Assignment -7.3

Name: Shaik Nishaaj
2303A51613
Btach:06

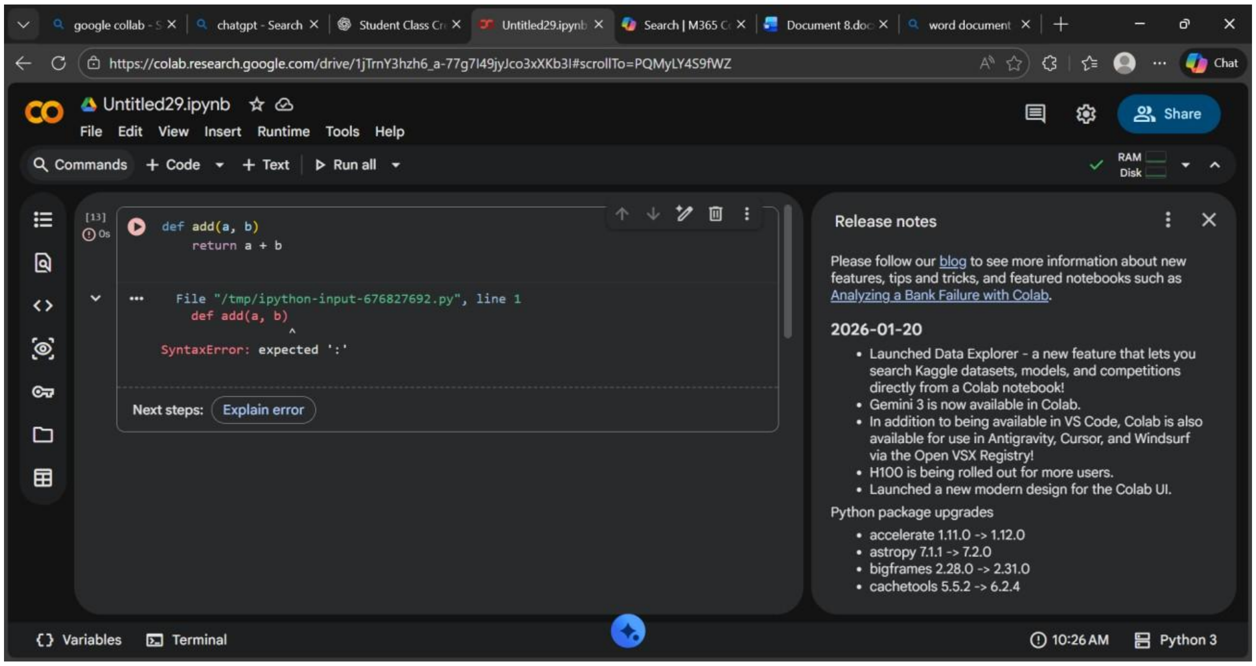
Ask 1: Fixing Syntax Errors

Prompt: The following Python function has a syntax error. Identify the issue and correct it. Also explain what the syntax error is.

def add(a, b)

return a + b Input:

Bug Code:



```
def add(a, b)
    return a + b
```

```
  File "/tmp/ipython-input-676827692.py", line 1
    def add(a, b)
                 ^
SyntaxError: expected ':'
```

Next steps: Explain error

Release notes

Please follow our blog to see more information about new features, tips and tricks, and featured notebooks such as Analyzing a Bank Failure with Colab.

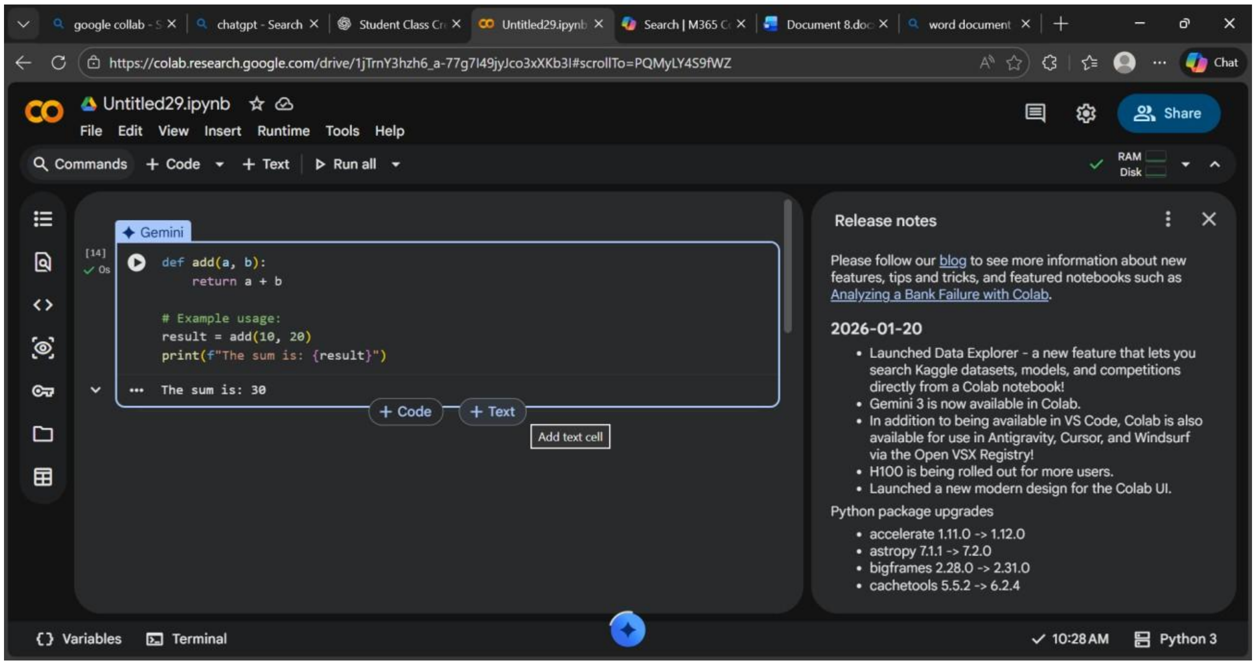
2026-01-20

- Launched Data Explorer - a new feature that lets you search Kaggle datasets, models, and competitions directly from a Colab notebook!
- Gemini 3 is now available in Colab.
- In addition to being available in VS Code, Colab is also available for use in Antigravity, Cursor, and Windsurf via the Open VSX Registry!
- H100 is being rolled out for more users.
- Launched a new modern design for the Colab UI.

