What Works

In Teaching Python

python™

IN EDUCATION

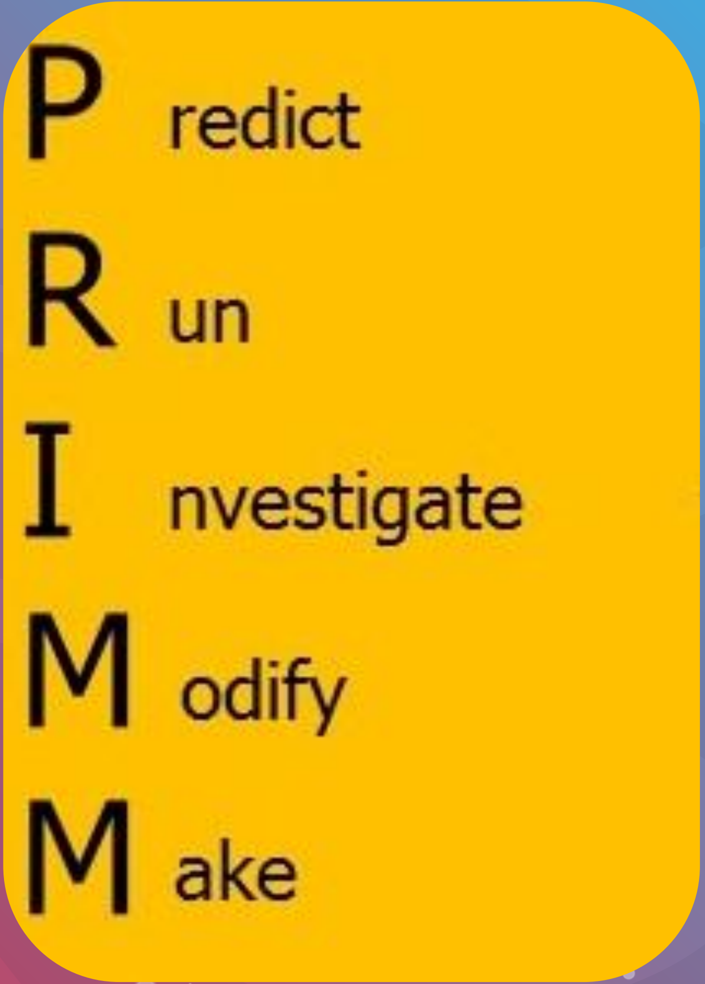
Toolkit Contents

- PRIMM
- Concept Before Code
- Worked Examples with Subgoal Labelling
- Project-based Learning
- Tell a Story with the Code

PRIMM

PRIMM is a structure that you can use to plan programming lessons. It is useful for programmers at different experience levels who are learning a new concept.

