DAVID MALAN: Hello, world.

My name is David Malan, and this is CS50's Introduction

to Programming with Python.

Whereas CS50 itself is an introduction to the intellectual enterprises

of computer science and the art of programming,

this course is specifically focused on programming in Python itself.

At the beginning of the course, we'll be focused

on a topic in programming known as functions and variables, mechanisms

via which you can write code that solves smaller problems,

but you can compose those smaller solutions into solutions

to larger problems still.

We'll then transition to a look at conditionals,

a way in code of expressing yourself logically, to maybe do something

if some question has an answer of true, or not

do something if the answer is false.

We'll transition thereafter to introducing you

to loops, the ability, in code, to do something again, and again,

and again some number of times.

We'll then transition to something a little more technical,

known as exceptions.

Unfortunately, a lot can go wrong when you're writing code, some of it

your fault, some of it perhaps someone else's fault.

But you can write code defensively, so to speak, and actually

catch those kinds of exceptions, those errors,

and handle them properly so that the users you're writing code for

don't actually see the same.

Thereafter, we'll take a look at libraries, third-party code,

written by other people, often, or perhaps yourself in the past,

that you can use and reuse in your own projects

so as to avoid reinventing the wheel again and again.

We'll look thereafter at something called unit tests.

It turns out, you'll actually write code to test your own code.

But you won't have to write tests for your tests.

Indeed, this is a best practice in industry, writing tests for your code

so that one, you can be sure that your code today is, hopefully,

if your tests are correct, correct itself.

But moreover, if you or someone else modifies your code tomorrow, or down

the line, you can rerun those same tests to ensure

that those new changes have not broken anything about your own code.

We'll then take a look at something called

File I/O, I/O for input and output, the ability

to not just store information inside of a computer's memory,

but rather save it persistently to disk, so to speak, to files and folders.

We'll then take a look at another technique, known

as regular expressions, whereby, in Python, you can define patterns

and you can validate data to make sure the human typed something in as you

expect.

You can use regular expressions to extract data, perhaps

from some data set you're trying to analyze.

We'll then take a look, ultimately, at object-oriented programming,

a paradigm, a way of writing code, whereby you can represent, in code,

real-world entities.

And this is in addition to other paradigms of programming that we'll

also explore, among them procedural programming,

where you write lots of those functions, procedures

really, top to bottom, to solve problems step by step,

and even something known as functional programming, as well.

And then at the very end of the course will we

equip you with all the more tools for your toolkit.

and additional building blocks, additional vocabulary

via which, after this same course, you can go off on your own

and either take other courses or solve projects of your own,

using all of these mechanisms.

Now this course itself assumes no prior programming background.

So you don't have to have written a single line of code in Python,

or any language, yet.

But this is also a course that you can take before, during, or even after CS50

itself, if you'd like to get all the more versed with Python.

Each week, via the courses lectures, will we introduce you

to any number of concepts that we'll then

drill down more deeply into in the form of problem sets each week.

That is, programming projects that will enable

you to apply some of those lessons learned to problems of your very own.

And by the end of the course, you'll have

solved so many problems that, ideally, are representative

of problems you'll eventually encounter in the real world,

whether you aspire to solve code in the technical world

or perhaps in the arts, the humanities, the social sciences,

the natural sciences, or beyond.

You'll have, ultimately, the vocabulary and the technical skills

via which to approach the same.

This, then, is CS50.

And this is CS50's Introduction to Programming with Python.