Python package upgrades

- accelerate 1.11.0 -> 1.12.0
- astropy 7.1.1 -> 7.2.0
- bigframes 2.28.0 -> 2.31.0
- cachetools 5.5.2 -> 6.2.4

2) corrected code:



```
def add(a, b):
    return a + b

# Example usage:
result = add(10, 20)
print(f"The sum is: {result}")
```

```
The sum is: 30
```

Release notes

Please follow our blog to see more information about new features, tips and tricks, and featured notebooks such as Analyzing a Bank Failure with Colab.

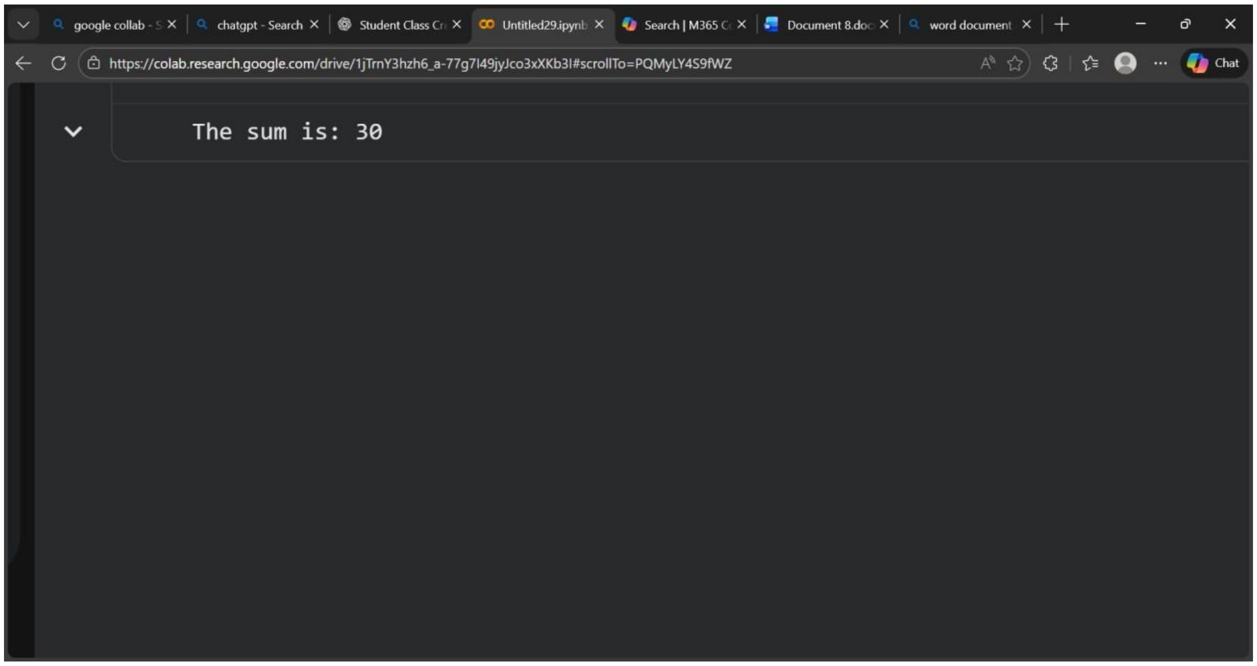
2026-01-20

- Launched Data Explorer - a new feature that lets you search Kaggle datasets, models, and competitions directly from a Colab notebook!
- Gemini 3 is now available in Colab.
- In addition to being available in VS Code, Colab is also available for use in Antigravity, Cursor, and Windsurf via the Open VSX Registry!
- H100 is being rolled out for more users.
- Launched a new modern design for the Colab UI.

Python package upgrades

- accelerate 1.11.0 -> 1.12.0
- astropy 7.1.1 -> 7.2.0
- bigframes 2.28.0 -> 2.31.0
- cachetools 5.5.2 -> 6.2.4

Output:

- 

```
The sum is: 30
```

Explanation:

- In Python, a colon **:** is required after defining a function header.
- Without the colon, Python cannot recognize the start of the function block, causing a **SyntaxError**.
- AI correctly identified the missing colon and fixed the function definition.

Task 2: Debugging Logic Errors in Loops

Prompt: The following Python loop runs infinitely. Identify the logic error, correct the loop, and explain the issue.