Learn More



P redict

R un

I nvestigate

M odify

M ake

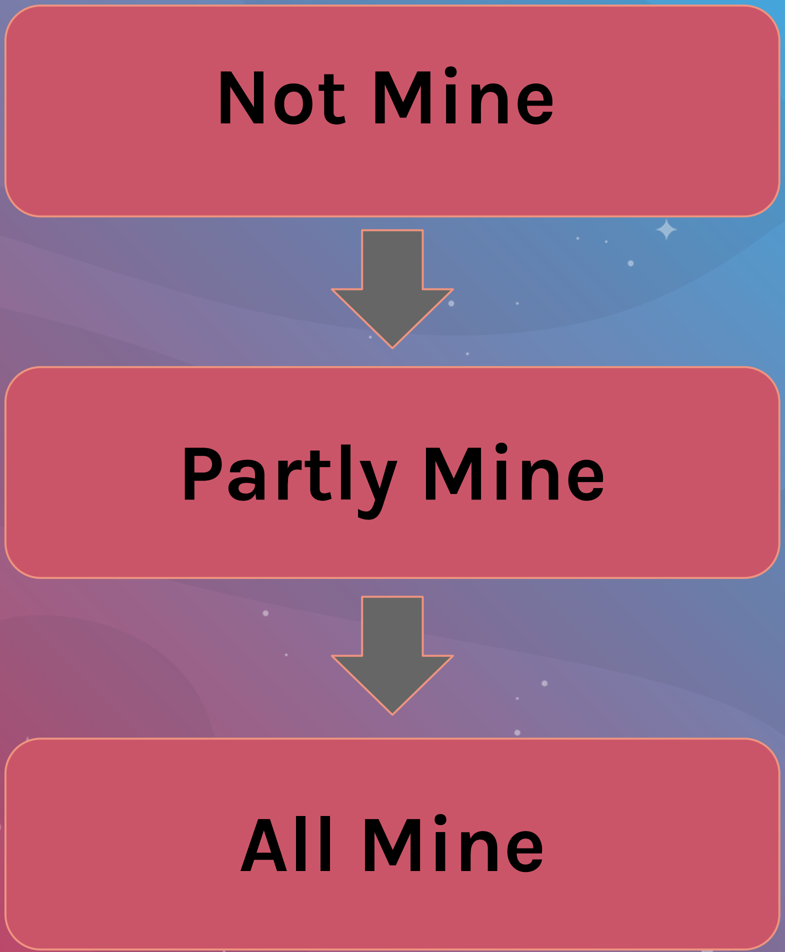
PRIMM

Example:

Replit

Intro to Python Teaching Curriculum

Created by Andrew Colley (@mracolley)

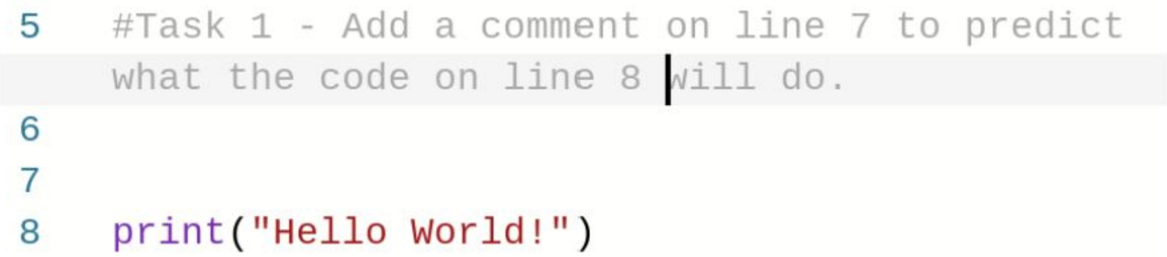


Not Mine

Partly Mine

All Mine

Programming – Output – Predict & Run



```
5   #Task 1 - Add a comment on line 7 to predict
    what the code on line 8 will do.
6
7
8   print("Hello World!")
```

Add comments to the code to predict exactly what it will output.

Run the code to see if you were correct.

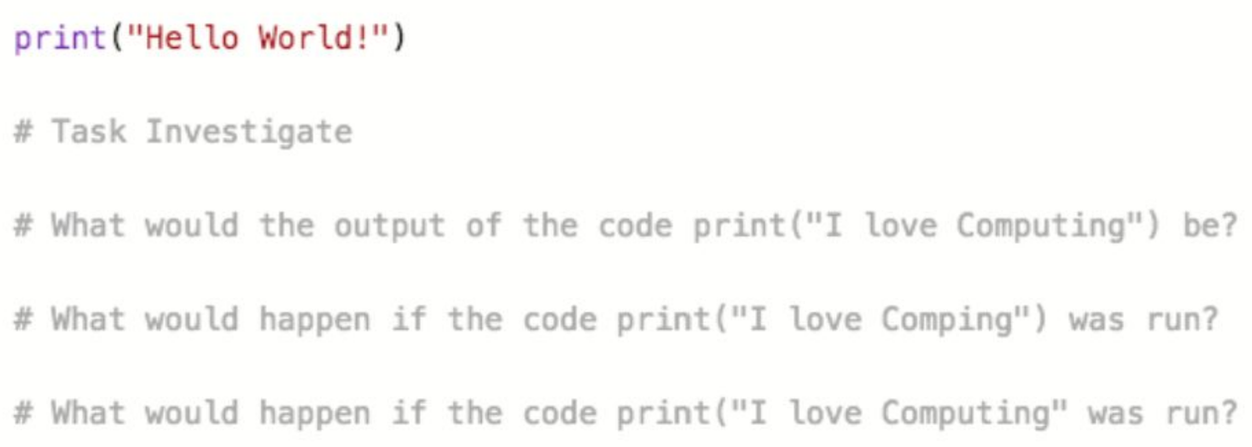
repl.it

https://repl.it/@MrAColley/11-Output

python IN EDUCATION

Resources created by Andy Colley (@MrAColley)

Programming – Output – Investigate



```
print("Hello World!")

# Task Investigate

# What would the output of the code print("I love Computing") be?

# What would happen if the code print("I love Comping") was run?

# What would happen if the code print("I love Computing" was run?
```

Add comments to the code to answer the questions.

