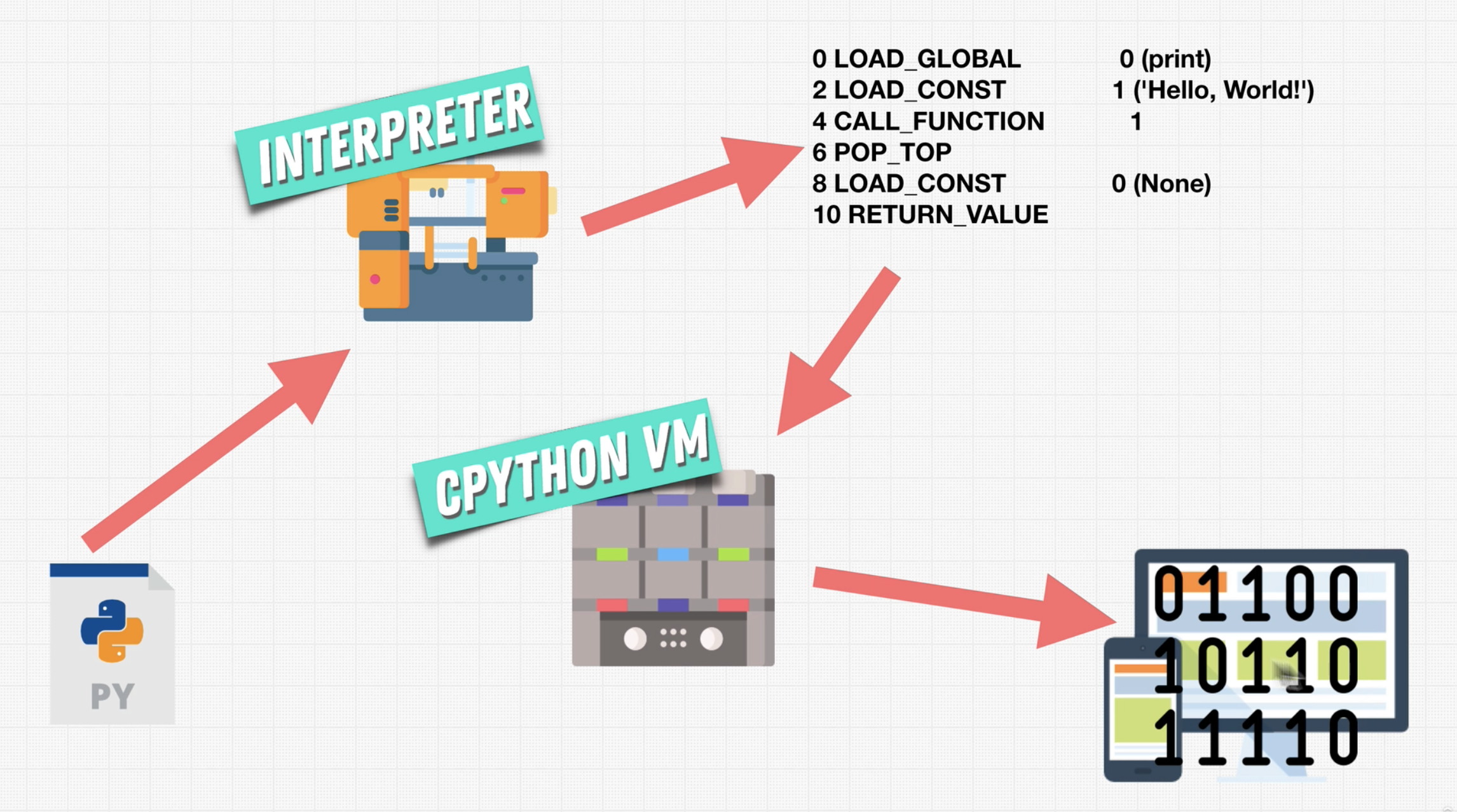
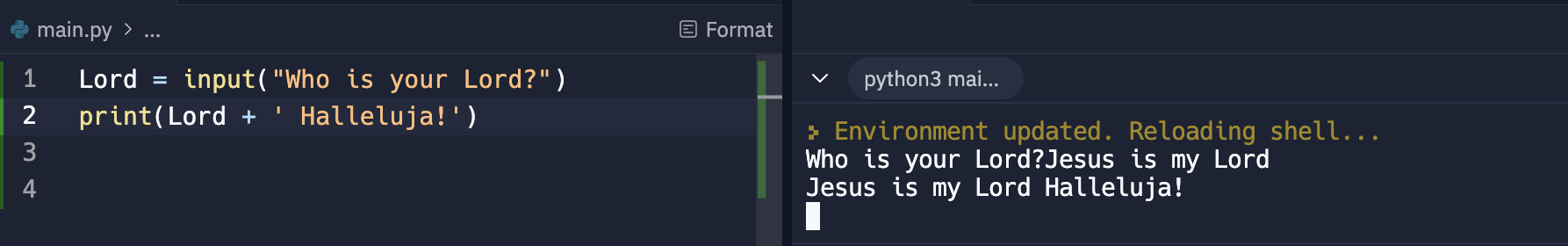
**Phyton Zero to Mastery**

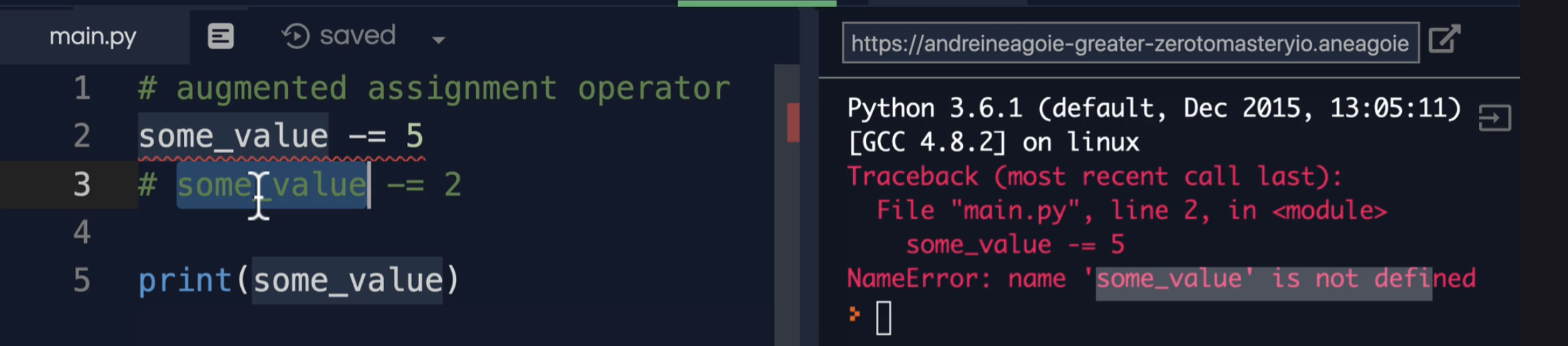
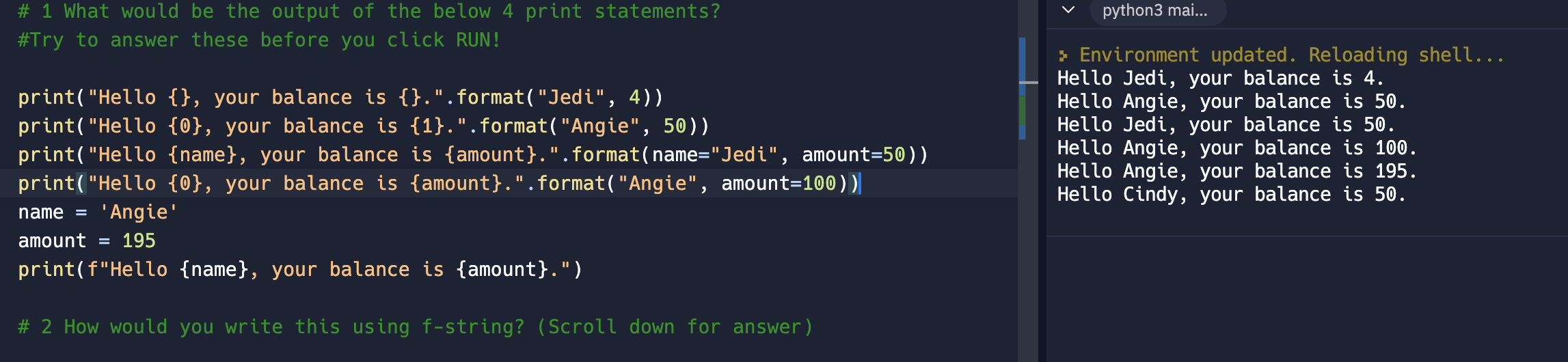
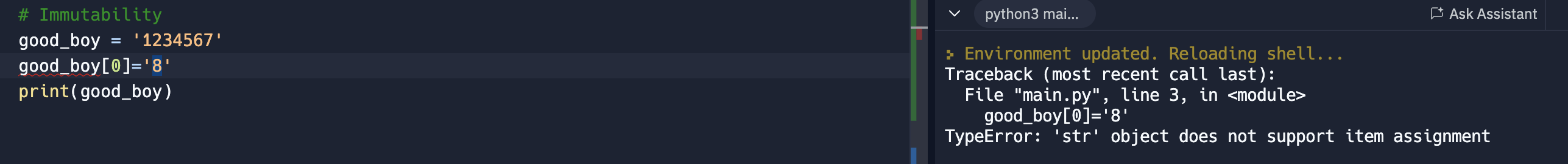
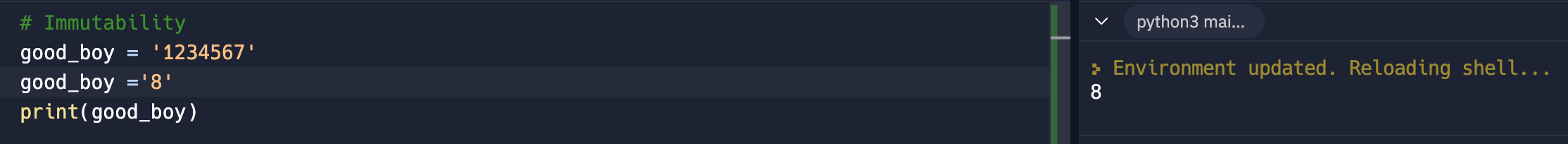
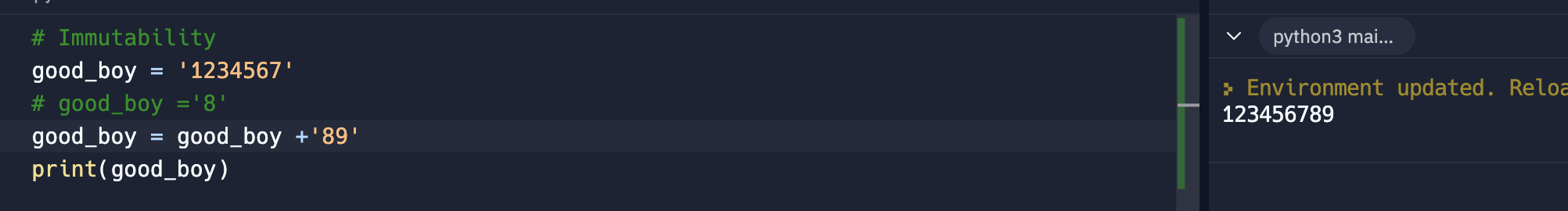
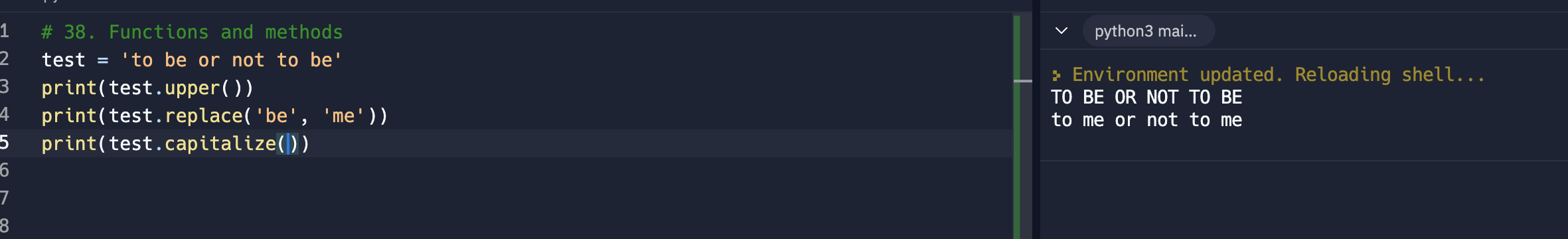