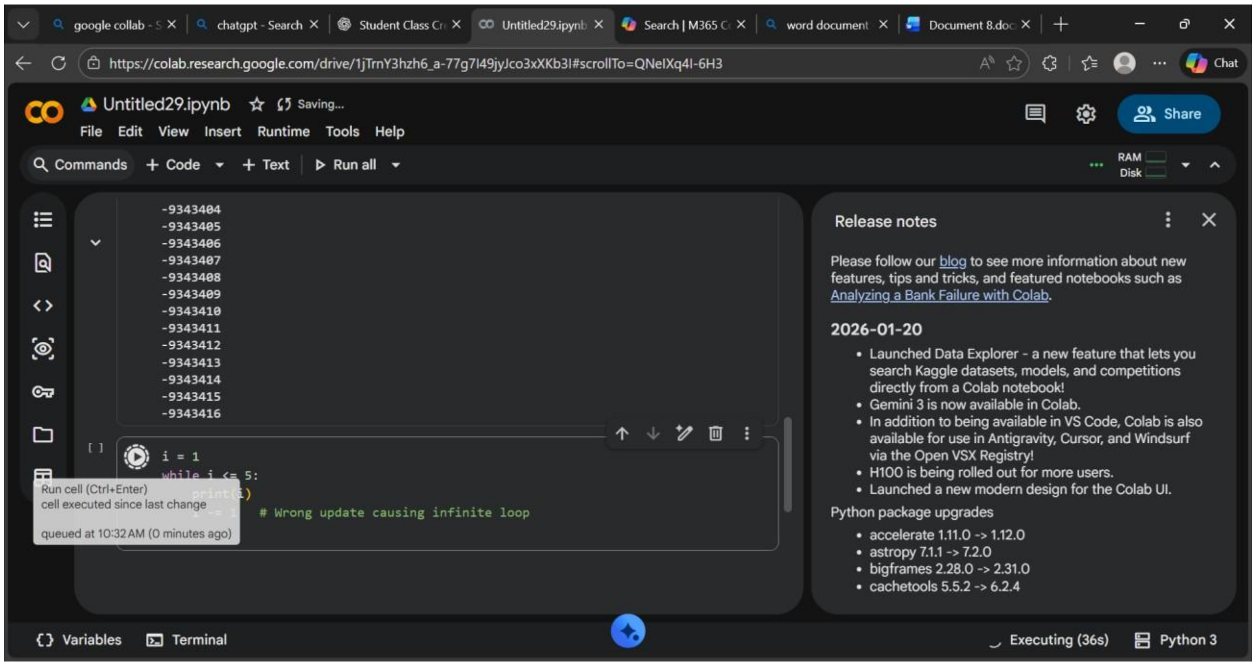
i = 1 while i

<= 5:

  print(i)

i -= 1

Input: Bug code:



```
-9343404
-9343405
-9343406
-9343407
-9343408
-9343409
-9343410
-9343411
-9343412
-9343413
-9343414
-9343415
-9343416
```

```
i = 1
while i <= 5:
    print(i)
    i -= 1    # Wrong update causing infinite loop
```

Run cell (Ctrl+Enter)
cell executed since last change

queued at 10:32 AM (0 minutes ago)

Release notes

Please follow our blog to see more information about new features, tips and tricks, and featured notebooks such as Analyzing a Bank Failure with Colab.

2026-01-20

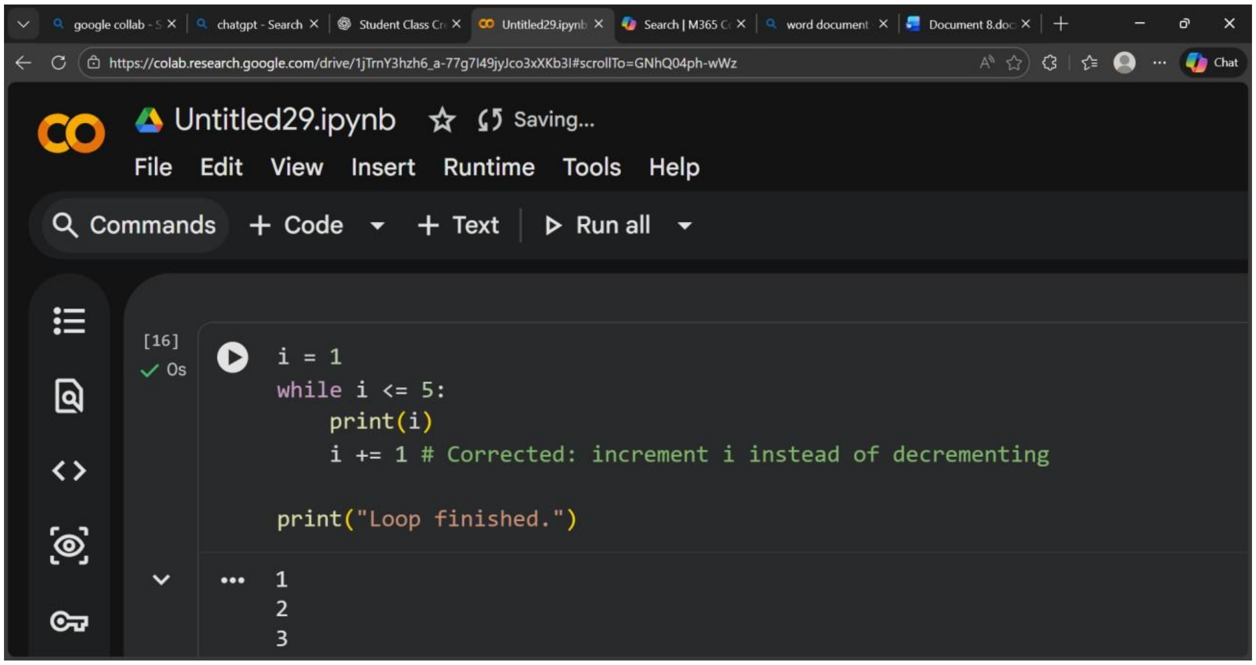
- Launched Data Explorer - a new feature that lets you search Kaggle datasets, models, and competitions directly from a Colab notebook!
- Gemini 3 is now available in Colab.
- In addition to being available in VS Code, Colab is also available for use in Antigravity, Cursor, and Windsurf via the Open VSX Registry!
- H100 is being rolled out for more users.
- Launched a new modern design for the Colab UI.