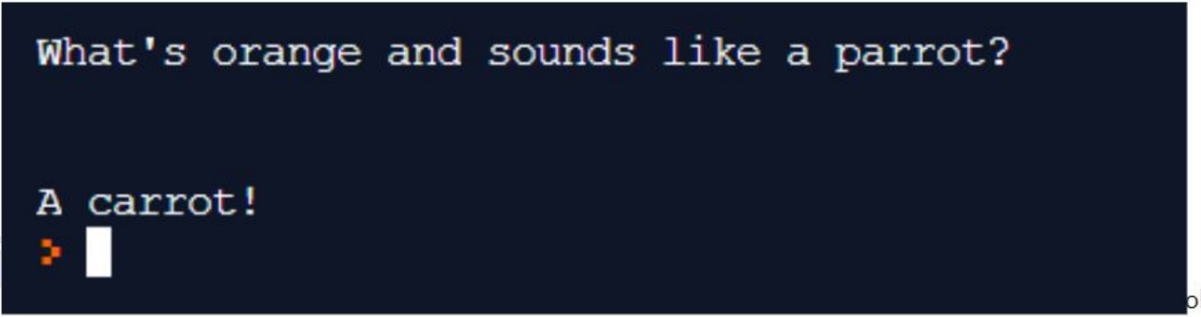
https://repl.it/@MrAColley/11-Output

Resources created by Andy Colley (@MrAColley)

Programming – Output – Modify & Make

Modify – reuse the print statement to add your own single line message to the program.

Make – use the print statement to output a joke that appears on multiple lines (keep it clean!). **Extra challenge** – can you figure out how to add blank lines like in my joke?



```
What's orange and sounds like a parrot?


A carrot!
>
```

olley (@MrAColley)

Concept Before Code

Concept Before Code is a strategy that involves exposing learners to a new programming concept through non-programming activities prior to using it in code for the first time.

Learn More

python™ IN EDUCATION

Concept Before Code

1. Define the concept.
2. Share an analogy related to learners' background knowledge.
3. Engage learners with unplugged or digital non-programming activities.
4. Show the Python code that represents the concept. Explicitly relate it to the prior activity.

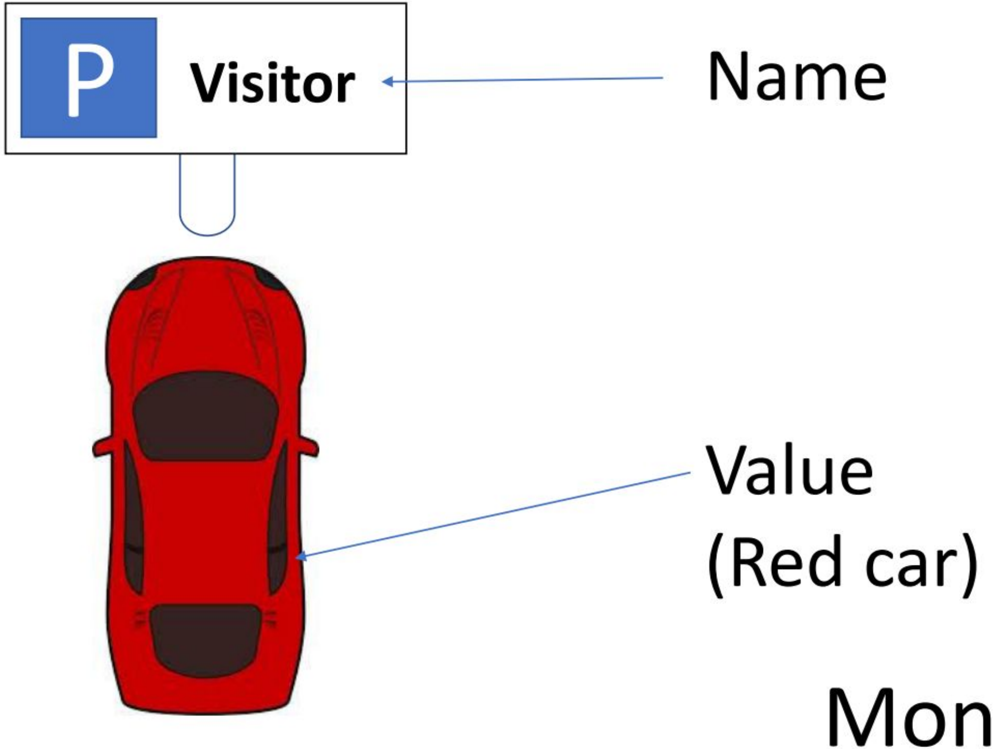
Example:

