoLab.Đây là một công cụ mã hóa rất giốngtới Máy tính xách tay Jupyter.Hãy chuyển sang video tiếp theo để tìm hiểu thêm.

***4.14. Review: Use Colab – Đánh giá: Sử dụng Colab***

|  |
| --- |
| **Review: Use Colab** |

**Đánh giá: Sử dụng Colab**

|  |
| --- |
| Colab Notebooks are Jupyter Notebooks that are hosted by Google’s Colaboratory. In Colab, you can write and run Python code. In this reading, you will learn more about how to use Colab and its features. |

Sổ tay Colab là Sổ tay Jupyter được lưu trữ bởi Colaboratory của Google. Trong Colab, bạn có thể viết và chạy mã Python. Trong bài đọc này, bạn sẽ tìm hiểu thêm về cách sử dụng Colab và các tính năng của Colab.

|  |
| --- |
| **About Colab** |

**Giới thiệu về Colab**

|  |
| --- |
| Colab is a web-based platform that allows you to write and run Python code really quickly in Google Drive. It is free and  ready to use with zero configuration required. You can simply visit [Colab](https://colab.sandbox.google.com/notebooks/basic_features_overview.ipynb) to get started! |

Colab là một nền tảng dựa trên web cho phép bạn viết và chạy mã Python rất nhanh chóng trong Google Drive. Nó miễn phí và sẵn sàng để sử dụng mà không cần cấu hình. Bạn có thể chỉ cần truy cập

|  |
| --- |
| **Colab’s Features** |

**Tính năng của Colab**

|  |
| --- |
| Colab provides all of the functionality that Python offers. The cells within Colab can include code, text, and images. The code cells include executable code and rich text which make it easy to write and run code. Colab also makes it easy to include markdown in your notebooks. This is a great feature for sharing notebooks because you can add headings, paragraphs, lists, mathematical formulas, and more. You can install Python packages using the pip command within the code cell. Colab Notebooks can easily be shared with other collaborators, too. |

Colab cung cấp tất cả chức năng mà Python cung cấp. Các ô trong Colab có thể bao gồm mã, văn bản và hình ảnh. Các ô mã bao gồm mã thực thi và văn bản đa dạng thức giúp bạn dễ dàng viết và chạy mã. Colab cũng giúp bạn dễ dàng đưa phần đánh dấu vào sổ tay của mình. Đây là một tính năng tuyệt vời để chia sẻ sổ ghi chép vì bạn có thể thêm tiêu đề, đoạn văn, danh sách, công thức toán học, v.v. Bạn có thể cài đặt các gói Python bằng lệnh pip trong ô mã. Colab Notebook cũng có thể dễ dàng được chia sẻ với các cộng tác viên khác.

|  |
| --- |
| Note: When you create a Colab Notebook it is stored in your Google Drive. You can easily share using the share button in the upper right hand corner of the notebook. |

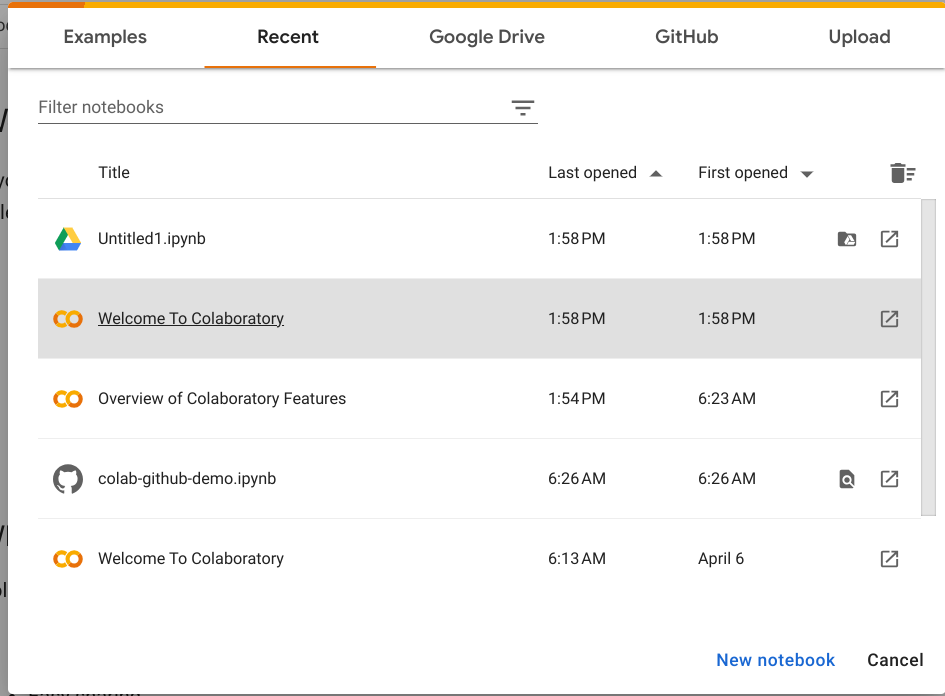
Lưu ý: Khi bạn tạo Sổ tay Colab, sổ tay này sẽ được lưu trữ trong Google Drive của bạn. Bạn có thể dễ dàng chia sẻ bằng nút chia sẻ ở góc trên bên phải của sổ ghi chép.

|  |
| --- |
| **Create a Colab Notebook** |

**Tạo sổ tay Colab**

|  |
| --- |
| 1. Open [Google Colab](https://colab.sandbox.google.com/notebooks/welcome.ipynb#recent=true). 2. Click on New Notebook in the bottom right corner. |

1. Mở[Google Colab](https://colab.sandbox.google.com/notebooks/welcome.ipynb#recent=true).
2. Bấm vào Sổ tay mới ở góc dưới bên phải.



|  |
| --- |
| 1. Begin writing your Python code. 2. To run your code and see its execution use the |

1. Bắt đầu viết mã Python của bạn.
2. Để chạy mã của bạn và xem việc thực thi nó, hãy sử dụng



|  |
| --- |
| **Pro tip:** To install a Python library use the pip command with an exclamation point.  **!pip install <library name>** |

**Mẹo chuyên nghiệp:** Để cài đặt thư viện Python, hãy sử dụng lệnh pip kèm theo dấu chấm than.

**!pip install <library name>**

|  |
| --- |
| **Resources for more information** |

**Tài nguyên để biết thêm thông tin**

|  |
| --- |
| Review the following resources to find out more about Colab:   * This [resource](https://colab.sandbox.google.com/notebooks/basic_features_overview.ipynb) provides an overview of Colab’s features with examples. * This is a [guide](https://colab.sandbox.google.com/notebooks/snippets/importing_libraries.ipynb) to installing and using Python libraries in Colab. * This [resource](https://colab.sandbox.google.com/github/googlecolab/colabtools/blob/main/notebooks/colab-github-demo.ipynb) provides directions on how to save and load Colab Notebooks in GitHub. |

Hãy xem lại các tài nguyên sau để tìm hiểu thêm về Colab:

* Cái này[nguồn](https://colab.sandbox.google.com/notebooks/basic_features_overview.ipynb)cung cấp thông tin tổng quan về các tính năng của Colab kèm theo ví dụ.
* Đây là một[hướng dẫn](https://colab.sandbox.google.com/notebooks/snippets/importing_libraries.ipynb)cách cài đặt và sử dụng thư viện Python trong Colab.
* Cái này[nguồn](https://colab.sandbox.google.com/github/googlecolab/colabtools/blob/main/notebooks/colab-github-demo.ipynb)cung cấp hướng dẫn về cách lưu và tải Sổ tay Colab trong GitHub.

***4.15. Use Colab – Sử dụng Colab***

|  |
| --- |
| - Recently, we covered how to use Jupyter Notebooks. In this video, we will go over Colab notebooks. Colab Notebooks are Jupyter Notebooks that are hosted by Google's Colaboratory. In Colab, you can write and run Python code. In this video, you'll learn more about how to use Colab and its features. Colab is a web-based platform that allows you to write and run Python code really quickly in Google Drive. It's free and ready to use with zero configuration required. You can simply visit colab.sandbox.google.com or search Google Colab to get started. |

- Gần đây, chúng tôi đã đề cập đến cách sử dụng Notebook Jupyter.Trong video này, chúng ta sẽ tìm hiểu về sổ tay Colab.Sổ tay Colab là Sổ tay Jupyter được lưu trữbởi Colaboratory của Google.Trong Colab, bạn có thể viết và chạy mã Python.Trong video này, bạn sẽ tìm hiểu thêmvề cách sử dụng Colab và các tính năng của Colab.Colab là một nền tảng dựa trên web cho phép bạn viếtvà chạy mã Python thực sự nhanh chóng trong Google Drive.Nó miễn phí và sẵn sàng để sử dụng mà không cần cấu hình.Bạn chỉ cần truy cập colab.sandbox.google.comhoặc tìm kiếm Google Colab để bắt đầu.

|  |
| --- |
| Colab provides all of the functionality that Python offers. The cells within Colab can include code, text, and images. The code cells include executable code and rich text, which makes it easy to write and run code. Colab also makes it easy to include markdown in your notebooks. This is a great feature for sharing notebooks because you can add headings, paragraphs, lists, mathematical formulas and more. You can install Python packages using the PIP command within the code cell. Colab Notebooks can easily be shared with other Colaborators too. Now, let's go over how to use Colab. First, open Google Colab. Again, you can do this by visiting colab.sandbox.google.com. Go to file and click on New notebook. Now you can begin writing your Python code. In the code cell, let's write a print statement that outputs your name. |

Colab cung cấp tất cả chức năng mà Python cung cấp.Các ô trong Colab có thể bao gồm mã, văn bản và hình ảnh.Các ô mã bao gồm mã thực thi và văn bản có định dạng,giúp việc viết và chạy mã trở nên dễ dàng.Colab cũng giúp bạn dễ dàng thêm phần giảm giátrong sổ tay của bạn.Đây là một tính năng tuyệt vời để chia sẻ sổ ghi chépbởi vì bạn có thể thêm tiêu đề, đoạn văn, danh sách,công thức toán học và nhiều hơn nữa.Bạn có thể cài đặt các gói Python bằng lệnh PIPtrong ô mã.Sổ tay Colab có thể được chia sẻ dễ dàngvới các Cộng tác viên khác nữa.Bây giờ, hãy cùng tìm hiểu cách sử dụng Colab.Đầu tiên, hãy mở Google Colab.Một lần nữa, bạn có thể thực hiện việc này bằng cách truy cập colab.sandbox.google.com.Đi tới tập tin và nhấp vào Sổ ghi chép mới.Bây giờ bạn có thể bắt đầu viết mã Python của mình.Trong ô mã,hãy viết một câu lệnh in xuất ra tên của bạn.