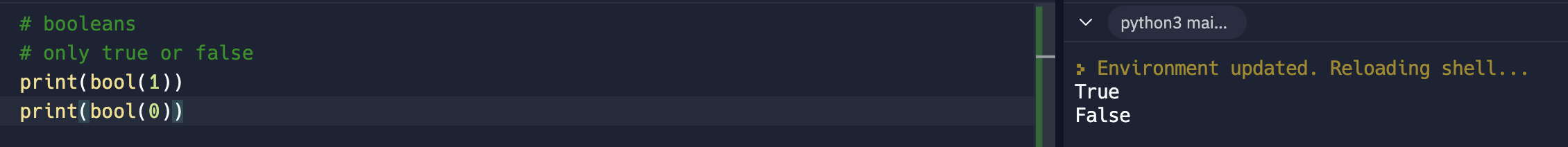
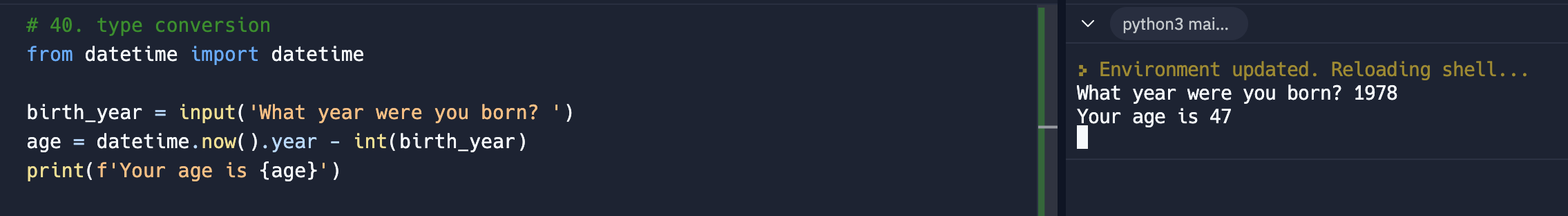
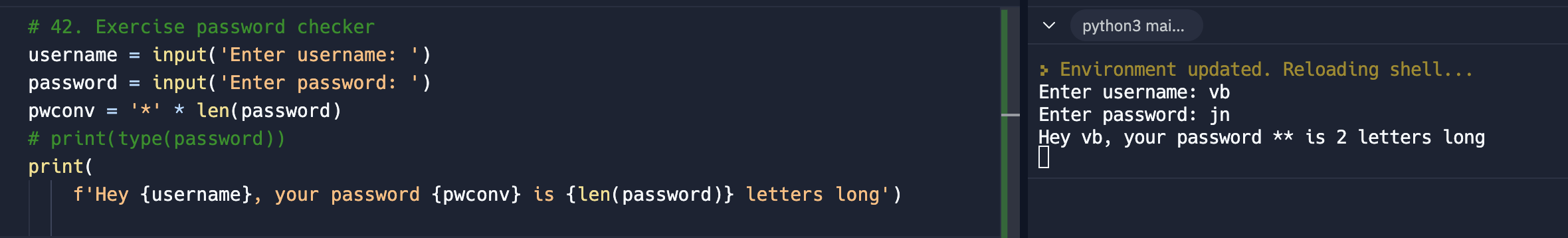
**Section 1: Intorduction**

1. Course Outline
2. Join online classroom
3. Meet your classmates and instructor
4. ZTM Resources
   1. <https://zerotomastery.io/>
   2. <https://www.youtube.com/@ZeroToMastery>
   3. <https://www.linkedin.com/groups/12121940/>

**Section 2: Phyton Introduction**

1. What is a programming language
2. Phyton Interpreter
   1. <https://www.python.org/>
3. How to run Phyton Code
   1. Basic computer setup requires
      1. Terminal
      2. Code editors (sublime or Visual Studio Code)
      3. IDEs (Spyder, Pycharm)
      4. Notebook (Jupiter)
   2. <https://replit.com/> - with pay
      1. Create an account
      2. Press Create App, Choose a template (phyton) and name the app, press create app
   3. <https://glot.io/> - free
4. Latest Version of Phyton (3)
5. Jumping to developer environment
6. 10. First Phyton Program
   1. 