Code-IT

Supporting Algorithmic & Programmatic Thinking Resources

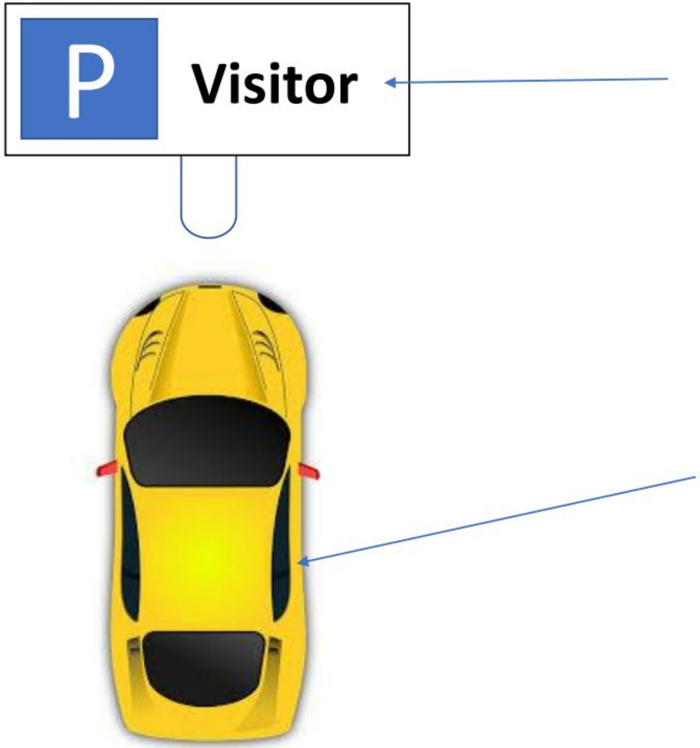
Created by Phil Bagge (@baggiepr)

Everyday variables Visitor Car Parking Space



Monday

Everyday variables Visitor Car Parking Space



Name

Value
(Yellow car)

Tuesday

Everyday variables Visitor Car Parking Space

Tell your neighbour how the value of the variable changed every day

Monday Red car

Tuesday Yellow car

Wednesday Empty

Worked Examples with Subgoal Labeling

Worked examples demonstrate programmatic solutions. They usually consist of a problem statement or goal and an example program or code snippets that represents the solution. When subgoals are labelled, the programmatic solution is broken down into discrete steps or chunked by functionality and these steps or parts are explicitly labelled.

