SLIDE SCRIPT

Slide Script

Every slide in this lesson has speaker notes. They say what to point to, what to say out loud and what answers to listen for. This script prints the same notes, day by day and slide by slide, so you can read ahead the night before or teach from paper.

Where to find the speaker notes

1. In PowerPoint, the notes sit in the box under each slide. If you can't see it, click Notes in the bar at the bottom of the window.
2. When you present, use Presenter View. You see the notes on your laptop, and students see only the slide.
3. In Google Slides, the notes sit in the box under each slide. If you can't see it, open the View menu and choose Show speaker notes.
4. Say: marks words to read out loud, and Ask: marks a question for the class. Every other line tells you what to do.

The lesson cover

Slide 1. Lesson cover

- **Say:** This lesson is Your First Program. It runs 4 days.
- **Ask:** How do you write a program that shows words on the screen?

Day 1 — What a Program Is

Slide 2. Day 1 divider: What a Program Is

- **Say:** Today you will learn what a program is and how `print()` shows text on the screen, one line at a time.

Slide 3. This phone says good morning.

- **Say:** This app shows a message on the screen.
- **Ask:** How does the phone know what words to show? Take two or three answers.
- **Say:** A person wrote code that tells the phone what to show. Today you'll learn what a program is and how it shows words on the screen.

Slide 4. Here are two words for today.

- **Say:** A program is a list of steps. The computer follows them in order.
- **Say:** The output is what the program shows on the screen, like the app's good morning message.
- **Say:** Today you'll read programs and figure out their output.

Slide 5. A program is a list of steps.

- **Say:** Think of the fruit cup steps from the bell ringer. The order mattered.
- **Say:** A program works the same way. The computer follows each step in order.
- **Say:** We write the steps as code. Each step goes on its own line of code.

Slide 6. `print()` shows text on the screen.

- **Say:** This program has one line of code. print is a command that shows text.
- Point to the parentheses. **Say:** What prints goes inside them.
- Point to the quote marks. **Say:** Text inside quote marks is called a string. The quotes show where it starts and stops.
- Point to the Output panel. **Say:** The quotes don't print. Only the text inside them shows.

Slide 7. Which line prints Hi! exactly?

- **Ask:** Which line prints Hi! exactly? Have students vote with 1, 2 or 3 fingers.
- **Say:** Option 1 is right. The text is inside quote marks, with the capital H and the exclamation point.
- **Say:** Option 2 has no quote marks, so Python stops and shows a message. Option 3 prints a small h and no exclamation point, so it doesn't match.

Slide 8. Predict what this program prints.

- **Say:** Read the program from top to bottom, one line at a time.
- **Say:** Write what prints in the What prints box in your notes before we look.
- Walk around and look for students who write all three steps on one line, or who write the quote marks.

Slide 9. Each `print()` starts a new line.

- **Say:** Line 1 prints first, then line 2, then line 3. Python runs the lines from top to bottom.
- Point to the Output panel. **Say:** Each print() starts a new line, so three print() lines make three lines of output.
- **Say:** The quote marks don't show in the output. Only the text inside them prints.

Slide 10. Code and output are not the same.

- **Say:** Code is what we write. Output is what the screen shows when the code runs.
- Point to the code. **Say:** The code has print(), parentheses and quote marks.
- Point to the Output panel. **Say:** The output shows only the text that was inside the quotes, Good morning! and Today is sunny.
- **Ask:** What's in the code that isn't in the output? Listen for: print, the parentheses and the quote marks.

Slide 11. Predict the output of eight programs.

- Hand out Predict the Output, 1 per student.
- **Say:** Read each program from top to bottom. Write each line of output on its own line, letter for letter, with no quote marks.
- **Say:** Part B has small details to watch, like capital letters and periods. Computers stay closed today.
- **Say:** You have 19 minutes to work. Then you'll check with a partner, and we'll go over the answers.

Slide 12. Big idea: A program is a list of steps that run in order.