   2. First sample code
      1. 
      2. Input is a function that accepts entry and was stored to the memory. Then the variable was printed
7. Monthly coding challenge
8. Phyton 2 vs Phyton 3
   1. The sotyr of Phyton - <https://www.youtube.com/watch?v=J0Aq44Pze-w>
   2. Phyton 2 vs Phyton 3 - <https://sebastianraschka.com/Articles/2014_python_2_3_key_diff.html>
   3. Phyton 2 vs Phyton 3 - <https://www.geeksforgeeks.org/important-differences-between-python-2-x-and-python-3-x-with-examples/>
9. Why so many languages
   1. List of languages - <https://en.wikipedia.org/wiki/List_of_programming_languages>
10. How Phyton works – Phyton is a high level programming language that boast for its ease of learning, a great tool for machine learning, data analyzation, web development etc. It starts with a .py file or the main.py as a source code, and when it runs, the code is pass to an interpreter (like replit.org). The interpreter converts the code to binary and process by the C Phyton Vm and splits the output to machine
11. Phyton cheat sheet –
    1. /Users/sherwinnofuente/Desktop/Python_Cheatsheet_Zero_To_Mastery_V105.pdf
    2. Finally, a fellow student has made this for all code used in this course! This way you can always check your code against it if you are getting any errors as you go through the course: <https://github.com/aneagoie/All-Python-codes-of-ZTM-course-by-Andrei-Neagoie>
12. Phyton Developer Monthly
    1. Every month, I accumulate all of the best resources and articles, as well as free resources around the web for Python Developers. If you want to stay up to date with the industry and make sure you don't miss any important news, you can check out the monthly newsletter here. [**It's completely free every month**](https://zerotomastery.io/newsletters/python-monthly/)!
    2. Don't worry if the topics in the newsletter are still difficult. By the end of this course, you will have all the skills necessary to read everything and understand them :)
    3. **AI & Machine Learning Monthly / Web Developer Monthly:** There is also a Machine Learning Monthly (written by Daniel) and Web Developer Monthly newsletter if you are interested. You can read past issues [**here**](https://zerotomastery.io/newsletters/)**.**
13. Endorsement on LinkedIn
    1. If you are looking to improve your LinkedIn profile and have others endorse your skills, we have a private ZTM[LinkedIn group here](https://www.linkedin.com/groups/12121940/). LinkedIn allows you to have recruiters message you with lots of job opportunities. You can join the group by clicking on "LinkedIn Group" and then go ahead and endorse some of the member's skills (other people will do the same for you as they join).  
         
       If you have any questions, reach out in our private Discord chat community in the #job-hunting channel!  
         
       UPDATE!!! Zero to Mastery is officially a recognized school! What does this mean for you? It means that you can add it as an educational institution on LinkedIn as part of your profile to wow those employers (as your education history). [Check it out here](https://www.linkedin.com/school/64685953/). To add it to your profile:
    2. Step 1: Go to personal LinkedIn profile
    3. Step 2: Scroll down to the Education section
    4. Step 3: Click the +
    5. Step 4: Type in Zero To Mastery Academy
    6. Course content
    7. AI Assistant

**Section 3: Phyton Basics**

1. Learning Phyton
   1. 4 key things to master
      1. Terms – definitions
      2. Data Types – what sort of data a language can hold
      3. Actions –
      4. Best Practices
2. Phyton Data Types - values in Phyton
   1. Fundamental Data Types
      1. Int – numbers
      2. Float
      3. Bool – Boolean
      4. Str – string
      5. Tuple
      6. Set
      7. Dict
   2. Classes – custom types
   3. Specialized data types – not built in to Phytons. Normally libraries or extensions
   4. None – simply nothing
3. How to succeed
   1. In order to get the most out of the course, I encourage you to code along with me throughout the next sections. Watching videos may help, but nothing beats hands on practice and actually coding to develop your skills and make you comfortable with Python. Let's get started!