Learn More

Worked Examples with Subgoal Labeling

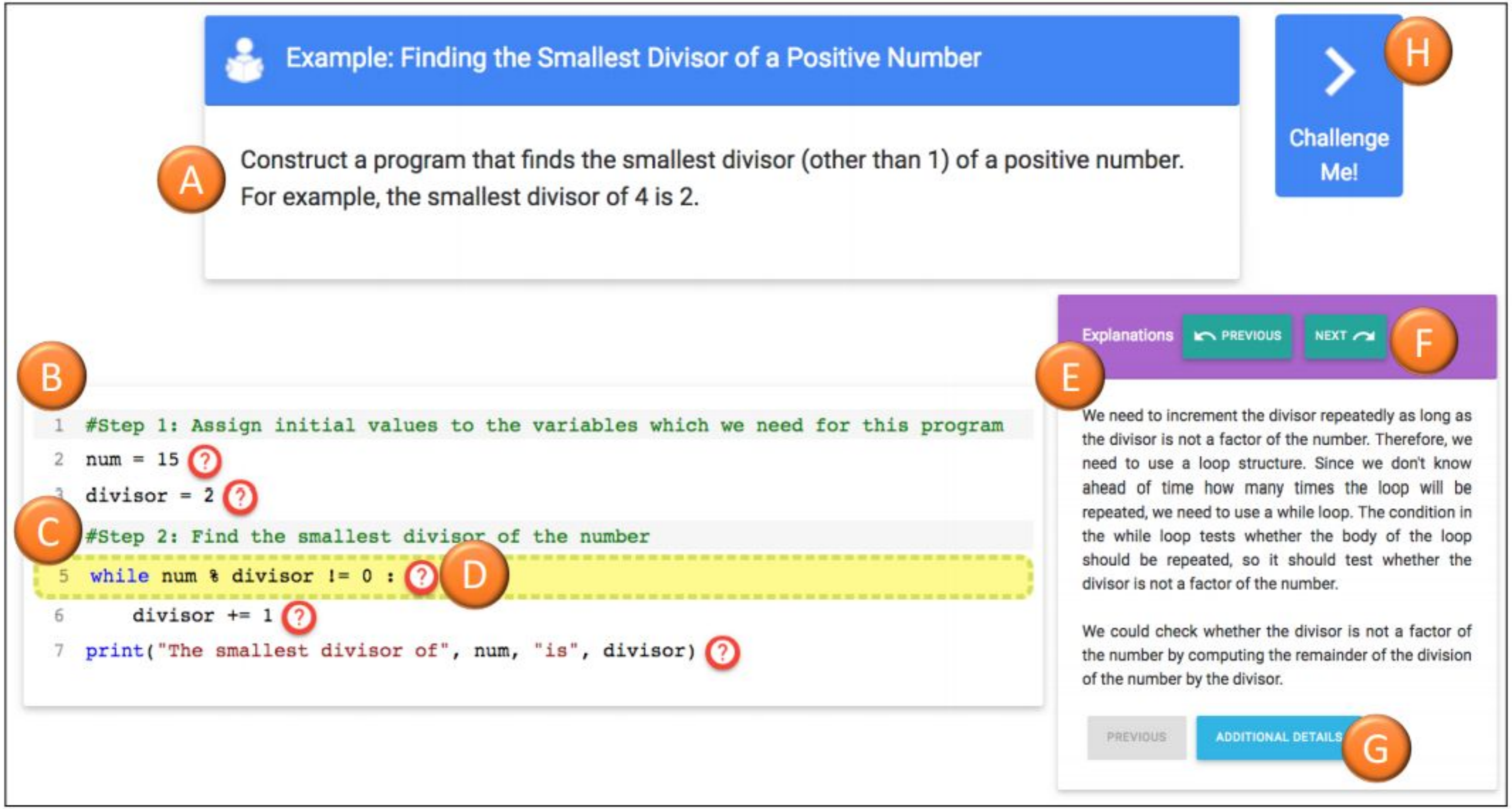
Examples:

Improving Engagement in Program Construction Examples for Learning Python Programming

By Hosseini, R., et al (2020)

Subgoals Help Students Solve Parson's Problems

By Morrison, B., Margulieux, L., Ericson, B., & Guzdial, M. (2015)



Construct a program that finds the smallest divisor (other than 1) of a positive number. For example, the smallest divisor of 4 is 2.

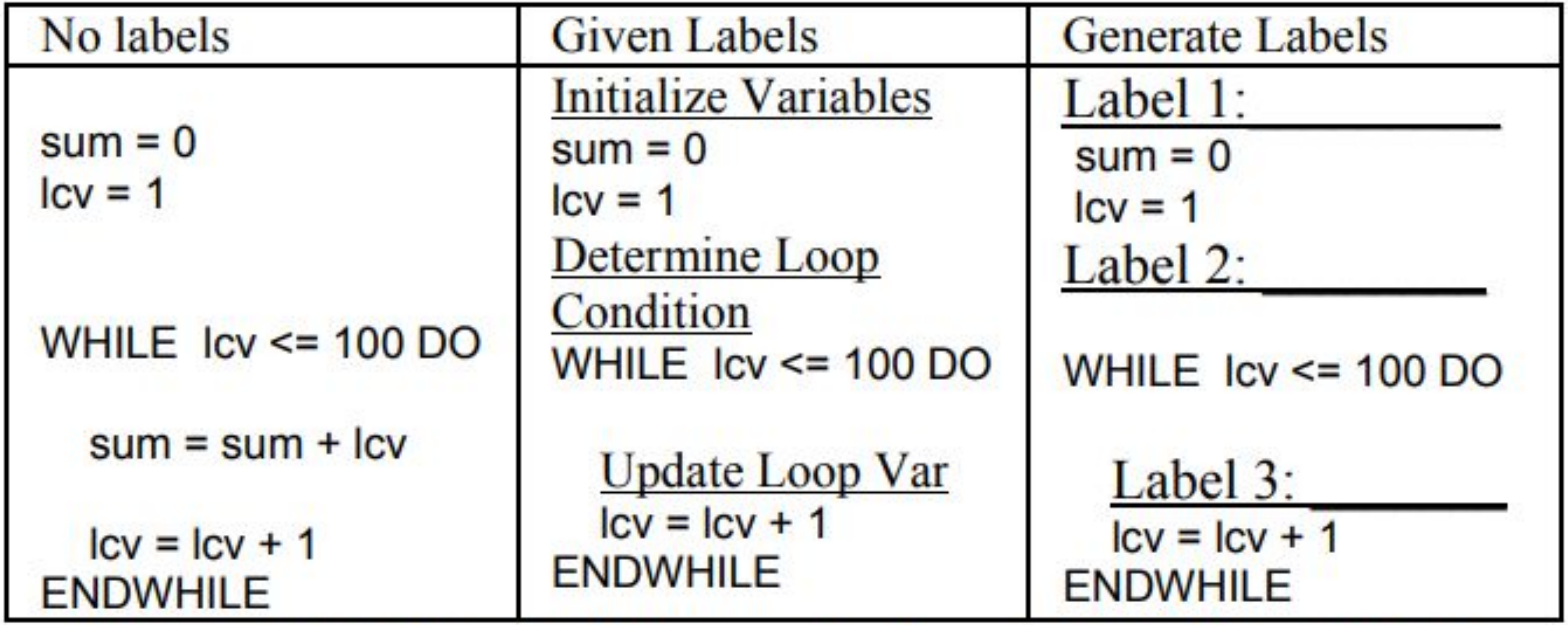
```
#Step 1: Assign initial values to the variables which we need for this program
num = 15
divisor = 2
#Step 2: Find the smallest divisor of the number
while num % divisor != 0 :
    divisor += 1
print("The smallest divisor of", num, "is", divisor)
```