|  |
| --- |
| To run your code and see its execution, use the run button to the left of the code cell. You can also give your coding project a name. At the top, you can change Untitled.ipynb to something like Name.ipynb. This will make it easier to find your code in the future. |

Để chạy mã của bạn và xem việc thực thi nó,sử dụng nút chạy ở bên trái của ô mã.Bạn cũng có thể đặt tên cho dự án mã hóa của mình.Ở trên cùng,bạn có thể thay đổi Untitled.ipynbđến một cái gì đó như Name.ipynb.Điều này sẽ giúp bạn tìm thấy mã của mình dễ dàng hơn trong tương lai.

|  |
| --- |
| The great thing about Colab is that it automatically stores your notebooks in Google Drive. To find your notebooks, you can simply visit your Google Drive, type in Colab Notebooks in the search bar and open up the folder titled Colab Notebooks. Here you will see all of the notebooks you've created. And that's how you use Colab. Pretty simple, right? I'm sure by now you're getting the hang of using these code editors. Let's move on to our final one, VS code. I'll see you in the next video. |

Điều tuyệt vời về Colablà nó tự động lưu trữ sổ ghi chép của bạntrong Google Drive.Để tìm sổ ghi chép của bạn,bạn chỉ cần truy cập Google Drive của mình,nhập Colab Notebooks vào thanh tìm kiếmrồi mở thư mục có tiêu đề Colab Notebooks.Ở đây bạn sẽ thấy tất cả các sổ ghi chép bạn đã tạo.Và đó là cách bạn sử dụng Colab.Khá đơn giản phải không?Tôi chắc chắn bây giờ bạn đã hiểu rõsử dụng các trình soạn thảo mã này.Hãy chuyển sang phần cuối cùng của chúng ta, mã VS.Tôi sẽ gặp bạn trong video tiếp theo.

***4.16. Review: Use VS Code – Đánh giá: Sử dụng mã VS***

|  |
| --- |
| **Review: Use VS Code** |

**Đánh giá: Sử dụng mã VS**

|  |
| --- |
| VS code is an open source code editor that includes developer tooling. It is similar to Jupyter Notebooks and Colab, but it includes more features. VS code provides built-in support with [Intellisense](https://code.visualstudio.com/docs/editor/intellisense#:~:text=IntelliSense%20is%20a%20general%20term,%2C%20and%20%22code%20hinting.%22) code completion, an interactive debugger, and other build and scripting tools. VS code has a simple design that is easy to use. Its intuitive features make it a great choice for coding in Python! |

Mã VS là trình soạn thảo mã nguồn mở bao gồm công cụ dành cho nhà phát triển. Nó tương tự như Jupyter Notebooks và Colab, nhưng có nhiều tính năng hơn. Mã VS cung cấp hỗ trợ tích hợp với[trí tuệ](https://code.visualstudio.com/docs/editor/intellisense#:~:text=IntelliSense%20is%20a%20general%20term,%2C%20and%20%22code%20hinting.%22)hoàn thành mã, trình gỡ lỗi tương tác cũng như các công cụ xây dựng và tạo tập lệnh khác. Mã VS có thiết kế đơn giản, dễ sử dụng. Các tính năng trực quan của nó làm cho nó trở thành một lựa chọn tuyệt vời để viết mã bằng Python!

|  |
| --- |
| **Using VS Code with Python** |

**Sử dụng mã VS với Python**

|  |
| --- |
| To use VS code for Python, you will need to have Python3, VS Code, and the VS Code Python extension installed. Make sure that you have Python3 installed on your computer by typing the following command in the terminal on your computer. |

Để sử dụng mã VS cho Python, bạn cần phải cài đặt tiện ích mở rộng Python3, VS Code và VS Code Python. Đảm bảo rằng bạn đã cài đặt Python3 trên máy tính của mình bằng cách nhập lệnh sau vào terminal trên máy tính.

|  |
| --- |
| Linux/macOS: open a Terminal Window and type the following command:  **python3 --version** |

Linux/macOS: mở Cửa sổ đầu cuối và gõ lệnh sau:

**python3 --version**

|  |
| --- |
| Windows: open a command prompt and run the following command:  **py -3 --version** |

Windows: mở dấu nhắc lệnh và chạy lệnh sau:

**py -3 --version**

|  |
| --- |
| If Python3 is installed the output should look like this:  **$python3 -- version**  **Python 3.11.3** |

Nếu Python3 được cài đặt, kết quả sẽ trông như thế này:

**$python3 -- version**

**Python 3.11.3**

|  |
| --- |
| If it is not installed the output will look like this:  **$python3 -- version**  **command not found: python** |

Nếu nó chưa được cài đặt, đầu ra sẽ trông như thế này:

**$python3 -- version**

**command not found: python**

|  |
| --- |
| **Download VS Code and install the Python extension** |

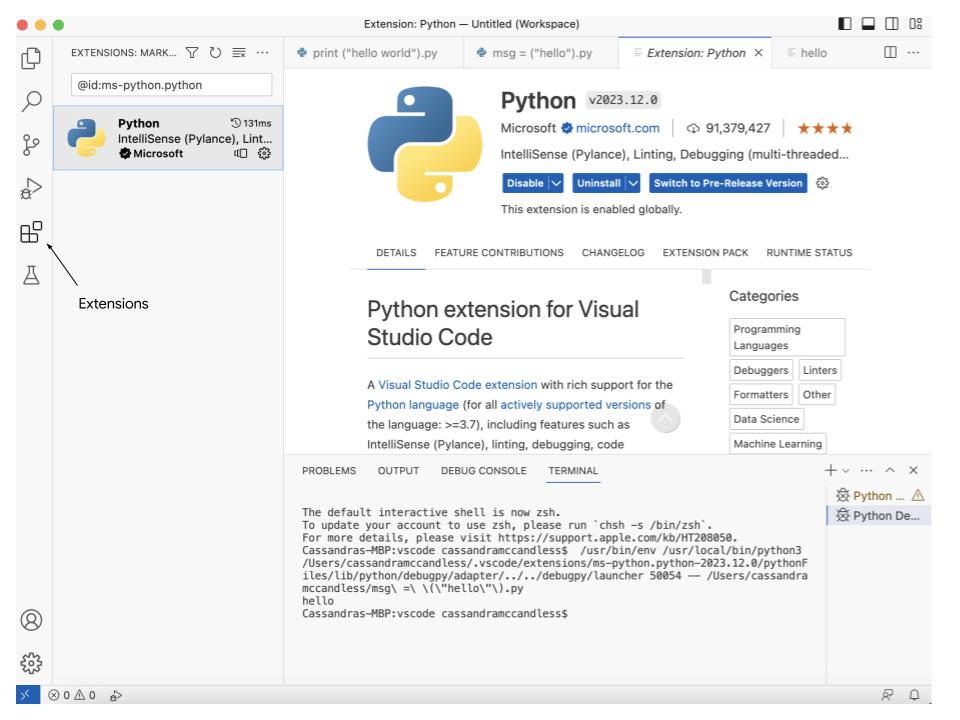
**Tải xuống Mã VS và cài đặt tiện ích mở rộng Python**

|  |
| --- |
| You can download VS code [here](https://code.visualstudio.com/Download). You can use VS code on Windows, Mac, and Linux operating systems. Download the version that is compatible with your operating system. Follow the download prompts. Once the download and installation is complete you will be able to open VS Code and begin using it. |

Bạn có thể tải xuống mã VS[đây](https://code.visualstudio.com/Download). Bạn có thể sử dụng mã VS trên hệ điều hành Windows, Mac và Linux. Tải xuống phiên bản tương thích với hệ điều hành của bạn. Làm theo lời nhắc tải xuống. Sau khi quá trình tải xuống và cài đặt hoàn tất, bạn sẽ có thể mở VS Code và bắt đầu sử dụng nó.

|  |
| --- |
| Next, you will install the Python extension. You can do this by visiting the [VS Code Marketplace](https://marketplace.visualstudio.com/items?itemName=ms-python.python). Follow the download instructions there.  You can check to make sure the Python extension was added successfully by clicking on this icon in VS Code. |

Tiếp theo, bạn sẽ cài đặt tiện ích mở rộng Python. Bạn có thể làm điều này bằng cách truy cập[Thị trường mã VS](https://marketplace.visualstudio.com/items?itemName=ms-python.python). Làm theo hướng dẫn tải xuống ở đó. Bạn có thể kiểm tra để đảm bảo tiện ích mở rộng Python đã được thêm thành công bằng cách nhấp vào biểu tượng này trong Mã VS.



|  |
| --- |
| **Note:** You cannot use  VS Code Marketplace to install the Python extension on MacOS. |

**Lưu ý:** Bạn không thể sử dụng VS Code Marketplace để cài đặt tiện ích mở rộng Python trên MacOS.

|  |
| --- |
| For MacOS, open the command palette in VS code. You can do this by hitting Cmd+Shift+P. Type ‘**shell command**’. Find the Shell Command: **Install ‘code’ command in PATH**. |

Đối với MacOS, hãy mở bảng lệnh trong mã VS. Bạn có thể thực hiện việc này bằng cách nhấn Cmd+Shift+P. Gõ ' **lệnh shell** '. Tìm lệnh Shell: **Cài đặt lệnh 'code' trong PATH** .

|  |
| --- |
| Once this is completed you can begin writing, running, and debugging Python code in VS Code! |

Sau khi hoàn tất, bạn có thể bắt đầu viết, chạy và gỡ lỗi mã Python trong Mã VS!

|  |
| --- |
| **Pro tip:** To try VS Code without downloading or installation click [here](https://vscode.dev/). |

**Mẹo chuyên nghiệp:** Để dùng thử VS Code mà không cần tải xuống hoặc cài đặt, hãy nhấp vào[đây](https://vscode.dev/).

|  |
| --- |
| **Create a Python File** |

**Tạo một tệp Python**

|  |
| --- |
| To create a Python file in VS Code go to File → New File… → Python File. A new workspace will appear, and you can begin writing your Python code. Let’s test it out with a simple statement. Type the following statement into the new workspace. |