Python package upgrades

- accelerate 1.11.0 -> 1.12.0
- astropy 7.1.1 -> 7.2.0
- bigframes 2.28.0 -> 2.31.0
- cachetools 5.5.2 -> 6.2.4

https://colab.research.google.com/drive/1jTrnY3hzh6_a-77g7I49jyJco3xXKb3I#scrollTo=QNeIXq4I-6H3
Untitled29.ipynb
Saving...
File Edit View Insert Runtime Tools Help
Commands
+ Code
+ Text
Run all
Share
Variables
Terminal
Executing (36s)
Python 3

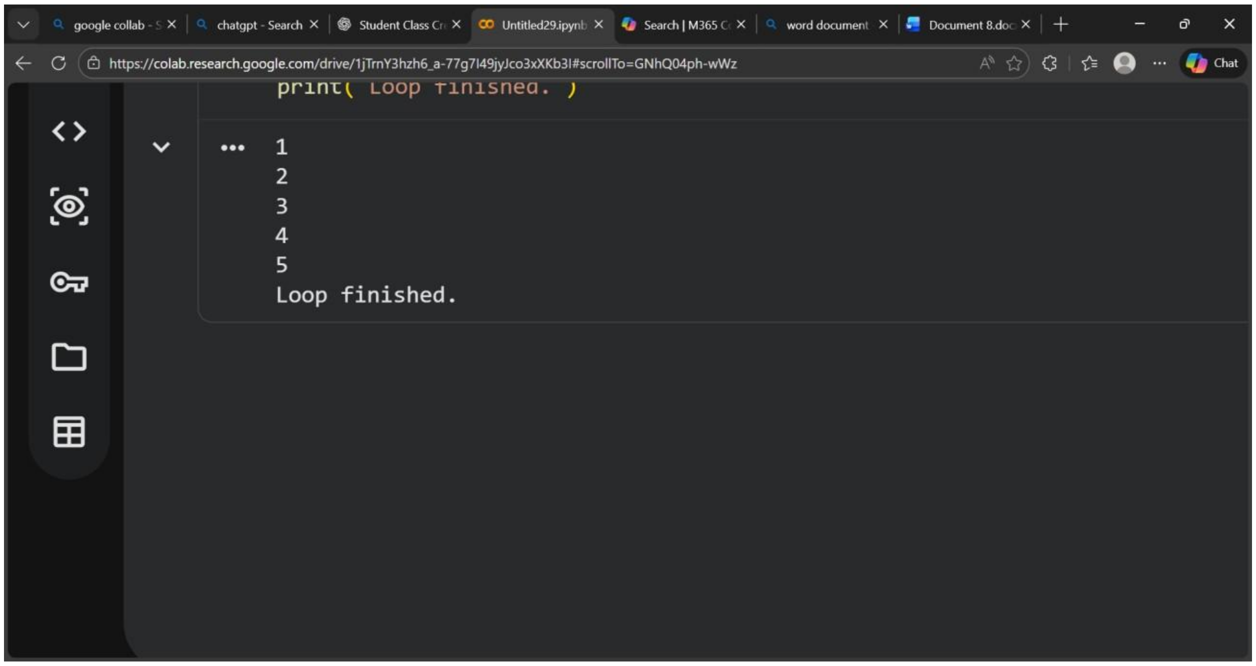
Corrected code:


Untitled29.ipynb
Saving...
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
[16]
0s

```
i = 1
while i <= 5:
    print(i)
    i += 1 # Corrected: increment i instead of decrementing

print("Loop finished.")
```

```
1
2
3
```

Output:



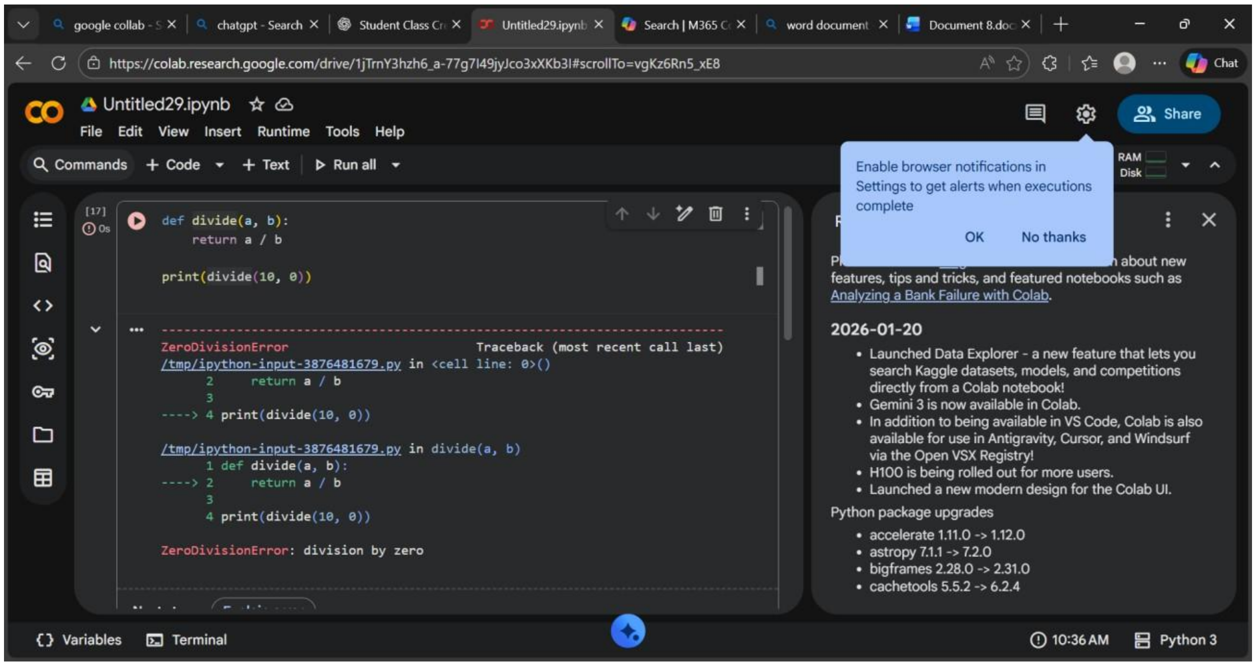
```
1
2
3
4
5
Loop finished.
```

Explantion: The variable `i` was decreasing (`i -= 1`) while the condition required it to increase, causing an infinite loop.
Changing it to `i += 1` allows the loop to reach the stopping condition and terminate correctly.