We need to increment the divisor repeatedly as long as the divisor is not a factor of the number. Therefore, we need to use a loop structure. Since we don't know ahead of time how many times the loop will be repeated, we need to use a while loop. The condition in the while loop tests whether the body of the loop should be repeated, so it should test whether the divisor is not a factor of the number.

We could check whether the divisor is not a factor of the number by computing the remainder of the division of the number by the divisor.

Fig. 1 A Python programming worked example in the PCEX activity. The example includes the goal (A), interactive worked code (B), the subgoal label presented as a comment (C), the link to instructional explanations (question mark symbols) (D), explanations (E), a navigation link to the explanation for the previous/next line (F), additional details for the highlighted line (G), and a challenge navigation link (H).

Hosseini, R., et al (2020)



No labels	Given Labels	Generate Labels

sum = 0
lcv = 1
WHILE lcv <= 100 DO
sum = sum + lcv
lcv = lcv + 1
ENDWHILE
	Initialize Variables
	sum = 0
	lcv = 1
	Determine Loop Condition
	WHILE lcv <= 100 DO
	Update Loop Var
	lcv = lcv + 1
	ENDWHILE
		Label 1: ______
		sum = 0
		lcv = 1
		Label 2: ______
		WHILE lcv <= 100 DO
		Label 3: ______
		lcv = lcv + 1
		ENDWHILE

Figure 1. Partial worked example formatted with no labels, given labels, or placeholders for generated labels.

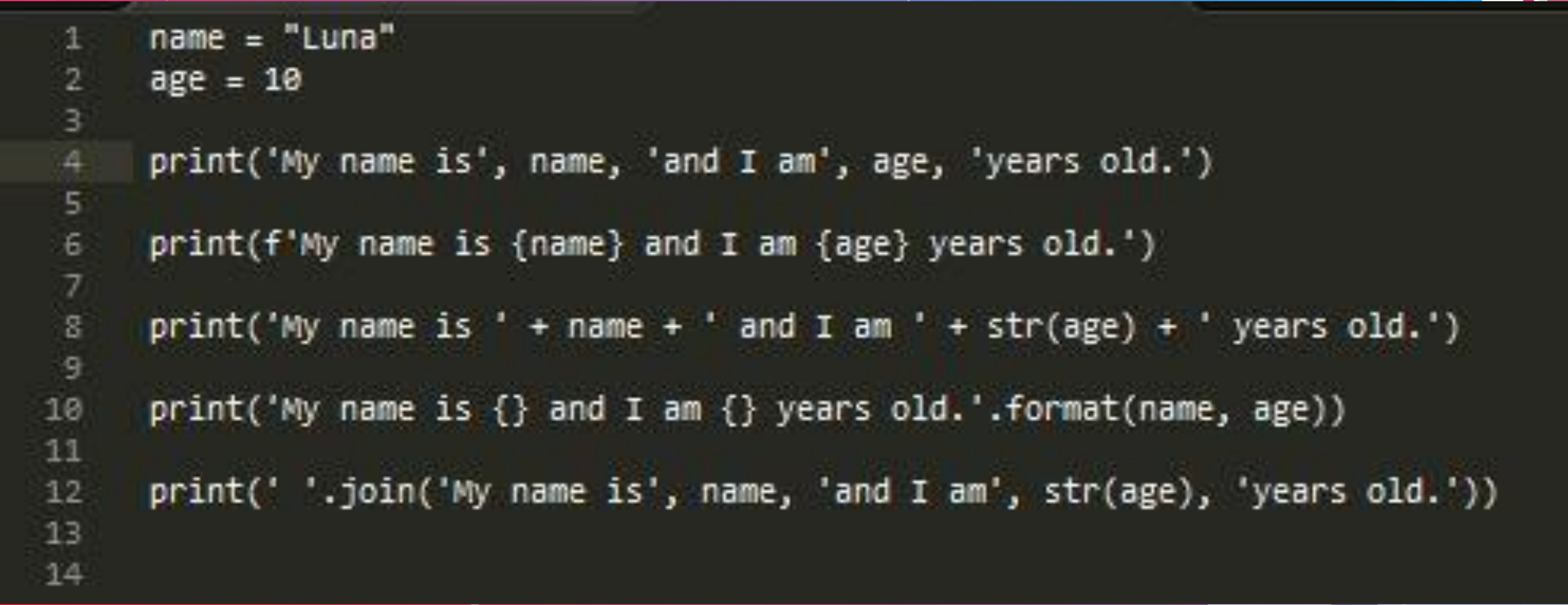
By Morrison, B., Margulieux, L., Ericson, B., & Guzdial, M. (2015)