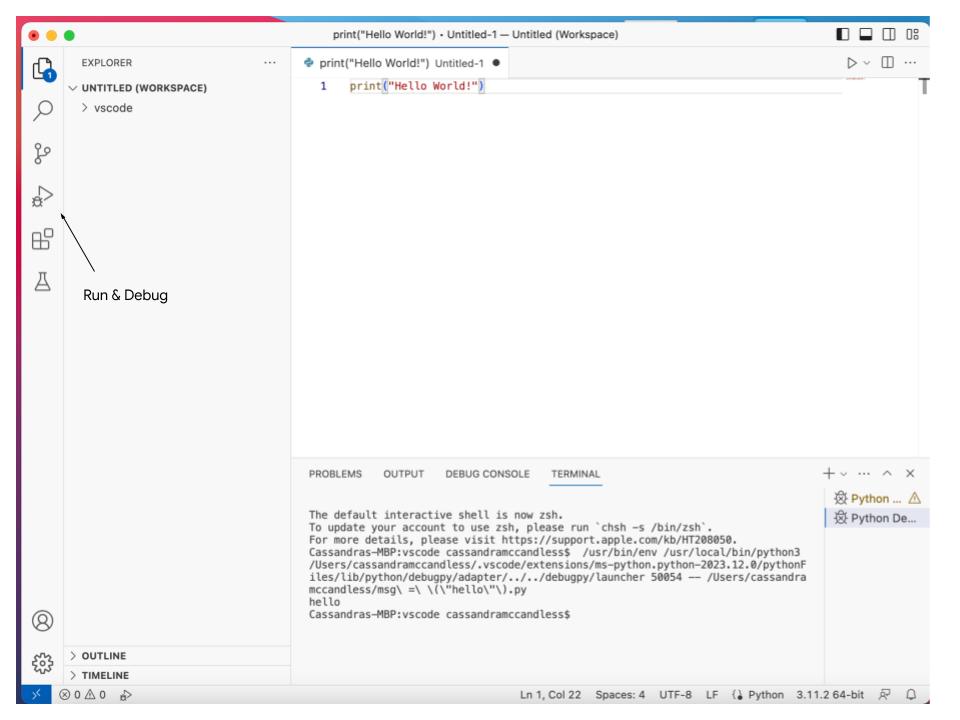
Để tạo tệp Python trong Mã VS, hãy đi tới Tệp → Tệp mới… → Tệp Python. Một không gian làm việc mới sẽ xuất hiện và bạn có thể bắt đầu viết mã Python của mình. Hãy kiểm tra nó bằng một tuyên bố đơn giản. Nhập câu lệnh sau vào không gian làm việc mới.

|  |
| --- |
| **print(“Hello World!”)** |

**print(“Hello World!”)**

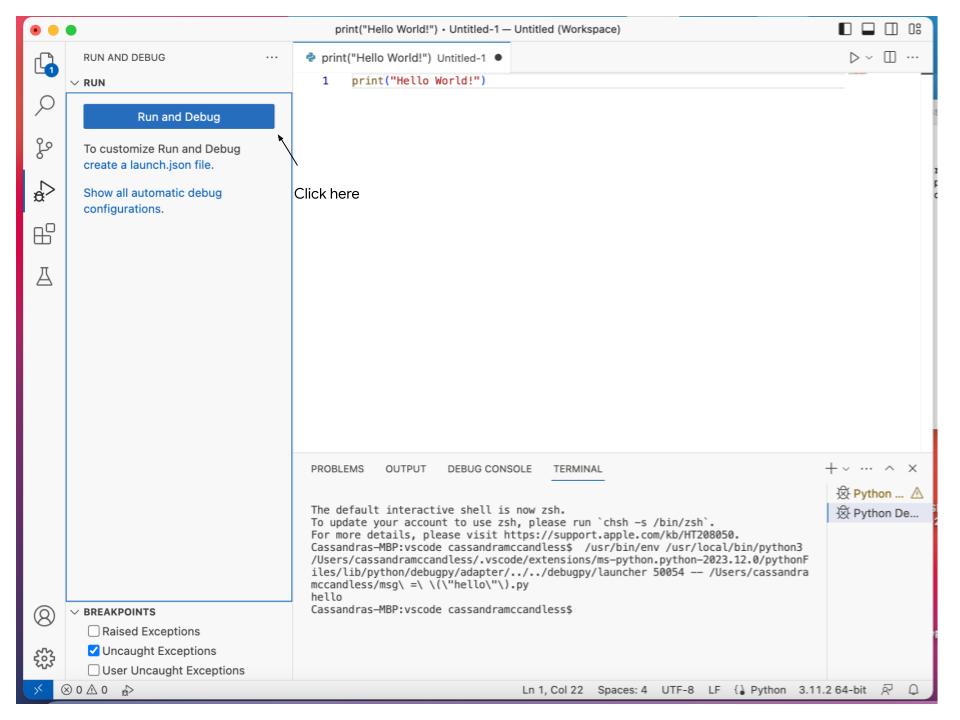
|  |
| --- |
| Next, click on  the Run and Debug icon in the left hand toolbar. |

Tiếp theo, nhấp vào biểu tượng Run and Debug trên thanh công cụ bên trái.



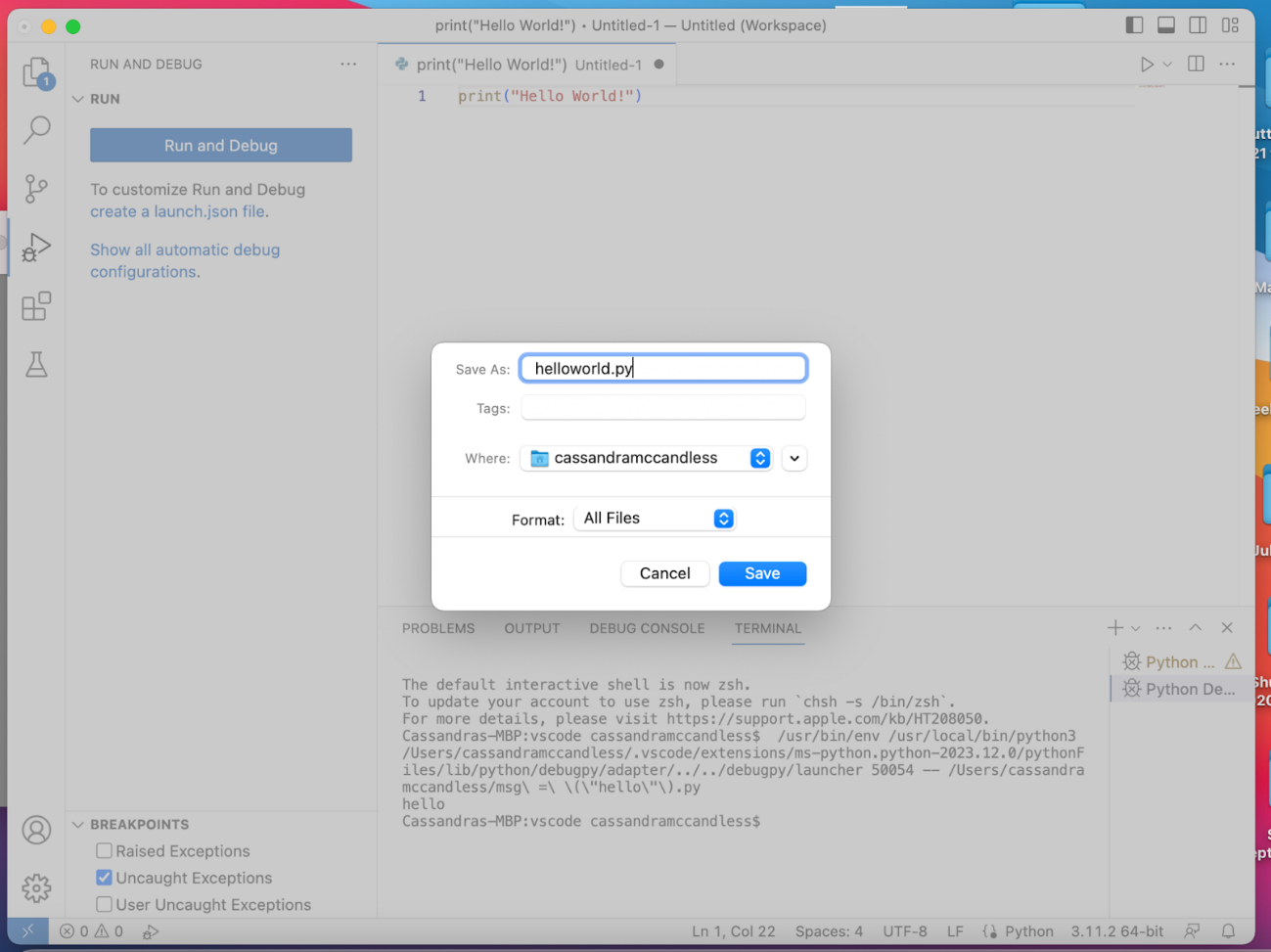
|  |
| --- |
| Then, click Run and Debug here: |

Sau đó, nhấp vào Chạy và gỡ lỗi tại đây:



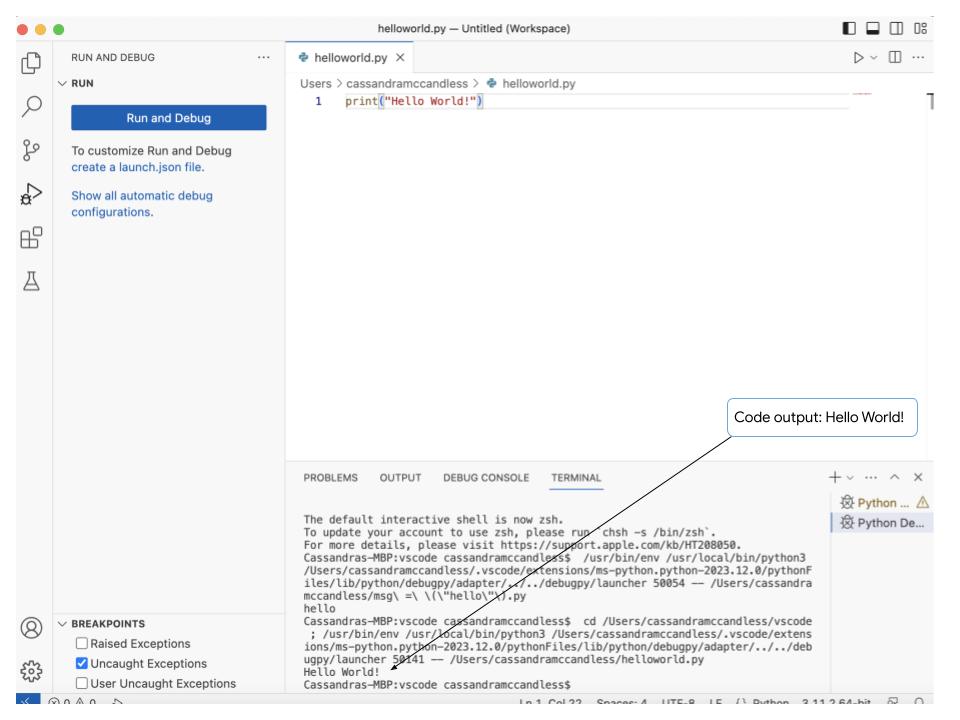
|  |
| --- |
| You will be prompted to name and save your file. Give your Python file a name and click save. |

