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 Welcome Back: The Python You Know. It runs 3 days.
- **Ask:** How can update operators and the editor's tools help you write and fix code faster?

Day 1 — Update Operators and Editor Tools

Slide 2. Day 1 divider: Update Operators and Editor Tools

- **Say:** Today you will learn how an update operator changes a variable in one step, and how the editor helps you write and fix code.

Slide 3. A scoreboard keeps a running score.

- **Say:** At a skate park contest, a scoreboard adds points after each trick.
- **Say:** Read Darby's program on your bell ringer and answer both questions.
- Walk around and look for students who forget the 2 points the rider starts with.

Slide 4. Each pass adds 3 to `score`.

- **Say:** Here's the bell ringer's program. score starts at 2.
- Point to the box. **Say:** Each pass adds 3. After the first pass, score holds 5. Then it holds 8, and then 11.
- **Say:** The plus-equals sign is an update operator. It changes a variable using its old value. Today you'll meet five more.

Slide 5. An update operator changes a variable in one step.

- **Say:** Each line uses the old value of coins and stores a new one.
- Point to line 2. **Say:** Slash-slash-equals divides and rounds down to a whole number. 20 divided by 3 is 6 and some left over, so coins holds 6.
- Point to lines 3 to 5. **Say:** Times-equals multiplies, so 6 becomes 24. Percent-equals keeps the remainder, so 24 becomes 4. Star-star-equals raises to a power, so 4 squared is 16.
- Point to line 6. **Say:** Slash-equals always gives a decimal. 16 divided by 2 is 8.0, not 8.

Slide 6. Here are three words for today.

- **Say:** An update operator changes a variable using its old value. It saves you from typing the name twice.
- **Say:** A unary minus is a minus sign in front of one value, like negative 2. It flips a number's sign. Positive turns negative, and negative turns positive.
- **Say:** Most math in Python works left to right. Two stars work right to left. You'll see why that matters next.

Slide 7. ** goes first, and it works right to left.

- Point to the left. **Say:** Two stars work right to left. 3 squared is 9 comes first, and then 2 to the 9th power is 512.
- **Say:** If Python went left to right, you'd get 8 squared, which is 64. That's a common mistake.
- Point to the right. **Say:** The power goes before the minus sign in front. 2 squared is 4, and then the minus flips the sign, so it's negative 4.
- **Ask:** How could you get positive 4? Take answers. Put parentheses around negative 2.

Slide 8. Track each value in Operator Relay.

- Hand out the Operator Relay sheet, 1 per student. Computers stay closed.
- **Say:** Partner A says the value after line 2, partner B after line 3, and so on. If you don't agree, work it out together.
- After 7 minutes, have partners compare their answers for 3 minutes. Then go over the answers with the class, for about 5 minutes.

Slide 9. The Error panel names the line to fix.

- **Say:** Python ran lines 1 and 2. Then line 3 asked for scor, a name that has no value, so the program stopped with a NameError.
- **Say:** The Error panel says which line to look at. The line numbers at the left of the editor help you find it fast.
- **Say:** As you type a name, the editor's suggestion list shows names your code already has. Pick score from the list, and you can't misspell it.

Slide 10. The editor helps you write code.

- **Say:** Line 2 starts with a hash mark, so Python skips it. laps stays 0, and the program prints 8.
- **Say:** To comment out a line, type a hash mark in front of it. Delete the hash mark to turn the line back on. On a keyboard, Ctrl and slash does it too, or Command and slash on a Mac.
- **Say:** When you press Enter after the colon on line 3, the editor starts the next line 4 spaces in. That's auto-indent.
- **Say:** As you type a name, the suggestion list opens. Press Tab or Enter to pick a word, or tap it on a tablet. Esc closes the list. You'll use all three tools in the try-it.

Slide 11. Practice your code in 1.1 Try It.

- **Say:** Open a browser and type the link on the screen. It opens 1.1 Try It in CodeGrove.
- **Say:** Write the code for each task, and press Check until it works. This is practice, so nothing is turned in.
- **Say:** When I say time is up, close your computer. The exit ticket is on paper.

Slide 12. Big idea: An update operator changes a variable in one step.

- **Say:** Here's today's big idea. An update operator uses a variable's old value to store a new one, in one step.
- Hand out the exit ticket.