Task 3: Handling Runtime Errors (Division by Zero)

Prompt: This Python code causes a runtime error. Identify the problem, fix it using tryexcept, and explain the issue. def divide(a, b): return a / b print(divide(10, 0))

Input:Bug Code



```
def divide(a, b):
    return a / b

print(divide(10, 0))
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
/tmp/ipython-input-3876481679.py in <cell line: 0>()
      2     return a / b
      3
----> 4 print(divide(10, 0))

/tmp/ipython-input-3876481679.py in divide(a, b)
      1 def divide(a, b):
----> 2     return a / b
      3
      4 print(divide(10, 0))

ZeroDivisionError: division by zero
```

features, tips and tricks, and featured notebooks such as Analyzing a Bank Failure with Colab.

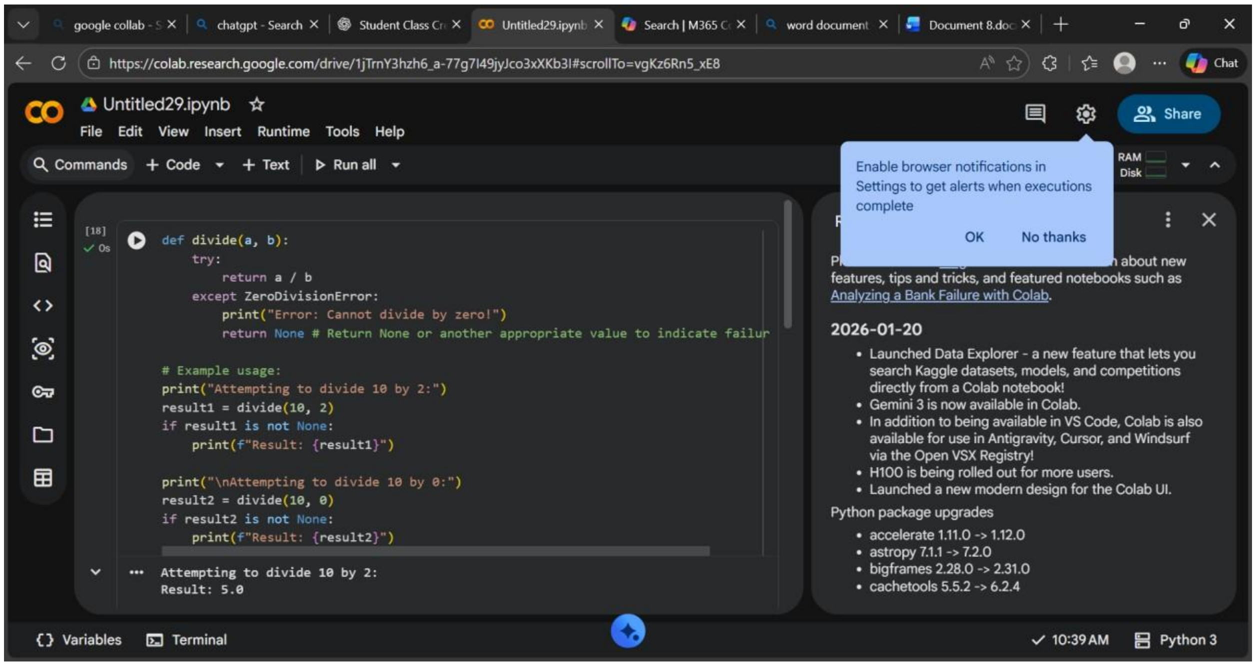
2026-01-20

- Launched Data Explorer - a new feature that lets you search Kaggle datasets, models, and competitions directly from a Colab notebook!
- Gemini 3 is now available in Colab.
- In addition to being available in VS Code, Colab is also available for use in Antigravity, Cursor, and Windsurf via the Open VSX Registry!
- H100 is being rolled out for more users.
- Launched a new modern design for the Colab UI.

Python package upgrades

- accelerate 1.11.0 -> 1.12.0
- astropy 7.1.1 -> 7.2.0
- bigframes 2.28.0 -> 2.31.0
- cachetools 5.5.2 -> 6.2.4

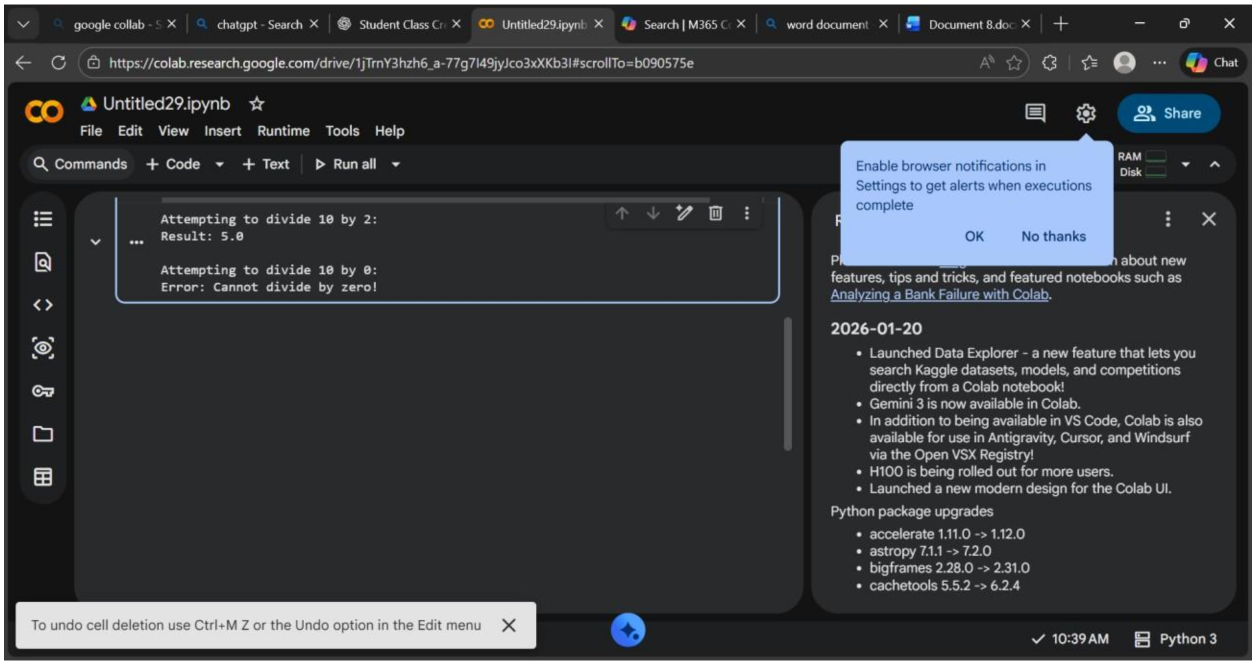
Corrected Code:



```
def divide(a, b):
    try:
        return a / b
    except ZeroDivisionError:
        print("Error: Cannot divide by zero!")
        return None # Return None or another appropriate value to indicate failur

# Example usage:
print("Attempting to divide 10 by 2:")
result1 = divide(10, 2)
if result1 is not None:
    print(f"Result: {result1}")

print("\nAttempting to divide 10 by 0:")
result2 = divide(10, 0)
if result2 is not None:
    print(f"Result: {result2}")
```

```
Attempting to divide 10 by 2:
Result: 5.0
```