Bạn sẽ được nhắc đặt tên và lưu tệp của mình. Đặt tên cho tệp Python của bạn và nhấp vào lưu.



|  |
| --- |
| Your code will run and you will see its output here. |

Mã của bạn sẽ chạy và bạn sẽ thấy đầu ra của nó ở đây.



|  |
| --- |
| Now  you know the basics of VS Code and how to use it! |

Bây giờ bạn đã biết kiến ​​thức cơ bản về VS Code và cách sử dụng nó!

|  |
| --- |
| **Key takeaways** |

**Bài học chính**

|  |
| --- |
| VS Code is an extremely robust source code editor. It uses [Intellisense](https://code.visualstudio.com/docs/editor/intellisense#:~:text=IntelliSense%20is%20a%20general%20term,%2C%20and%20%22code%20hinting.%22) technology which provides syntax highlighting and autocomplete for coding. VS code allows you to debug right in the editor with its interactive console. Overall, VS code is extremely interactive and customizable. There is also a large library of extensions that are easily integrated! |

VS Code là một trình soạn thảo mã nguồn cực kỳ mạnh mẽ. Nó sử dụng[trí tuệ](https://code.visualstudio.com/docs/editor/intellisense#:~:text=IntelliSense%20is%20a%20general%20term,%2C%20and%20%22code%20hinting.%22)công nghệ cung cấp tính năng đánh dấu cú pháp và tự động hoàn thành mã hóa. Mã VS cho phép bạn gỡ lỗi ngay trong trình chỉnh sửa bằng bảng điều khiển tương tác của nó. Nhìn chung, mã VS cực kỳ tương tác và tùy biến. Ngoài ra còn có một thư viện lớn các tiện ích mở rộng có thể dễ dàng tích hợp!

|  |
| --- |
| **Resources for more information** |

**Tài nguyên để biết thêm thông tin**

|  |
| --- |
| Here are some resources on what VS Code has to offer! |

Dưới đây là một số tài nguyên về những gì VS Code cung cấp!

|  |
| --- |
| * This [resource](https://code.visualstudio.com/#powerful-debugging) provides an overview of VS Code. * This [resource](https://code.visualstudio.com/docs/languages/python) provides more information on using Python with VS Code and also includes a tutorial you can follow along with. * This is the [extension library](https://marketplace.visualstudio.com/VSCode) for VS Code. |

* Cái này[nguồn](https://code.visualstudio.com/#powerful-debugging)cung cấp cái nhìn tổng quan về Mã VS.
* Cái này[nguồn](https://code.visualstudio.com/docs/languages/python)cung cấp thêm thông tin về cách sử dụng Python với VS Code và cũng bao gồm hướng dẫn mà bạn có thể làm theo.
* Đây là[thư viện mở rộng](https://marketplace.visualstudio.com/VSCode)cho Mã VS.

***4.17. Use VS Code – Sử dụng mã VS***

|  |
| --- |
| - Now that you're familiar with a few different coding tools, let's take a look at VS Code. VS Code is an open source code editor that includes developer tooling. It's similar to Jupyter Notebooks and Colab, but it includes more features. VS Code provides built-in support with IntelliSense code-completion, an interactive debugger, and other build and scripting tools. VS Code has a simple layout that makes it easy to learn and use. Its intuitive features make it a great choice for coding in Python. To use VS code for Python, you will need to have Python 3, VS Code, and the VS Code Python extension installed. In this video, we're gonna show you how to do that. You can use VS Code on Windows, Mac, and Linux operating systems. For this video, we'll be using the Linux operating system. If you're using another operating system, we have provided a reading on using VS Code with the other operating systems that you could take a look at later on. Now let's get started. We'll start by opening a terminal window. Type the following command, python3 --version. |

- Bây giờ thì cậu đã quen rồi.với một vài công cụ mã hóa khác nhau,chúng ta hãy xem Mã VS.VS Code là trình soạn thảo mã nguồn mởbao gồm công cụ dành cho nhà phát triển.Nó tương tự như Jupyter Notebooks và Colab,nhưng nó bao gồm nhiều tính năng hơn.Mã VS cung cấp hỗ trợ tích hợpvới tính năng hoàn thiện mã IntelliSense, một trình gỡ lỗi tương tác,và các công cụ xây dựng và viết kịch bản khác.VS Code có bố cục đơn giảngiúp bạn dễ dàng học và sử dụng.Các tính năng trực quan của nó làm cho nó trở thành một lựa chọn tuyệt vờiđể viết mã bằng Python.Để sử dụng mã VS cho Python,bạn sẽ cần phải có Python 3, Mã VS,và phần mở rộng VS Code Python đã được cài đặt.Trong video này, chúng tôi sẽ chỉ cho bạn cách làm điều đó.Bạn có thể sử dụng Mã VS trên Windows,Hệ điều hành Mac và Linux.Đối với video này, chúng tôi sẽ sử dụng hệ điều hành Linux.Nếu bạn đang sử dụng hệ điều hành khác,chúng tôi đã cung cấp bài đọc về cách sử dụng Mã VSvới các hệ điều hành khácmà bạn có thể xem xét sau này.Bây giờ, hãy bắt đâù.Chúng ta sẽ bắt đầu bằng cách mở một cửa sổ terminal.Nhập lệnh sau, python3 --version.

|  |
| --- |
| If Python 3 is installed, the output will show the Python version that is installed on your computer. An example of what that output would look like is python 3-10-12. If it is not installed, the output will return command not found. This means you'll need to start by installing Python 3. To do this, go to your terminal. At the command line, you will type sudo apt install python3, and press enter. It will prompt you to enter your password. Enter your password, and it will begin installing. As Python 3 is installing, it may prompt you to continue by asking, do you want to continue? Yes slash no. Select yes and press enter. Now Python 3 is installed and ready to go. Let's move on to VS Code. You can download VS Code at codevisualstudio.com/download. |

Nếu Python 3 được cài đặt,đầu ra sẽ hiển thị phiên bản Pythonđược cài đặt trên máy tính của bạn.Một ví dụ về kết quả đầu ra đó sẽ như thế nàolà trăn 3-10-12.Nếu nó không được cài đặt,đầu ra sẽ trả về lệnh không tìm thấy.Điều này có nghĩa là bạn sẽ cần bắt đầu bằng cách cài đặt Python 3.Để làm điều này, hãy đi đến thiết bị đầu cuối của bạn.Tại dòng lệnh, bạn sẽ gõ sudo apt install python3,và nhấn enter.Nó sẽ nhắc bạn nhập mật khẩu của bạn.Nhập mật khẩu của bạn và nó sẽ bắt đầu cài đặt.Khi Python 3 đang cài đặt, nó có thể nhắc bạn tiếp tụcbằng cách hỏi, bạn có muốn tiếp tục không?Có gạch chéo không.Chọn có và nhấn enter.Bây giờ Python 3 đã được cài đặt và sẵn sàng hoạt động.Hãy chuyển sang Mã VS.Bạn có thể tải xuống Mã VS tại codevisualstudio.com/download.

|  |
| --- |
| Here we're going to download the version that is compatible with Linux. Let's select the .deb version of VS Code. We will give it a second to finish downloading. |

Ở đây chúng ta sẽ tải xuống phiên bảntương thích với Linux.Hãy chọn phiên bản .deb của VS Code.Chúng tôi sẽ đợi một giây để hoàn tất tải xuống.

|  |
| --- |
| Next, we're going to go back to the terminal. We will navigate to the downloads folder. We will do this by typing cd Downloads/. Then we'll use the LS command. This will list the Debian file we just downloaded. The name of the file you just downloaded will appear. Next, you will type sudo dpkg -i, and the name of the file listed. |

Tiếp theo, chúng ta sẽ quay lại terminal.Chúng tôi sẽ điều hướng đến thư mục tải xuống.Chúng ta sẽ thực hiện việc này bằng cách gõ cd Downloads/.Sau đó chúng ta sẽ sử dụng lệnh LS.Điều này sẽ liệt kê tệp Debian chúng tôi vừa tải xuống.Tên của file bạn vừa tải xuống sẽ xuất hiện.Tiếp theo, bạn sẽ gõ sudo dpkg -i, và tên của tập tin được liệt kê.

|  |
| --- |
| Once that command is executed, Visual Studio Code is installed. |

Khi lệnh đó được thực thi,Mã Visual Studio đã được cài đặt.

|  |
| --- |
| To open up Visual Studio Code, we type code at the command line, |

Để mở Visual Studio Code,chúng tôi gõ mã ở dòng lệnh,

|  |
| --- |
| and Visual Studio Code will open right up for us. The final step is setting up Python in VS Code. To do this, we will browse for extensions. To view extensions in VS Code, we will type ctrl shift X. This will show you a list of the most popular VS Code extensions on the VS Code marketplace. You can search for Python using the search bar, and click install. Once it's done installing, it will give you the option to get started with Python development. We'll jump right in by selecting create Python file. This will open up the code editor. |

và Visual Studio Code sẽ mở ra ngay cho chúng ta.Bước cuối cùng là thiết lập Python trong VS Code.Để làm điều này, chúng tôi sẽ duyệt tìm các tiện ích mở rộng.Để xem các tiện ích mở rộng trong VS Code, chúng ta sẽ gõ ctrl shift X.Điều này sẽ cho bạn thấy một danh sáchtrong số các tiện ích mở rộng VS Code phổ biến nhấttrên thị trường VS Code.Bạn có thể tìm kiếm Python bằng thanh tìm kiếm,và nhấp vào cài đặt.Sau khi cài đặt xong, nó sẽ cung cấp cho bạn tùy chọnđể bắt đầu phát triển Python.Chúng ta sẽ bắt đầu ngay bằng cách chọn tạo tệp Python.Điều này sẽ mở trình soạn thảo mã.

|  |
| --- |
| Let's test it out with a simple statement. We're going to type print hello world into the new workspace. Next, click on run to run the code. |