Day 2 — Fix and Finish with Editor Tools

Slide 13. Day 2 divider: Fix and Finish with Editor Tools

- **Say:** Today you will learn how to use update operators and the editor's tools to fix and finish short programs.

Slide 14. A snack stand packs bars into bags.

- **Say:** A snack stand packs some snacks into bags before the rush.
- **Say:** Read Leilani's program on your bell ringer and answer both questions.
- Walk around and look for students who use 3 for the bags, which is the leftover count.

Slide 15. `//` counts full bags, and `%=` keeps the remainder.

- **Say:** Here's the bell ringer's program. Line 2 counts full bags. 11 divided by 4 is 2, rounded down to a whole number.
- **Say:** Line 3 keeps the remainder. 2 bags of 4 hold 8 bars, so 3 are left, and bars now holds 3.
- **Ask:** What would print if lines 2 and 3 traded places? Take answers. Line 3 would change bars to 3 first, so line 2 would count 0 bags.

Slide 16. A missing indent stops the program.

- **Say:** Line 2 ends in a colon, so the line under it must start 4 spaces in. Line 3 starts at the left edge, so Python stops with an IndentationError.
- **Say:** Python checks the whole program before it runs a line, so nothing printed, not even the first number.
- **Say:** Auto-indent helps here. Press Enter after a colon, and the editor starts the next line 4 spaces in. Tab adds 4 spaces, and Shift and Tab take 4 away.

Slide 17. Comment out a test line, and turn it back on.

- **Say:** A test line prints a value so you can see it while you work. You don't want it in the finished program.
- **Say:** Type a hash mark in front of the line. Python skips it, and the line waits for you. On a keyboard, Ctrl and slash does the same, or Command and slash on a Mac. On a tablet, type the hash mark.
- **Ask:** Why turn a test line off instead of deleting it? Have partners agree on one reason, then take two answers.
- **Say:** You can turn it back on later, when you need to test again.

Slide 18. Fix and finish two programs in Warm-Up Arcade.

- Hand out the Warm-Up Arcade lab sheet, 1 per student.
- **Say:** Check shows if your code works, and you may use up to three hints. Your sheet is what you turn in.
- **Say:** Use today's tools. Read the line number in the Error panel, and type a hash mark to turn off a test line.

Slide 19. Big idea: The editor's tools help you fix code fast.

- **Say:** Here's today's big idea. Read the line the Error panel names, and turn off test lines with a hash mark when you finish.
- Hand out the exit ticket.

Day 3 — Find the Bonus Bug

Slide 20. Day 3 divider: Find the Bonus Bug

- **Say:** Today you will review update operators and the editor's tools, fix a bug that gives a wrong answer and take the lesson quiz.

Slide 21. Each update uses the old value of `laps`.

- **Say:** Here's the bell ringer's program. laps starts at 10.
- Point to the box. **Say:** Line 2 divides by 3 and rounds down to a whole number, so laps holds 3. Line 3 squares it, so laps holds 9.
- **Say:** Line 2 does the same job as laps equals laps floor-divided by 3. The update operator saves you from typing the name twice.

Slide 22. What does this program print?

- **Say:** This line has two powers in a row.
- **Ask:** What does it print? Have students vote with 1, 2 or 3 fingers.
- **Say:** Hold on to your vote. The next slide works it out.

Slide 23. ** works right to left, from the right end.

- **Say:** Option 2 is right. Two stars work right to left, so 2 to the 3rd power comes first. That's 8.
- **Say:** Then 2 to the 8th power is 256.
- **Say:** Option 1 works left to right, which is the common mistake. 2 squared is 4, and 4 cubed is 64.

Slide 24. A SyntaxError names the line Python can't read.

- **Say:** Python checks the whole program before it runs a line. Line 2 has no colon at its end, so Python can't read it.
- **Say:** The Error panel names line 2. Add a colon there, and the program runs.
- **Say:** Some bugs don't stop the program. A logic error gives a wrong answer with no message, so compare the output with what it should be.

Slide 25. Fix the bug in 1.1 Quiz Task.

- Hand out the 1.1 Quiz Task sheet, 1 per student. Students work alone.
- **Say:** You have 10 minutes. Then computers close for the quiz.
- Collect the sheets before the quiz.

Slide 26. Big idea: Update operators change a value in one step.

- **Say:** Here's the big idea of this lesson. An update operator uses a variable's old value, and the editor's tools help you find and fix bugs.
- Hand out the exit ticket.