Output:

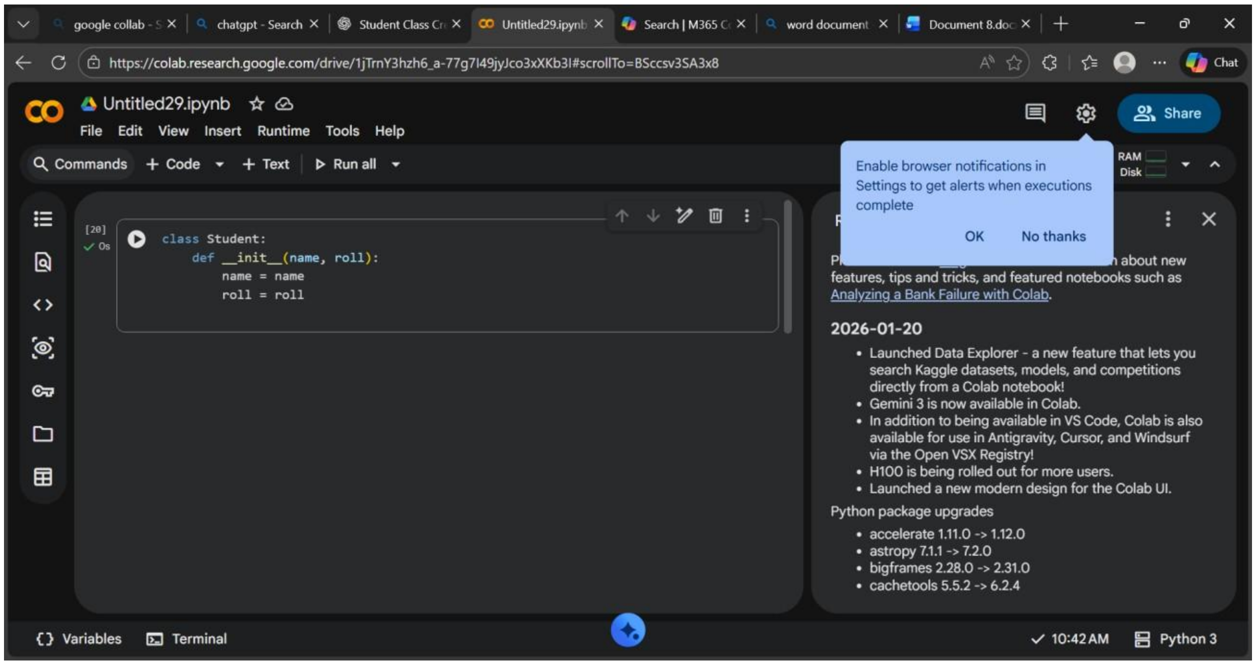

```
Attempting to divide 10 by 2:
Result: 5.0

Attempting to divide 10 by 0:
Error: Cannot divide by zero!
```

Explanation: he program crashes because division by zero is not allowed in Python, causing a `ZeroDivisionError`.
 Using `try-except` prevents the crash and safely handles the error.

Task 4: Debugging Class Definition Errors

Prompt: The following Python class has an error in the constructor. Identify the issue, correct the class definition, and explain why the fix is needed.
class Student: def **init**(name, roll): name = name roll = roll

Input:Bug Code



```
[20]
class Student:
    def __init__(name, roll):
        name = name
        roll = roll
```

Enable browser notifications in Settings to get alerts when executions complete

OK No thanks

Pl about new features, tips and tricks, and featured notebooks such as Analyzing a Bank Failure with Colab.