Hãy kiểm tra nó bằng một tuyên bố đơn giản.Chúng ta sẽ gõ chữ print hello worldvào không gian làm việc mới.Tiếp theo bấm vào chạy để chạy mã.

|  |
| --- |
| You'll be prompted to name and save your file. Give your Python file a name and click save. In this example, we will give our file the name helloworld.py. Your code will run, and you will see its output here. Great job. Now, to summarize, VS Code is an extremely robust source code editor. It uses IntelliSense technology, which provides syntax highlighting and auto complete for coding. VS Code allows you to debug right in the editor with its interactive console. Overall, VS Code is extremely interactive and customizable. There's also a large library of extensions that are easily integrated. Great job. That was a lot to cover. Feel free to review these videos as many times as you need to. And remember, one of the best ways to figure out which tools work best for you are to explore all the features they have to offer and practice using them. |

Bạn sẽ được nhắc đặt tên và lưu tệp của mình.Đặt tên cho tệp Python của bạn và nhấp vào lưu.Trong ví dụ này, chúng tôi sẽ cung cấp cho tập tin của chúng tôitên helloworld.py.Mã của bạn sẽ chạy và bạn sẽ thấy đầu ra của nó ở đây.Bạn đã làm rất tốt.Bây giờ, để tóm tắt,VS Code là một trình soạn thảo mã nguồn cực kỳ mạnh mẽ.Nó sử dụng công nghệ IntelliSense,cung cấp đánh dấu cú phápvà tự động hoàn thành để mã hóa.VS Code cho phép bạn gỡ lỗi ngay trong trình soạn thảovới bảng điều khiển tương tác của nó.Nhìn chung, VS Code cực kỳ tương tác và tùy biến.Ngoài ra còn có một thư viện tiện ích mở rộng lớncó thể dễ dàng tích hợp.Bạn đã làm rất tốt.Đó là rất nhiều để trang trải.Hãy thoải mái xem lại những video nàybao nhiêu lần tùy thích.Và hãy nhớ, một trong những cách tốt nhấtđể tìm ra công cụ nào phù hợp nhất với bạnlà khám phá tất cả các tính nănghọ phải cung cấp và thực hành sử dụng chúng.

***4.18. More on IDEs and code editors – Thông tin thêm về IDE và trình soạn thảo mã***

|  |
| --- |
| **More on IDEs and code editors** |

**Thông tin thêm về IDE và trình soạn thảo mã**

|  |
| --- |
| There are lots of code editors and IDEs out there. In this reading, we will cover some other popular IDEs. Feel free to try them out and see which ones you like best! |

Có rất nhiều trình soạn thảo mã và IDE. Trong bài đọc này, chúng tôi sẽ đề cập đến một số IDE phổ biến khác. Hãy thử chúng và xem cái nào bạn thích nhất!

|  |
| --- |
| **PyCharm** |

**PyCharm**

|  |
| --- |
| PyCharm is a very popular IDE for Python development. It is an open source IDE that offers many great features to help you write better code faster. It is an intelligent code editor that has useful features, including code completion, error detection, and on-the-fly code fixes. PyCharm also allows you to navigate easily through your code with code folding, code search, and code structure diagrams. The built-in debugger provides breakpoints, step-by-step execution, and variable inspection which allows you to find and fix bugs very quickly. You can test your code within the IDE which allows you to ensure your code is reliable and error free. Finally, PyCharm also offers a wide range of plug-ins created by its community of developers which has expanded its functionality. |

PyCharm là một IDE rất phổ biến để phát triển Python. Đây là một IDE nguồn mở cung cấp nhiều tính năng tuyệt vời giúp bạn viết mã tốt hơn nhanh hơn. Đây là trình chỉnh sửa mã thông minh có các tính năng hữu ích, bao gồm hoàn thành mã, phát hiện lỗi và sửa mã nhanh chóng. PyCharm cũng cho phép bạn điều hướng dễ dàng qua mã của mình bằng cách gấp mã, tìm kiếm mã và sơ đồ cấu trúc mã. Trình gỡ lỗi tích hợp cung cấp các điểm dừng, thực thi từng bước và kiểm tra biến cho phép bạn tìm và sửa lỗi rất nhanh. Bạn có thể kiểm tra mã của mình trong IDE, điều này cho phép bạn đảm bảo mã của mình đáng tin cậy và không có lỗi. Cuối cùng, PyCharm cũng cung cấp nhiều loại plugin được tạo bởi cộng đồng các nhà phát triển đã mở rộng chức năng của nó.

|  |
| --- |
| Overall, PyCharm is a great choice when it comes to an IDE. It is great for beginners because it is easy to use and has a large community of users which means there are tons of tutorials and resources available online. PyCharm is also highly customizable. It is a popular IDE that many developers use and love! |

Nhìn chung, PyCharm là một lựa chọn tuyệt vời khi nói đến IDE. Nó rất tốt cho người mới bắt đầu vì nó dễ sử dụng và có cộng đồng người dùng lớn, điều đó có nghĩa là có rất nhiều hướng dẫn và tài nguyên có sẵn trực tuyến. PyCharm cũng có khả năng tùy biến cao. Đây là một IDE phổ biến được nhiều nhà phát triển sử dụng và yêu thích!

|  |
| --- |
| **PyDev** |

**PyDev**

|  |
| --- |
| PyDev is an open source IDE for Python Development that is based on [Eclipse](https://en.wikipedia.org/wiki/Eclipse_(software)). PyDev includes an intelligent code editor, code navigation, debugging, testing, and refactoring. Its refactoring engine can help you improve the structure and readability of your code which makes maintenance and debugging easier. PyDev is also fully integrated with Eclipse which means you can use all of Eclipse’s features like project explorer, the perspective system, and its built-in debugger. |

PyDev là một IDE mã nguồn mở để phát triển Python dựa trên[nhật thực](https://en.wikipedia.org/wiki/Eclipse_(software)). PyDev bao gồm trình soạn thảo mã thông minh, điều hướng mã, gỡ lỗi, kiểm tra và tái cấu trúc. Công cụ tái cấu trúc của nó có thể giúp bạn cải thiện cấu trúc và khả năng đọc mã của bạn, giúp việc bảo trì và gỡ lỗi dễ dàng hơn. PyDev cũng được tích hợp hoàn toàn với Eclipse, nghĩa là bạn có thể sử dụng tất cả các tính năng của Eclipse như trình khám phá dự án, hệ thống phối cảnh và trình gỡ lỗi tích hợp của nó.

|  |
| --- |
| PyDev is a great IDE for developers at any stage! It's free to use, compatible with many versions of Python, and its plugins make it extensible.  PyDev has a large community of users and developers which means there are lots of tutorials, resources, and forums online. |

PyDev là một IDE tuyệt vời dành cho các nhà phát triển ở mọi giai đoạn! Nó miễn phí sử dụng, tương thích với nhiều phiên bản Python và các plugin của nó giúp nó có thể mở rộng. PyDev có một cộng đồng người dùng và nhà phát triển lớn, điều đó có nghĩa là có rất nhiều hướng dẫn, tài nguyên và diễn đàn trực tuyến.

|  |
| --- |
| **Notepad++** |

**Notepad++**

|  |
| --- |
| Notepad++ is an open source text editor for Microsoft Windows environments. It supports many programming languages, including Python. Notepad++ can open multiple files and tabs. It also does syntax highlighting for known file types. Syntax highlighting helps you edit code because it displays text in different colors and fonts that help you keep track of the classes, methods, functions, and objects in your code. It also helps you identify errors in your code so you can spend less time debugging. Notepad++ has a number of other advanced features, like auto-completion and file management systems that can help you code more efficiently. |

Notepad++ là trình soạn thảo văn bản mã nguồn mở dành cho môi trường Microsoft Windows. Nó hỗ trợ nhiều ngôn ngữ lập trình, bao gồm cả Python. Notepad++ có thể mở nhiều tệp và tab. Nó cũng làm nổi bật cú pháp cho các loại tệp đã biết. Làm nổi bật cú pháp giúp bạn chỉnh sửa mã vì nó hiển thị văn bản với các màu sắc và phông chữ khác nhau giúp bạn theo dõi các lớp, phương thức, hàm và đối tượng trong mã của mình. Nó cũng giúp bạn xác định lỗi trong mã của mình để bạn có thể tốn ít thời gian hơn cho việc sửa lỗi. Notepad++ có một số tính năng nâng cao khác, như hệ thống quản lý tệp và tự động hoàn thành có thể giúp bạn viết mã hiệu quả hơn.

|  |
| --- |
| Notepad++ is a great text editor for developers who work in a Microsoft Windows environment! It has an active, collaborative community. Users can ask the community for help and even contribute to the project! |

Notepad++ là trình soạn thảo văn bản tuyệt vời dành cho các nhà phát triển làm việc trong môi trường Microsoft Windows! Nó có một cộng đồng tích cực và hợp tác. Người dùng có thể yêu cầu cộng đồng trợ giúp và thậm chí đóng góp cho dự án!

|  |
| --- |
| **Key Takeaways** |

**Bài học chính**

|  |
| --- |
| PyCharm, PyDev, and Notepad++ are popular tools used for writing Python code. To try them out, you can download and install them here! |

PyCharm, PyDev và Notepad++ là những công cụ phổ biến được sử dụng để viết mã Python. Để dùng thử, bạn có thể tải xuống và cài đặt chúng tại đây!

|  |
| --- |
| * For more information on PyCharm and to download it,  click [here](https://www.jetbrains.com/pycharm/). * For more information on PyDev and to download it, click [here](https://www.pydev.org/download.html). * For more information on Notepad++ and to download it, click [here](https://notepad-plus-plus.org/). |

* Để biết thêm thông tin về PyCharm và tải xuống, hãy nhấp vào[đây](https://www.jetbrains.com/pycharm/).
* Để biết thêm thông tin về PyDev và tải xuống, hãy nhấp vào[đây](https://www.pydev.org/download.html).
* Để biết thêm thông tin về Notepad++ và để tải xuống, hãy nhấp vào[đây](https://notepad-plus-plus.org/).

|  |
| --- |
| **Resources for more information** |

**Tài nguyên để biết thêm thông tin**

|  |
| --- |
| Now, we have covered different coding tools and how to use them. Code editors and IDEs are essential for writing efficient Python code. Every programmer has their favorite tools, and you can find yours by comparing features and testing them out. |

Bây giờ, chúng ta đã đề cập đến các công cụ mã hóa khác nhau và cách sử dụng chúng. Trình soạn thảo mã và IDE rất cần thiết để viết mã Python hiệu quả. Mỗi lập trình viên đều có các công cụ yêu thích của họ và bạn có thể tìm thấy công cụ của mình bằng cách so sánh các tính năng và thử nghiệm chúng.