Choose Code that Makes Processes Visible

When there are multiple ways to meet a goal, professional software developers often choose the option that is the most efficient to run and/or to write, which usually utilizes abstraction. In education, efficiency isn’t the priority. Rather making abstract concepts explicit and visible is. Let the code “tell the story.”

Learn More

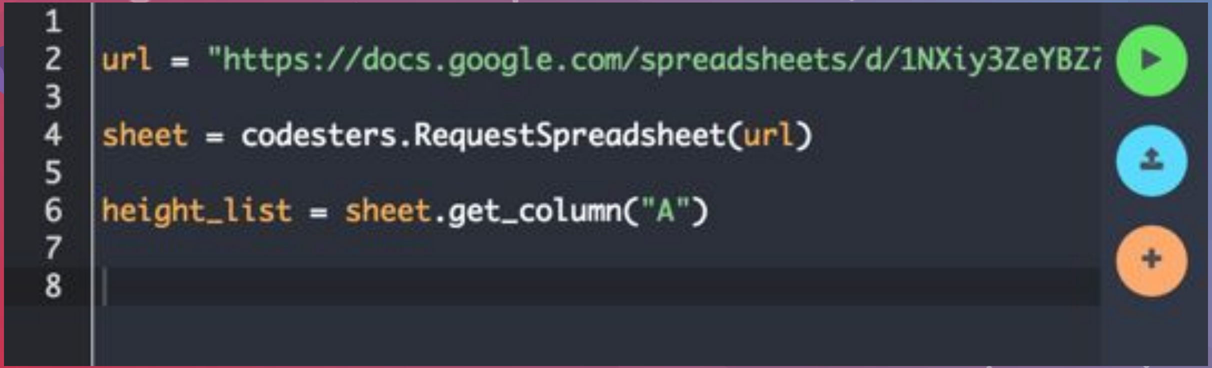
Which print statement tells the clearest story?



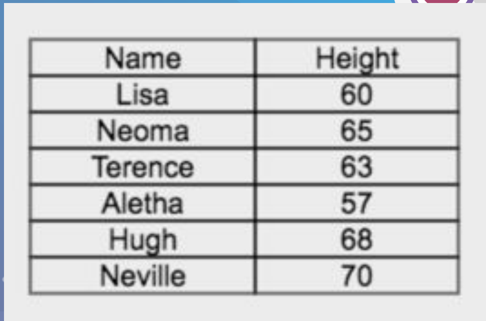
```python
name = "Luna"
age = 10

print('My name is', name, 'and I am', age, 'years old.')

print(f'My name is {name} and I am {age} years old.')

print('My name is ' + name + ' and I am ' + str(age) + ' years old.')

print('My name is {} and I am {} years old.'.format(name, age))

print(' '.join('My name is', name, 'and I am', str(age), 'years old.'))