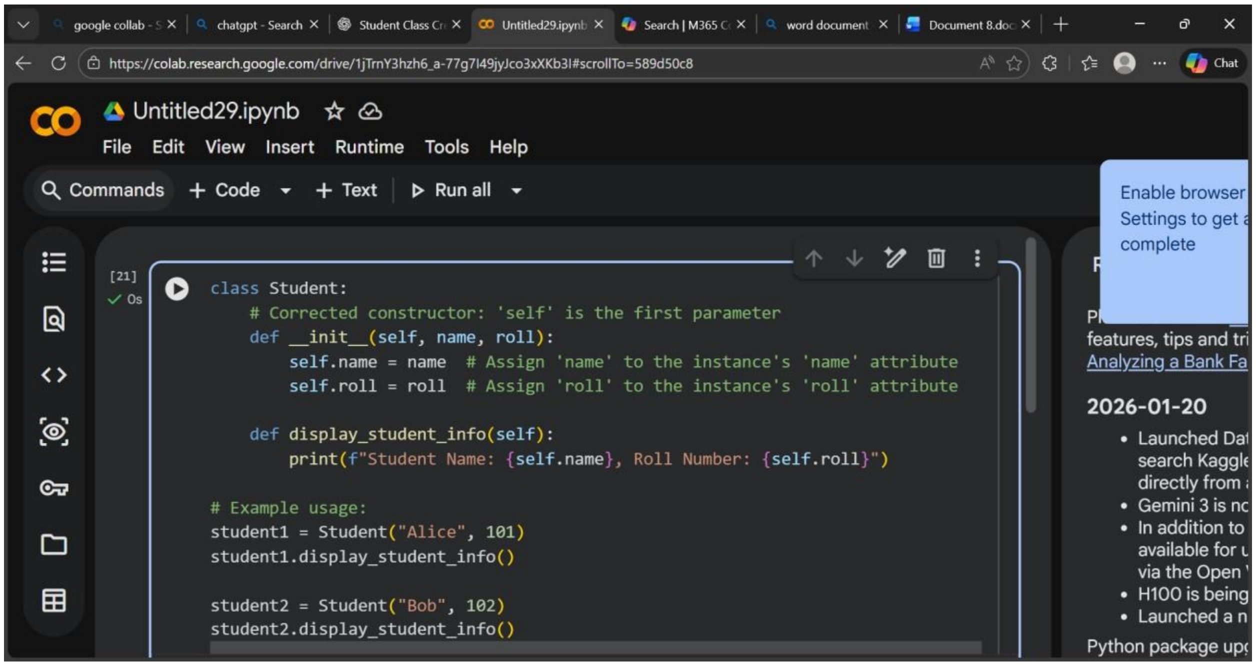
2026-01-20

- Launched Data Explorer - a new feature that lets you search Kaggle datasets, models, and competitions directly from a Colab notebook!
- Gemini 3 is now available in Colab.
- In addition to being available in VS Code, Colab is also available for use in Antigravity, Cursor, and Windsurf via the Open VSX Registry!
- H100 is being rolled out for more users.
- Launched a new modern design for the Colab UI.

Python package upgrades

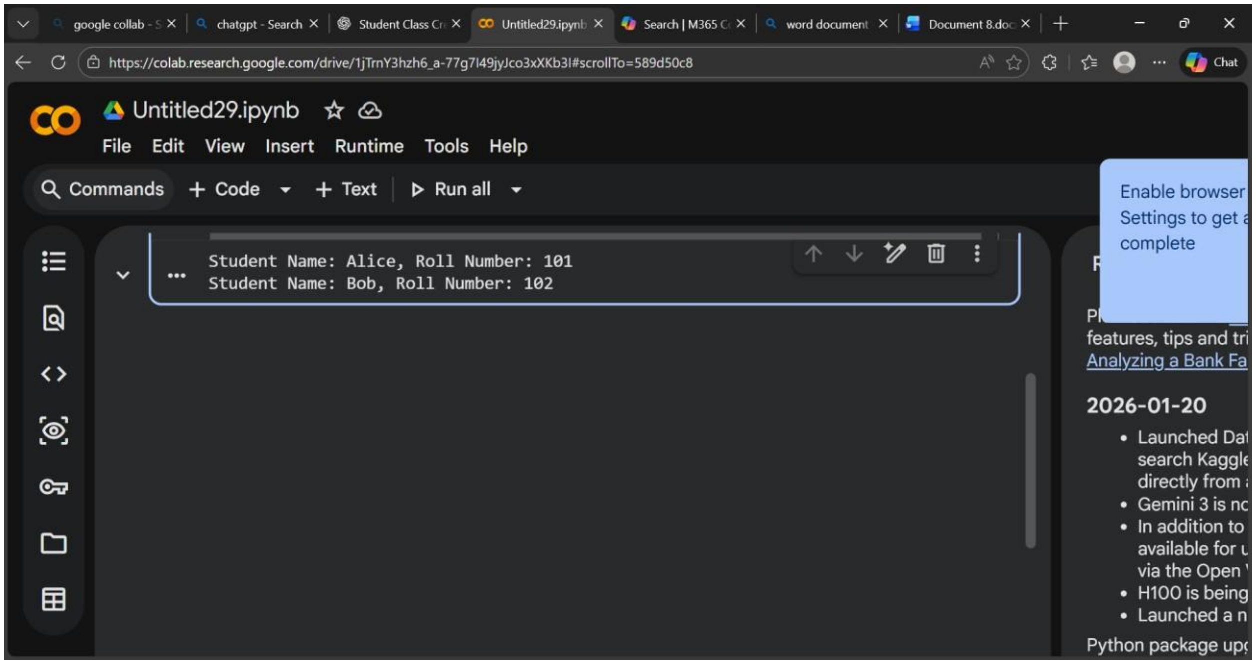
- accelerate 1.11.0 -> 1.12.0
- astropy 7.1.1 -> 7.2.0
- bigframes 2.28.0 -> 2.31.0
- cachetools 5.5.2 -> 6.2.4

Corrected code:



```
[21]
class Student:
    # Corrected constructor: 'self' is the first parameter
    def __init__(self, name, roll):
        self.name = name  # Assign 'name' to the instance's 'name' attribute
        self.roll = roll  # Assign 'roll' to the instance's 'roll' attribute

    def display_student_info(self):
        print(f"Student Name: {self.name}, Roll Number: {self.roll}")

# Example usage:
student1 = Student("Alice", 101)
student1.display_student_info()

student2 = Student("Bob", 102)
student2.display_student_info()
```

Output:


Untitled29.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all

```
Student Name: Alice, Roll Number: 101
Student Name: Bob, Roll Number: 102
```

Explanation: The constructor was missing the `self` parameter, which is required to refer to the object instance.