|  |
| --- |
| Here are some resources you might find useful as you dive deeper into Python programming. |

Dưới đây là một số tài nguyên bạn có thể thấy hữu ích khi tìm hiểu sâu hơn về lập trình Python.

|  |
| --- |
| * This [blog](https://blog.jupyter.org/) includes the latest news and updates on Jupyter Lab and Jupyter Notebooks. * This [resource](https://code.visualstudio.com/docs/getstarted/tips-and-tricks) includes tons of tips and tricks for using VS Code. * This [reading](https://amitness.com/2020/06/google-colaboratory-tips/) includes more information on some of the features Colab has to offer, including GitHub integration, keyboard shortcuts, and more! * This [resource](https://realpython.com/python-idle/) has tons of information on IDLE. It covers everything from working with files to debugging and more! |

* Cái này[Blog](https://blog.jupyter.org/)bao gồm những tin tức và cập nhật mới nhất về Jupyter Lab và Jupyter Notebooks.
* Cái này[nguồn](https://code.visualstudio.com/docs/getstarted/tips-and-tricks)bao gồm rất nhiều mẹo và thủ thuật để sử dụng VS Code.
* Cái này[đọc](https://amitness.com/2020/06/google-colaboratory-tips/)bao gồm thêm thông tin về một số tính năng mà Colab cung cấp, bao gồm tích hợp GitHub, phím tắt, v.v.!
* Cái này[nguồn](https://realpython.com/python-idle/)có rất nhiều thông tin về IDLE. Nó bao gồm mọi thứ từ làm việc với các tập tin đến gỡ lỗi và hơn thế nữa!

**5. Module Review – Đánh giá mô-đun**

***5.1. First Steps Wrap Up – Kết thúc bước đầu tiên***

|  |
| --- |
| Congrats. You made it to the end of the first module. Great job. You've taken the first steps to learning a new programming language, and growing your IT skill set. Getting there shows real determination and a will to learn. We've covered a lot of topics, and many might be new to you if you've never learned about programming before. You've discovered what scripting is, what the syntax and semantics of a programming language are all about, and how they relate to automation. We've got to grip some small blocks of Python code, talked about why Python is relevant to IT, and explored what other programming languages are available. We've had our first approach to how to input data, and write a script that puts this data to use, and we've seen how you can use Python to perform typical math calculations. Not bad for your first Python steps, right? This is just the beginning of an exciting journey, learning to code, and we hope you're eager to learn more. Coming up, get ready for your first graded assessment. These assessments help you check whether you've understood all the concepts and that you're ready to move on to the next stage. Now, don't worry. If at any point you're not sure about a question, you can always review the videos and readings to remind yourself of the answer. Remember, that everybody learns at different speeds. So take your time, really get familiar with the concepts. Once you feel ready, the assessment is there waiting for you. I'll see you back here once you've nailed it. |

Chúc mừng. Bạn đã hoàn thành xong mô-đun đầu tiên.Bạn đã làm rất tốt. Bạn đã thực hiện những bước đầu tiên để học một ngôn ngữ lập trình mới,và phát triển bộ kỹ năng CNTT của bạn.Đến đó cho thấy quyết tâm thực sự và ý chí học hỏi.Chúng tôi đã đề cập đến rất nhiều chủ đề và nhiều chủ đề có thể mới đối với bạn nếu bạn chưa từngđã học về lập trình trước đó.Bạn đã khám phá ra kịch bản là gì, cú pháp và ngữ nghĩa của mộtngôn ngữ lập trình là tất cả về nó và cách chúng liên quan đến tự động hóa.Chúng ta phải nắm vững một số khối mã Python nhỏ, nói về lý do tại sao Python lạiliên quan đến CNTT và khám phá những ngôn ngữ lập trình khác hiện có.Chúng tôi đã có cách tiếp cận đầu tiên về cách nhập dữ liệu và viết một tập lệnh đặtdữ liệu này để sử dụng và chúng tôi đã thấy cách bạn có thể sử dụng Python để thực hiện các công việc điển hìnhtính toán toán học.Không tệ cho những bước Python đầu tiên của bạn, phải không?Đây chỉ là khởi đầu của một hành trình thú vị, học cách viết mã vàchúng tôi hy vọng bạn mong muốn tìm hiểu thêm.Sắp tới, hãy sẵn sàng cho bài đánh giá xếp loại đầu tiên của bạn.Những đánh giá này giúp bạn kiểm tra xem bạn đã hiểu tất cả các nội dung chưacác khái niệm và bạn đã sẵn sàng chuyển sang giai đoạn tiếp theo.Bây giờ, đừng lo lắng.Nếu tại bất kỳ thời điểm nào bạn không chắc chắn về một câu hỏi, bạn luôn có thể xem lạivideo và bài đọc để nhắc nhở bản thân về câu trả lời.Hãy nhớ rằng mọi người đều học ở tốc độ khác nhau.Vì vậy, hãy dành thời gian của bạn, thực sự làm quen với các khái niệm.Một khi bạn cảm thấy sẵn sàng, việc đánh giá sẽ ở đó chờ bạn.Tôi sẽ gặp lại bạn ở đây sau khi bạn hoàn thành xong việc đó.

***5.2. Meet Marga, the Curriculum Developer – Gặp Marga, Nhà phát triển chương trình giảng dạy***

|  |
| --- |
| My name is Margarita Manterola. I'm a site reliability engineer in the gLinux team. I'm very passionate about teaching programming to everybody. I think it's very important that everybody can learn how to tell the computer what to do. One of the things that really inspired me about the first program was that it was targeted to people that were not from a computer science background. As I, myself don't come from a computer science background, I felt really identified with that and I wanted to help all these people keep growing and keep learning and keep getting more and more skills. That's why I want this program to be successful as well. I got involved in developing this program early on because I was really excited about being able to reach thousands of learners and being able to help all these people grow their skills. Working together with people, it's really important. For this program, I collaborated with a ton of different people from people inside Google, outside Google. I got reviews from people. I got new ideas. I think it's really important because it made the scripts better. The reviewers suggested new things that should be added or pointed things that didn't make sense. I kept incorporating the ideas into the scripts so that the content delivered was the best possible. If it had been just me, that wouldn't have been possible. It's really important to incorporate many different points of view. It's been a lot of work, but I really enjoyed it. I enjoyed the process and it's really exciting to get it to come to life. I hope this program helps people keep growing and keep expanding their horizons so that they can reach their full potential. |

Tên tôi là Margarita Manterola.Tôi là kỹ sư về độ tin cậy của trang web trong nhóm gLinux.Tôi rất đam mêdạy lập trình cho mọi ngườiTôi nghĩ nó rất quan trọngmọi người đều có thể học cách ra lệnh cho máy tính phải làm gì.Một trong những điều thực sự truyền cảm hứngtôi về chương trình đầu tiên là nóđã được nhắm mục tiêu đến những ngườikhông đến từ nền tảng khoa học máy tính.Như tôi, tôi không đếntừ nền tảng khoa học máy tính,Tôi cảm thấy thực sự đồng cảm với điều đó và tôi muốn giúp đỡtất cả những người này tiếp tục phát triển và tiếp tụchọc hỏi và ngày càng trau dồi thêm nhiều kỹ năng.Đó là lý do tại sao tôi muốn chương trình này thành công.Tôi tham gia vào việc phát triểnchương trình này sớm vì tôi thực sự rất hào hứngvề khả năng tiếp cận hàng nghìn người họcvà có thể giúp đỡtất cả những người này đều phát triển kỹ năng của mình.Làm việc cùng với mọi người, điều đó thực sự quan trọng.Đối với chương trình này,Tôi đã cộng tác với rất nhiềunhững người khác với mọi ngườibên trong Google, bên ngoài Google.Tôi đã nhận được đánh giá từ mọi người.Tôi có những ý tưởng mới.Tôi nghĩ nó thực sự quan trọngbởi vì nó làm cho kịch bản tốt hơn.Những người đánh giá đã đề xuất những điều mới cầnđược thêm vào hoặc chỉ ra những điều không có ý nghĩa.Tôi tiếp tục kết hợp các ý tưởng vào kịch bảnđể nội dung được truyền tải là tốt nhất có thể.Nếu chỉ có tôi,điều đó là không thể được.Điều thực sự quan trọng là phải kết hợpnhiều quan điểm khác nhau.Đã có rất nhiều công việc,nhưng tôi thực sự thích nó.Tôi rất thích quá trình này và nó thực sựthú vị để làm cho nó trở nên sống động.Tôi hy vọng chương trình này sẽ giúp mọi người tiếp tục phát triển vàtiếp tục mở rộng tầm nhìn của họ đểhọ có thể phát huy hết tiềm năng của mình.

***5.3. Glossary terms from course 1, module 1 – Thuật ngữ khóa 1, học phần 1***

|  |
| --- |
| **Glossary terms from course 1, module 1** |

**Thuật ngữ khóa 1, học phần 1**

|  |
| --- |
| **Terms and definitions from Course 1, Module 1** |

**Các thuật ngữ và định nghĩa trong Khóa 1, Học phần 1**

|  |
| --- |
| **Automation:** The process of replacing a manual step with one that happens automatically |

**Tự động hóa:** Quá trình thay thế một bước thủ công bằng một bước diễn ra tự động

|  |
| --- |
| **Client-side scripting language:** Primarily for web programming; the scripts are transferred from a web server to the end-user’s internet browser, then executed in the browser |

**Ngôn ngữ kịch bản phía máy khách:** Chủ yếu dành cho lập trình web; các tập lệnh được chuyển từ máy chủ web sang trình duyệt internet của người dùng cuối, sau đó được thực thi trong trình duyệt

|  |
| --- |
| **Code editors:** Tools to provide features, including syntax highlighting, automatic indentation, error checking, and autocompletion |

**Trình chỉnh sửa mã:** Công cụ cung cấp các tính năng, bao gồm tô sáng cú pháp, thụt lề tự động, kiểm tra lỗi và tự động hoàn thành

|  |
| --- |
| **Computer program:** A step-by-step list of instructions that a computer follows to reach an intended goal |

**Chương trình máy tính:** Danh sách từng bước hướng dẫn mà máy tính tuân theo để đạt được mục tiêu đã định

|  |
| --- |
| **Functions:** A reusable block of code that performs a specific task |

**Chức năng:** Một khối mã có thể tái sử dụng để thực hiện một tác vụ cụ thể

|  |
| --- |
| **IDE:** A software application that provides comprehensive facilities for software development |