- **Say:** Here's today's big idea. A program is a list of steps, and Python runs them from top to bottom.
- Hand out the exit ticket.

Day 2 — First Words in the Editor

Slide 13. Day 2 divider: First Words in the Editor

- **Say:** Today you will learn how to write `print()` lines in a code editor, match an Example output exactly and explain code with comments.

Slide 14. Every letter counts in the output.

- **Say:** Let's look back at the bell ringer.
- **Ask:** Which output matched the sign? Take an answer. **Say:** B, because A had a small l and no period.
- **Say:** Today you'll write code in an editor, and your output has to match the Example exactly.

Slide 15. A code editor has two sides.

- **Say:** CodeGrove has a code editor. You type your code on one side, and the output shows on the other.
- Point to the tab. **Say:** Your code is saved in a file named main dot py. The Run button runs it from top to bottom.
- Point to the Output panel. **Say:** It shows exactly what your program prints, one line for each print(). Two print() lines make two lines, Hello! and This is my first program.

Slide 16. A comment is a note for people.

- Point to line 1. **Say:** This line starts with a hash mark, so it's a comment.
- **Say:** A comment is a note for the people who read the code. Python skips it.
- Point to the Output panel. **Say:** Only the print() line shows. The comment never prints.

Slide 17. Predict what this program prints.

- **Say:** Read the program from top to bottom. Remember what Python does with a comment.
- **Say:** Write what prints in the What prints box in your notes.
- Walk around and look for students who copy the comment into the output.

Slide 18. Python skips the comment.

- Point to line 1. **Say:** Python skips the comment, so it doesn't print.
- Point to the Output panel. **Say:** Two print() lines make two lines of output.
- **Ask:** Who wrote the comment in the output? That's the most common mistake today.

Slide 19. Which output matches the Example `Lunch is at noon.` exactly?

- **Ask:** Which output matches the Example exactly? Have students vote with 1, 2 or 3 fingers.
- **Say:** Option 3 is right. It has the capital L and the period.
- **Say:** Option 1 has a small l. Option 2 has no period. Check counts both as wrong.

Slide 20. Check is a helper, not your grade.

- **Say:** In CodeGrove, Check runs your code and compares its output with the Example.
- **Say:** Check only tells you if your code works. Your grade comes from your sheet.
- **Say:** On paper, you check your code by reading it from top to bottom.

Slide 21. Write three programs in First Words.

- Hand out the First Words sheet, 1 per student. Print only one version.
- **Say:** If you have the ONLINE sheet, write your code in CodeGrove, press Run and press Check. Write your lines on the sheet as you go.
- **Say:** If you have the PAPER sheet, write your code on the sheet and read it from top to bottom to check it.
- **Say:** You have about 14 minutes for the programs and 4 more minutes to finish your boxes. Then we'll go over them.

Slide 22. Big idea: Match the Example output exactly.

- **Say:** Here's today's big idea. Your output must match the Example exactly, and a comment never prints.
- Hand out the exit ticket.

Day 3 — Make a Sign

Slide 23. Day 3 divider: Make a Sign

- **Say:** Today you will learn how an empty `print()` prints a blank line and how spaces and symbols inside quotes print exactly.

Slide 24. This sign is easy to read.

- **Say:** Here's a sign on a school door.
- **Ask:** What makes it easy to read? Take two or three answers, like the stars or the empty space.
- **Say:** Today you'll write programs that print signs like this one, with borders and blank lines.

Slide 25. An empty `print()` prints a blank line.

- Point to line 2. **Say:** This print() has nothing inside its parentheses.
- Point to the Output panel. **Say:** An empty print() prints a blank line.
- **Say:** Blank lines split a sign into parts, so it's easier to read.

Slide 26. Predict what this program prints.

- **Say:** Read the program from top to bottom. Count the print() lines.
- **Say:** Write what prints in the What prints box in your notes. Leave a blank line for each empty print().
- Walk around and look for students who skip the blank lines.