4. Numbers
   1. Int – whole number
   2. Float – number with decimal point; <https://www.youtube.com/watch?v=PZRI1IfStY0>
   3. Double multiply ( \*\* ) - power. Example 2 \*\* 2 = 4; 2^2=4
   4. Double divide ( // ); round down; 2 // 4 = 0.
   5. Modulo ( % ); the remainder; 5 % 4 = 1
5. Math functions - <https://www.programiz.com/python-programming/modules/math>
   1. Round(number, ndigits)
   2. Abs – absolute; no negative
6. Developer Fundamentals I
   1. How to avoid mistakes
   2. Key things to make efficient in learning and becoming a good programmer
   3. Do not read the dictionary – learn only what is needed; little by little
   4. Be good in googling
   5. Understand what exist and how to use
7. Operator Precedence
   1. () – bracket first
   2. \*\* - power
   3. Multiply or divide
   4. Addition or subtraction
8. Repit code –
   1. In the next lesson you will start seeing links to my Replit profile which will have code from the videos you will watch throughout the course. Sometimes, Replit servers can have issues/or be down, so please take note of 2 important things you can do to make sure you can follow the course without interruptions:  
        
      1. You can always find the Replit code on [my github repo specific to this course right here](https://github.com/aneagoie/ztm-python-course-exercises) so you have it in one place throughout the course.  
        
      2. When you click on the link of a Replit project (like in the next lecture), you will see something similar to below. Make sure you CLICK ON THE "SHOW CODE" icon so you can see my code:
      1. 
9. Exercise: Operator Precedence
10. Optional: bin() and complex - <https://www.geeksforgeeks.org/python-int-function/>
11. Variables - <https://www.w3schools.com/python/python_ref_keywords.asp>
    1. Stores information
    2. Snake\_case – space is underscore or private variable
    3. Start with lower case or underscore
    4. Letters, numbers, underscore
    5. Case sensitive
    6. Don’t overwrite words
    7. Do not create variable with \_\_
12. Expression vs Statements
    1. Example: user\_age = 2 /4
    2. Expression – 2 / 4
    3. Statement – user\_age = 2 / 4
13. Augmented Assignment Operator
    1. 
14. Strings – text. Expressed by double or single quotes
15. String Concatenation – joining text
16. Type conversion – print(type(str(100))) – will result to 100 as string and not int
17. Escape Sequences – this is to address the hyphen in strings
18. Formatted Strings
    1. 
    2. .format is an old way of formatting strings.
    3. f at the beginning of the expression is the new way of formatting strings
19. String indexes – put each character in an array
    1. Sample is good\_boy = 1234567
    2. [start, stop, step over] – is string slicing
    3. [0] – 1
    4. [0:3] – 123;
    5. [1:] – up to the end; 234567
    6. [::2] – 1357; step over from the first character and jump or step over every two characters
    7. [-7] – 7; its like RIGHT in excel
    8. [::-1] -7654321; reverse the counting. Starts at the beginning, end at the end but the counting starts from the end to start
    9. 
20. Immutability
    1. Strings are immutable or can’t just be changed. The variable needs to re-assign the value
    2. Changing the value of an existing variable from the memory. See the error below
    3. 
    4. 
    5. 
    6. Now the value of good\_boy = ‘8’ and the old goo\_boy = ‘1234567’ doesn’t exist anymore
21. Built in functions and + Methods
    1. Phyton Built in functions - <https://docs.python.org/3/library/functions.html>
    2. Phyton string methods - https://www.w3schools.com/python/python\_ref\_string.asp
    3. Len() – calculate the length of the string
    4. Methods are like functions but owned by something. Example, print() is a function, while count() is a method of that only strings can perform
       1. Function use parenthesis like print()
       2. Methods use . and parenthesis to initiate like .format()
    5. 
22. Booleans
    1. Booleans can only be true or false
    2. 1 – means True
    3. 0 – means False
    4. 
23. Exercise: Type conversion
    1. At this stage int, float, str and bool were discusses
    2. Below is an exercise to define the age after a question of what year you were born was asked
    3. 
24. Developer Fundamentals II
    1. Commenting best practices - <https://realpython.com/python-comments-guide/>
    2. Be concise in making comments
25. Exercise: Password Checker
    1. 
26. Lists
    1. Like array in JS
    2. A data structure
    3. Li = [1, ‘yes’. True, 2.5]
27. List slicing