**IDE:** Một ứng dụng phần mềm cung cấp cơ sở vật chất toàn diện cho việc phát triển phần mềm

|  |
| --- |
| **Interpreter:** The program that reads and executes code |

**Trình thông dịch:** Chương trình đọc và thực thi mã

|  |
| --- |
| **Input:** Information that is provided to a program by the end user |

**Đầu vào:** Thông tin được người dùng cuối cung cấp cho chương trình

|  |
| --- |
| **Logic errors:** Errors in code that prevent it from running correctly |

**Lỗi logic:** Lỗi trong mã khiến mã không thể chạy chính xác

|  |
| --- |
| **Machine language:** Lowest-level computer language. It communicates directly with computing machines in binary code (ones and zeros) |

**Ngôn ngữ máy:** Ngôn ngữ máy tính cấp thấp nhất. Nó giao tiếp trực tiếp với các máy tính bằng mã nhị phân (số một và số không)

|  |
| --- |
| **Object-oriented programming language:** Most coding elements are considered to be objects with configurable properties |

**Ngôn ngữ lập trình hướng đối tượng:** Hầu hết các phần tử mã hóa được coi là đối tượng có thuộc tính có thể định cấu hình

|  |
| --- |
| **Output:** the end result of a task performed by a function or computer program |

**Đầu ra:** kết quả cuối cùng của một tác vụ được thực hiện bởi một chức năng hoặc chương trình máy tính

|  |
| --- |
| **Platform-specific scripting language:** Language used by system administrators on those specific platforms |

**Ngôn ngữ tập lệnh dành riêng cho nền tảng:** Ngôn ngữ được quản trị viên hệ thống sử dụng trên các nền tảng cụ thể đó

|  |
| --- |
| **Programming:** The process of writing a program to behave in different ways |

**Lập trình:** Quá trình viết một chương trình để hoạt động theo những cách khác nhau

|  |
| --- |
| **Programming code**: A set of written computer instructions, guided by rules, using a computer programming language |

**Mã lập trình** : Một tập hợp các lệnh máy tính bằng văn bản, được hướng dẫn bởi các quy tắc, sử dụng ngôn ngữ lập trình máy tính

|  |
| --- |
| **Programming languages:** Language with syntax and semantics to write computer programs |

**Ngôn ngữ lập trình:** Ngôn ngữ có cú pháp và ngữ nghĩa để viết chương trình máy tính

|  |
| --- |
| **Python:** A general purpose programming language |

**Python:** Ngôn ngữ lập trình có mục đích chung

|  |
| --- |
| **Python interpreter:** Program that reads and executes Python code by translating Python code into computer instructions |

**Trình thông dịch Python:** Chương trình đọc và thực thi mã Python bằng cách dịch mã Python thành các lệnh máy tính

|  |
| --- |
| **Script:** Often used to automate specific tasks |

**Script:** Thường được sử dụng để tự động hóa các tác vụ cụ thể

|  |
| --- |
| **Semantics:** The intended meaning or effect of statements, or collections of words, in both human and computer languages |

**Ngữ nghĩa:** Ý nghĩa hoặc tác dụng dự kiến ​​của các câu lệnh hoặc tập hợp các từ trong cả ngôn ngữ của con người và máy tính

|  |
| --- |
| **Syntax:** The rules for how each statements are constructed in both human and computer languages |

**Cú pháp:** Các quy tắc về cách xây dựng mỗi câu lệnh trong cả ngôn ngữ của con người và máy tính

|  |
| --- |
| **Variables:** These are used to temporarily store changeable values in programming code |

**Biến:** Chúng được sử dụng để lưu trữ tạm thời các giá trị có thể thay đổi trong mã lập trình

***5.4. Study Guide: Module 1 Graded Quiz – Hướng dẫn học: Bài kiểm tra chấm điểm Học phần 1***

***5.5. Module 1 Graded Assessment – Đánh giá theo điểm học phần 1***

***5.6. Your First Programming Code – Mã lập trình đầu tiên của bạn***

**Module 2: Basic Python Syntax – Cú pháp Python cơ bản**

|  |
| --- |
| In this module you’ll learn about different data types in Python, how to identify them, and how to convert between them. You’ll also learn how to use variables to assign data and to reference variables. You’ll deep dive into functions: how to define them, pass them parameters, and have them return information. You’ll explore the concepts of code reuse, code style, and refactoring complex code, along with effectively using code comments. Finally, you’ll learn about comparing data using equality and logical operators, and leveraging these to build complex branching scripts using if statements. |

Trong mô-đun này, bạn sẽ tìm hiểu về các loại dữ liệu khác nhau trong Python, cách xác định chúng và cách chuyển đổi giữa chúng. Bạn cũng sẽ tìm hiểu cách sử dụng các biến để gán dữ liệu và tham chiếu các biến. Bạn sẽ đi sâu vào các hàm: cách xác định chúng, truyền tham số cho chúng và yêu cầu chúng trả về thông tin. Bạn sẽ khám phá các khái niệm về tái sử dụng mã, kiểu mã và tái cấu trúc mã phức tạp, cùng với việc sử dụng nhận xét mã một cách hiệu quả. Cuối cùng, bạn sẽ tìm hiểu về cách so sánh dữ liệu bằng cách sử dụng các toán tử đẳng thức và logic, đồng thời tận dụng chúng để xây dựng các tập lệnh phân nhánh phức tạp bằng cách sử dụng câu lệnh if.

|  |
| --- |
| **Learning Objectives**   * Differentiate and convert between different data types utilizing variables * Define and call functions utilizing parameters and return data * Refactor code and write comments to reduce complexity and enhance code readability and code reuse * Compare values using equality operators and logical operators * Build complex branching scripts utilizing if, else and elif statements |

**Mục tiêu học tập**

* Phân biệt và chuyển đổi giữa các loại dữ liệu khác nhau bằng cách sử dụng các biến
* Xác định và gọi hàm bằng cách sử dụng tham số và trả về dữ liệu
* Cấu trúc lại mã và viết nhận xét để giảm độ phức tạp và nâng cao khả năng đọc mã và tái sử dụng mã
* So sánh các giá trị bằng toán tử đẳng thức và toán tử logic
* Xây dựng các tập lệnh phân nhánh phức tạp bằng cách sử dụng các câu lệnh if, else và elif

**1. Expressions and Variables – Biểu thức và biến**

***1.1. Basic Python Syntax introduction – Giới thiệu cú pháp Python cơ bản***

|  |
| --- |
| Hi there, welcome back, and well done for completing your first graded assessment. You're doing a great job making it this far. Chances are some topics we've covered may have been a little tricky at times, especially if you're completely new to programming. Don't worry if something wasn't obvious right away. We went through a lot of new concepts and it might take several passes until you feel comfortable with them. And that's totally normal. We all went through it when we were learning how to code. In the previous module we explored some basic concepts, like programming and automation. We called out that each programming language has a specific syntax, which we need to learn so we can tell the computer what to do. We then got a sneak preview of some of the things we could do with Python. Up next, we'll dive deeper into some basic building blocks of Python syntax, things like variables, expressions, functions, and conditional blocks. At first glance these pieces may seem pretty simple, but when we start to combine them they become a lot more powerful. Understanding a programming language's syntax isn't too different from learning a spoken language. For example, the best way to learn Spanish is to visit a Spanish speaking country, immerse yourself in the culture, listen to the people. Then figure out how to arrange the words to form a sentence that another speaker can understand. The same is true for programming. When you immerse yourself in Python programming you'll learn how to formulate statements of code that the computer can understand. This is called syntax. Okay, so as you go through the next few videos keep in mind that our main goal is to learn the language's syntax. So we'll focus on how to tell our computer what to do, not on how to get it to do complicated tasks. Like before, we'll run through some simple exercises to help you see the concepts in action. And as you pick up the new skills and get to grips with different tools we'll start to write more advanced scripts that tackle more challenging problems. Again, if at any point you feel confused or that something just isn't clear, remember you can watch the videos and take the practice quizzes as many times as you need. The key to getting good at programming is practice, practice and practice. You have to keep working your programming muscles in order to get strong, just like building muscles in the gym. Train hard, train regularly, and you'll be tackling more weighty coding problems in no time. All right, ready to jump back in? In the next video we're going to learn all about data types. Let's get started. |

Xin chào, chào mừng bạn đã trở lại và bạn đã hoàn thành tốt bài học đầu tiên của mìnhđánh giá. Bạn đang làm rất tốt khi tiến xa đến mức này.Rất có thể một số chủ đề chúng tôi đề cập đôi khi có thể hơi phức tạp,đặc biệt nếu bạn là người hoàn toàn mới làm quen với lập trình.Đừng lo lắng nếu có điều gì đó chưa rõ ràng ngay lập tức.Chúng tôi đã trải qua rất nhiều khái niệm mới và có thể phải mất vài lần mới thực hiện đượcbạn cảm thấy thoải mái với họ.Và điều đó hoàn toàn bình thường.Tất cả chúng ta đều đã trải qua điều đó khi học cách viết mã.Trong mô-đun trước, chúng ta đã khám phá một số khái niệm cơ bản, như lập trình vàtự động hóa. Chúng tôi chỉ ra rằng mỗi ngôn ngữ lập trình có một đặc điểm riêngcú pháp mà chúng ta cần học để có thể ra lệnh cho máy tính phải làm gì.Sau đó, chúng tôi đã xem trước một số điều chúng tôi có thể làm với Python.Tiếp theo, chúng ta sẽ đi sâu hơn vào một số khối xây dựng cơ bản của cú pháp Python,những thứ như biến, biểu thức, hàm và khối điều kiện.Thoạt nhìn những phần này có vẻ khá đơn giản, nhưng khi chúng ta bắt đầukết hợp chúng lại chúng trở nên mạnh mẽ hơn rất nhiều.Hiểu cú pháp của ngôn ngữ lập trình không quá khác biệt so với việc họcmột ngôn ngữ nói. Ví dụ, cách tốt nhất để học tiếng Tây Ban Nha là đến thăm một trường học.Đất nước nói tiếng Tây Ban Nha, hòa mình vào văn hóa, lắng nghemọi người. Sau đó tìm cách sắp xếp các từ để tạo thành một câungười nói khác có thể hiểu được.Điều này cũng đúng với việc lập trình.Khi bạn đắm mình vào lập trình Python, bạn sẽ học cáchxây dựng các câu lệnh mã mà máy tính có thể hiểu được.Đây được gọi là cú pháp.Được rồi, khi bạn xem qua một số video tiếp theo, hãy nhớ rằng mục tiêu chính của chúng talà học cú pháp của ngôn ngữ.Vì vậy, chúng ta sẽ tập trung vào cách ra lệnh cho máy tính phải làm gì, chứ không phải làm thế nào để có đượcnó để thực hiện những công việc phức tạp.Giống như trước đây, chúng ta sẽ thực hiện một số bài tập đơn giản để giúp bạn thấy đượccác khái niệm trong hành động.Và khi bạn tiếp thu các kỹ năng mới và làm quen với các công cụ khác nhau, chúng tôi sẽbắt đầu viết các tập lệnh nâng cao hơn để giải quyết các vấn đề khó khăn hơn.Xin nhắc lại, nếu tại bất kỳ thời điểm nào bạn cảm thấy bối rối hoặc có điều gì đó không ổnrõ ràng, hãy nhớ rằng bạn có thể xem video và làm các câu hỏi thực hành nhưnhiều lần khi bạn cần.Chìa khóa để giỏi lập trình là thực hành, thực hành và thực hành.Bạn phải tiếp tục rèn luyện cơ bắp lập trình của mình để có đượcmạnh mẽ, giống như xây dựng cơ bắp trong phòng tập thể dục.Hãy rèn luyện chăm chỉ, luyện tập thường xuyên và bạn sẽ giải quyết được những vấn đề mã hóa nặng nề hơnkhông có thời gian. Được rồi, sẵn sàng quay trở lại chưa?Trong video tiếp theo, chúng ta sẽ tìm hiểu tất cả về các loại dữ liệu.Bắt đầu nào.