Slide 27. Each empty `print()` makes one blank line.

- Point to lines 2 and 4. **Say:** Each empty print() makes one blank line.
- Point to the Output panel. **Say:** Five print() lines make five lines of output, and two of them are blank.
- **Ask:** Who wrote only three lines? A blank line still counts as a line.

Slide 28. Symbols and spaces print exactly.

- Point to line 1. **Say:** Each star and each space inside the quotes prints.
- **Say:** A row of symbols on the top and the bottom is called a border.
- Point to line 2. **Say:** The two spaces before ART SHOW print too, so the words move to the right.

Slide 29. Plan the output first.

- **Say:** Before you write code, plan the output. Write each line of the sign in order.
- **Say:** Then write one print() for each line, and an empty print() for each blank line.
- **Say:** A comment above each section of code tells the reader which part of the sign it makes.

Slide 30. Make three signs in Sign Maker.

- Hand out the Sign Maker sheet, 1 per student. Print only one version.
- **Say:** If you have the ONLINE sheet, write your code in CodeGrove, press Run and press Check. Write your lines on the sheet as you go.
- **Say:** If you have the PAPER sheet, write your code on the sheet and read it from top to bottom to check it.
- **Say:** You have about 14 minutes for the signs and 4 more minutes to finish your boxes. Then we'll go over them.

Slide 31. Big idea: Blank lines and symbols make a sign readable.

- **Say:** Here's today's big idea. An empty print() prints a blank line, and every symbol and space inside the quotes prints.
- Hand out the exit ticket.

Day 4 — Quotes and Careful Output

Slide 32. Day 4 divider: Quotes and Careful Output

- **Say:** Today you will learn how to use a matching pair of quotes and why text with an apostrophe goes in double quotes.

Slide 33. Which line runs and prints the apostrophe?

- **Say:** These are Lines A, B and C from the bell ringer. Here they're lines 1, 2 and 3, and each line is its own try.
- **Ask:** Which line runs and prints the apostrophe? Have students vote with 1, 2 or 3 fingers.
- **Say:** Option 2 is right. Double quotes hold the whole line, apostrophe and all.
- **Say:** In line 1, the apostrophe ends the text too soon, so Python stops and shows a message. Line 3 runs, but its text has no apostrophe.

Slide 34. Both kinds of quotes make text.

- **Say:** A string can use double quotes or single quotes. Both make text.
- Point to the left. **Say:** This print() uses double quotes, and it prints Hi. The quote marks don't print.
- Point to the right. **Say:** This one uses single quotes, and it prints Hi too. Each string starts and ends with the same kind of quote.

Slide 35. Text with an apostrophe goes in double quotes.

- Point to line 2. **Say:** Don't has an apostrophe, so the text goes in double quotes.
- Point to line 4. **Say:** Walk, please has no apostrophe, so single quotes work fine.
- Point to the Output panel. **Say:** The comment doesn't print, and the empty print() makes a blank line.

Slide 36. A missing quote stops the program.

- **Say:** Line 2 has no closing quote, so Python can't tell where the text ends.
- Point to the Error box. **Say:** Python stops and shows a message. It names line 2, the line with the missing quote.
- **Say:** Nothing prints, not even line 1. The fix is to give every quote a partner of the same kind.

Slide 37. Make the Field Trip Flyer.

- Hand out the Field Trip Flyer sheet, 1 per student. Print only one version.
- **Say:** Work alone today. If you have the ONLINE sheet, write the program in CodeGrove and press Check. Then write your lines on the sheet.
- **Say:** If you have the PAPER sheet, write the program on the sheet and read it from top to bottom to check it.
- **Say:** You have 12 minutes. Then you'll take the lesson quiz.

Slide 38. Big idea: Quotes come in matching pairs.

- **Say:** Here's today's big idea. Quotes come in matching pairs, and text with an apostrophe goes in double quotes.
- Hand out the exit ticket.