```

From Snakelets to Pythonistas, Meg Ray (2017)

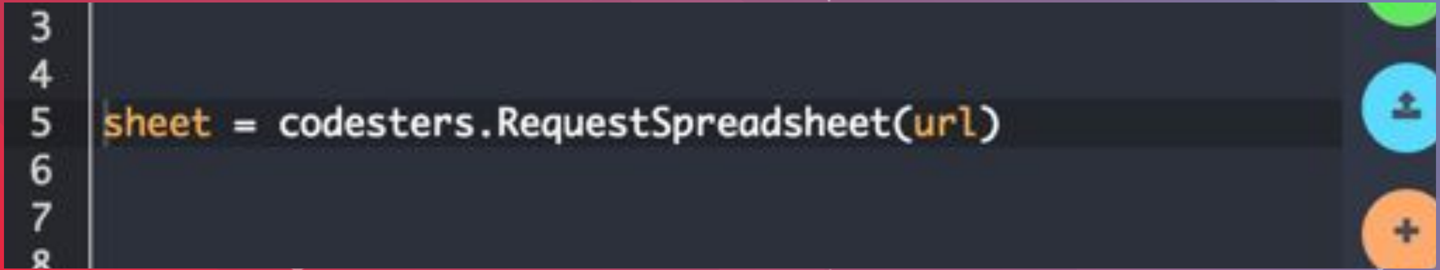
Which code snippet tells the clearest story about how the data in the table was obtained?



```
url = "https://docs.google.com/spreadsheets/d/1NXiy3ZeYBZ7

sheet = codesters.RequestSpreadsheet(url)

height_list = sheet.get_column("A")
```

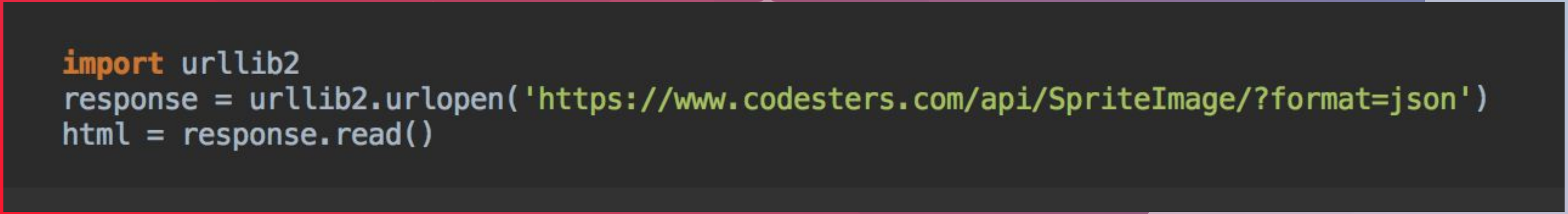


Name	Height
Lisa	60
Neoma	65
Terence	63
Aletha	57
Hugh	68
Neville	70



```
sheet = codesters.RequestSpreadsheet(url)
```

Opening the Magic Box, Farfan, Ray, & Ricard (2016)



```
import urllib2
response = urllib2.urlopen('https://www.codesters.com/api/SpriteImage/?format=json')
html = response.read()
```

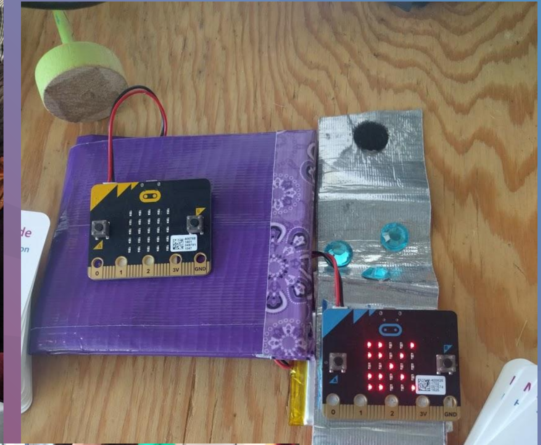
Project-Based Learning

Project-Based Learning (PBL), puts learning in the context of a project that is personally meaningful to learners. Project-building is iterative and students are assessed based on multiple aspects of the final version of their project. When projects have a tangible component to them, it is often even more effective.

Learn More - general PBL

Learn More - PBL for CS

Examples: PyGotham Young Coders, Pycon UK Education Summit, Big Red STEM Day



python™
IN EDUCATION

python™ IN EDUCATION

Do you have suggestions for strategies that you'd like to see added to this toolkit?

Send them to education@python.org

Content by Meg Ray

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, and infographics & images by Freepik

Please keep this slide for attribution