***1.2. Explore Python syntax – Khám phá cú pháp Python***

|  |
| --- |
| **Explore Python syntax** |

|  |
| --- |
| Python is a flexible programming language used in a wide range of fields, including software development, machine learning, and data analysis. Python is one of the most popular programming languages for data professionals, so getting familiar with its fundamental syntax and semantics will be useful for your future career. In this reading, you will learn about Python’s syntax and semantics, as well as where to find resources to further your learning. |

|  |
| --- |
| **The Language of Python** |

|  |
| --- |
| People use language to communicate and give instructions to each other. Computers do the same thing, except computers use languages like Python, C++, and Java. So, in order to communicate instructions to the computer, programmers need to arrange ideas and concepts into a language it will understand. |

|  |
| --- |
| Python syntax includes words that represent objects and commands, as well as punctuation that gives the words structure, hierarchy, and context. Together, the words and punctuation communicate ideas and processes; this is known as semantics. Semantics is the meaning conveyed by the syntax. The best way to learn syntax and semantics is through exposure. Practice coding and become familiar and comfortable with reading other people’s code. In addition, there are some general conventions that practitioners use to help maintain stylistic uniformity within the language. |

|  |
| --- |
| Coding languages are similar to spoken languages in that they have a way to classify words according to their function. For example, English sentences are composed of nouns, verbs, prepositions, etc. Here are some of the basics: |

|  |
| --- |
| * **Variables:** Represent data stored as strings, tuples, dictionaries, lists, and objects (note: future readings explain these categories) * **Keywords:** Special words that are reserved for specific purposes and that can only be used for those purposes |

|  |
| --- |
| in  not  or  for  while  return |

in

not

or

for

while

return

|  |
| --- |
| * **Operators:** Symbols that perform operations on objects and values |

|  |
| --- |
| +  -  \*  /  \*\*  %  //  >  <  == |

+

-

\*

/

\*\*

%

//

>

<

==

|  |
| --- |
| * **Expressions:** A combination of numbers, symbols, and variables to compute and return a result upon evaluation * **Functions:** A group of related statements to perform a task and return a value |

|  |
| --- |
| def to\_celsius(x):     '''Convert Fahrenheit to Celsius'''     return (x-32) \* 5/9  to\_celsius(75) |

def to\_celsius(x):

   '''Convert Fahrenheit to Celsius'''

   return (x-32) \* 5/9

to\_celsius(75)

|  |
| --- |
| * **Conditional statements:** Sections of code that direct program execution based on specified conditions |

|  |
| --- |
| number = -4  if number > 0:     print('Number is positive.')  elif number == 0:     print('Number is zero.')  else:     print('Number is negative.') |

number = -4

if number > 0:

   print('Number is positive.')

elif number == 0:

   print('Number is zero.')

else:

   print('Number is negative.')

|  |
| --- |
| As you’ll surely discover, Python generates syntax errors for incorrectly used keywords and syntax. |

|  |
| --- |
| Example:  print(This will throw an error because I didn’t make it a string.) |

|  |
| --- |
| **Naming rules and conventions** |

|  |
| --- |
| When assigning names to objects, programmers adhere to a set of rules and conventions which help to standardize code and make it more accessible to everyone. Here are some naming rules and conventions that you should know: |

|  |
| --- |
| * Names cannot contain spaces. * Names may be a mixture of upper and lower case characters. * Names can’t start with a number but may contain numbers after the first character. * Variable names and function names should be written in snake\_case, which means that all letters are lowercase and words are separated using an underscore. * Descriptive names are better than cryptic abbreviations because they help other programmers (and you) read and interpret your code. For example, student\_name is better than sn. It may feel excessive when you write it, but when you return to your code you’ll find it much easier to understand. |

|  |
| --- |
| Tim Peters, a Python programmer, wrote this now-famous “poem” of guiding principles for coding in Python: |

|  |
| --- |
| **The Zen of Python** |

|  |
| --- |
| Beautiful is better than ugly.  Explicit is better than implicit.  Simple is better than complex.  Complex is better than complicated.  Flat is better than nested.  Sparse is better than dense.  Readability counts.  Special cases aren't special enough to break the rules.  Although practicality beats purity.  Errors should never pass silently.  Unless explicitly silenced.  In the face of ambiguity, refuse the temptation to guess.  There should be one—and preferably only one—obvious way to do it.  Although that way may not be obvious at first unless you're Dutch.  Now is better than never.  Although never is often better than \*right\* now.  If the implementation is hard to explain, it's a bad idea.  If the implementation is easy to explain, it may be a good idea.  Namespaces are one honking great idea -- let's do more of those! |

|  |
| --- |
| Finally, it’s helpful to bookmark the [PEP 8 Style Guide for Python](https://peps.python.org/pep-0008/) so you can reference it as needed. This reading is limited in scope, and PEP 8 is a more exhaustive resource for style-related matters. PEP stands for Python Enhancement Proposals. These are a running catalog of ways to improve or standardize Python as a language. Because Python is open source, PEP offers a framework to guide developers and build consensus around ideas. It’s a useful and trusted resource. |

|  |
| --- |
| **Key takeaways** |

|  |
| --- |
| Syntax and semantics are what give form and meaning to a language, including Python.  A large part of learning a new language is familiarizing yourself with its syntax and semantics. Much of this comes through exposure and practice, but there are a few guiding principles and resources that can help you along the way. If you learn the rules about naming objects and build a bank of resources that you can reference for guidance, you’ll surely make progress as a Python learner. As you get more familiar with Python, you’ll be able to communicate more efficiently with computers and do more with your data analysis tools! |

|  |
| --- |
| **Resources for more information** |

|  |
| --- |
| Here are a few useful resources to help you get more familiar with Python:   * Python [Reference Library](https://docs.python.org/3/library/)   + [Built-in Data types](https://docs.python.org/3/library/stdtypes.html)   + [Built-in functions](https://docs.python.org/3/library/functions.html#built-in-functions) * [Python operators](https://python-reference.readthedocs.io/en/latest/docs/operators/index.html) |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

***1.3. Review: Data types – Ôn tập: Các kiểu dữ liệu***

***1.4. Data Types – Kiểu dữ liệu***

***1.5. Categorize: Naming variables – Phân loại: Đặt tên biến***

***1.6. Annotating variables by type – Chú thích các biến theo loại***

***1.7. Review: Expressions, numbers, and type conversions – Ôn tập: Chuyển đổi biểu thức, số và loại***

***1.8. Expressions, numbers, and type conversions – Chuyển đổi biểu thức, số và loại***

***1.9. Implicit vs explicit conversion – Chuyển đổi ngầm định và rõ ràng***

***1.10. Study guide: Expressions and variables – Hướng dẫn học: Biểu thức và biến***

***1.11. Practice quiz: Expressions and variables –Bài tập thực hành: Biểu thức và biến***

**2. Functions – Hàm**

***2.1. Review: Defining functions – Ôn tập: Xác định hàm***

***2.2. Defining Functions – Xác định hàm***

***2.3. Built-in functions – Chức năng tích hợp sẵn***

***2.4. Review: Returning values – Đánh giá: Trả về giá trị***

***2.5. Returning Values – Giá trị trả về***

***2.6. Review: The principles of code reuse – Đánh giá: Các nguyên tắc tái sử dụng mã***

***2.7. The principles of code reuse – Nguyên tắc tái sử dụng mã***

***2.8. Review: Code style – Đánh giá: Kiểu mã***

***2.9. Code style – Kiểu mã***

***2.10. Study guide: Functions – Hướng dẫn học: Hàm***

***2.11. Practice Quiz: Functions – Bài kiểm tra thực hành: Chức năng***

**3. Conditionals – Câu điều kiện**

***3.1. Review: Comparing things***

***3.2. Comparing things***

***3.3. Comparison Operators with Equations***

***3.4. Comparison Operators with Strings***

***3.5. Logical Operators***

***3.6. Review: Branching with if statements***

***3.7. Branching with if Statements***

***3.8. Review: else statements***

***3.9. else Statements***

***3.10. else Statements and the Modulo Operator***

***3.11. Review: elif statements***

***3.12. elif Statements***

***3.13. Complex Branching with elif Statements***

***3.14. Explore: If, elif, and else statements***

***3.15. Study Guide: Conditionals***

***3.16. Practice Quiz: Conditionals***

***4. Module Review – Đánh giá mô-đun***

***4.1. In Marga's Words: Why I Like Python – Theo lời của Marga: Tại sao tôi thích Python***

***4.2. Glossary terms from course 1, module 2 – Thuật ngữ khóa 1, học phần 2***

***4.3. Study Guide: Module 2 Graded Quiz – Hướng dẫn học: Bài kiểm tra chấm điểm Học phần 2***

***4.4. Module 2 Graded Assessment – Đánh giá theo điểm học phần 2***

***4.5. Basic Syntax Wrap Up – Tóm tắt cú pháp cơ bản***

***4.6. Basic Python Syntax Review – Đánh giá cú pháp Python cơ bản***