Using `self.name` and `self.roll` stores values inside the object properly. Task 5:

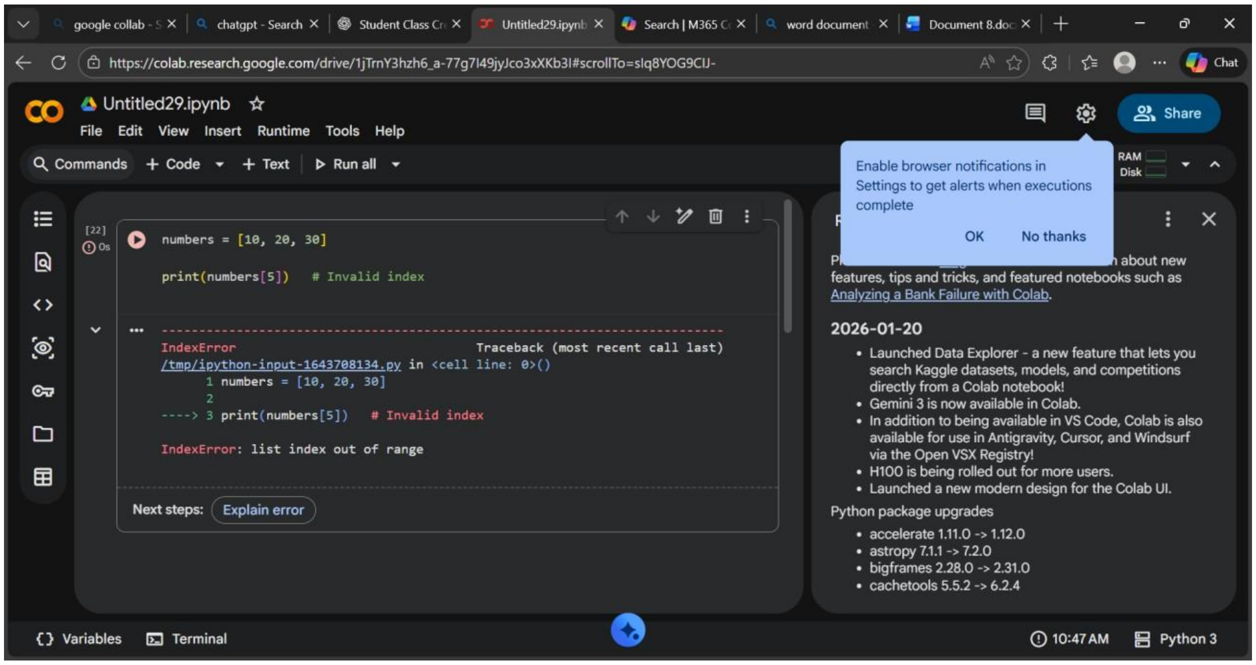
Resolving Index Errors in Lists

Prompt: This Python code causes an IndexError. Identify the issue, correct the code using safe access methods, and explain the problem.

numbers = [10, 20, 30]

(numbers[5])

Input:Bug code



```
numbers = [10, 20, 30]

print(numbers[5])   # Invalid index
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
/tmp/ipython-input-1643708134.py in <cell line: 0>()
      1 numbers = [10, 20, 30]
      2 
----> 3 print(numbers[5])   # Invalid index

IndexError: list index out of range
```

Next steps: Explain error

Enable browser notifications in Settings to get alerts when executions complete

about new features, tips and tricks, and featured notebooks such as Analyzing a Bank Failure with Colab.

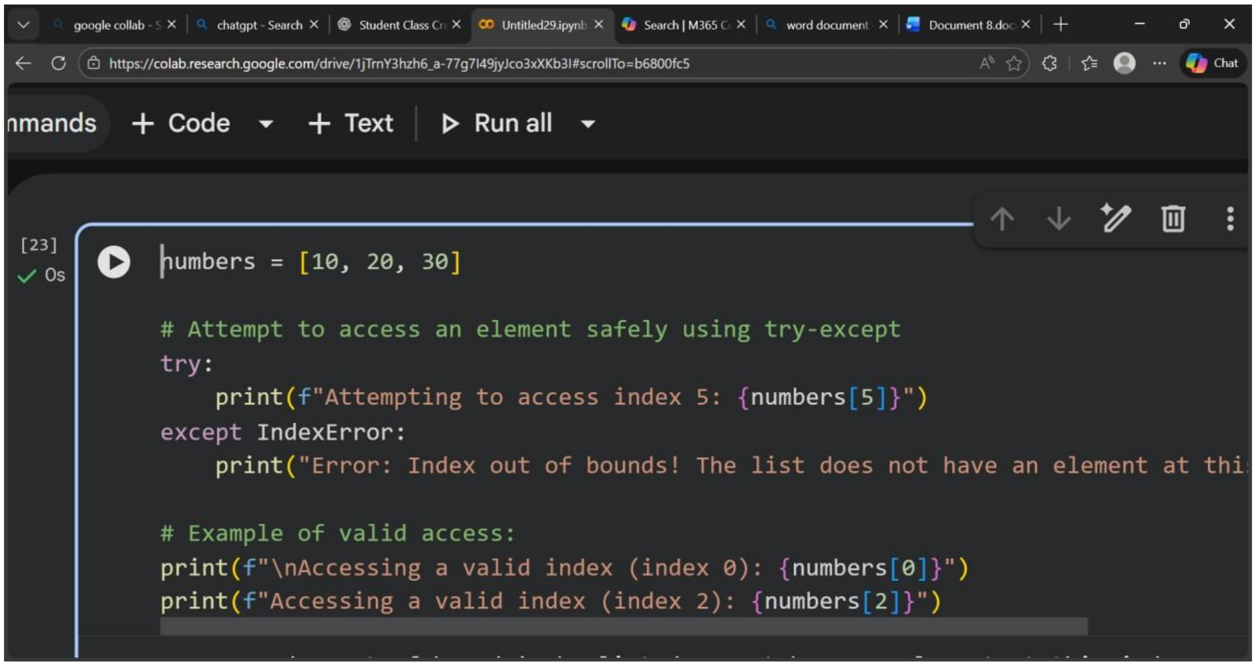
2026-01-20

- Launched Data Explorer - a new feature that lets you search Kaggle datasets, models, and competitions directly from a Colab notebook!
- Gemini 3 is now available in Colab.
- In addition to being available in VS Code, Colab is also available for use in Antigravity, Cursor, and Windsurf via the Open VSX Registry!
- H100 is being rolled out for more users.
- Launched a new modern design for the Colab UI.

Python package upgrades

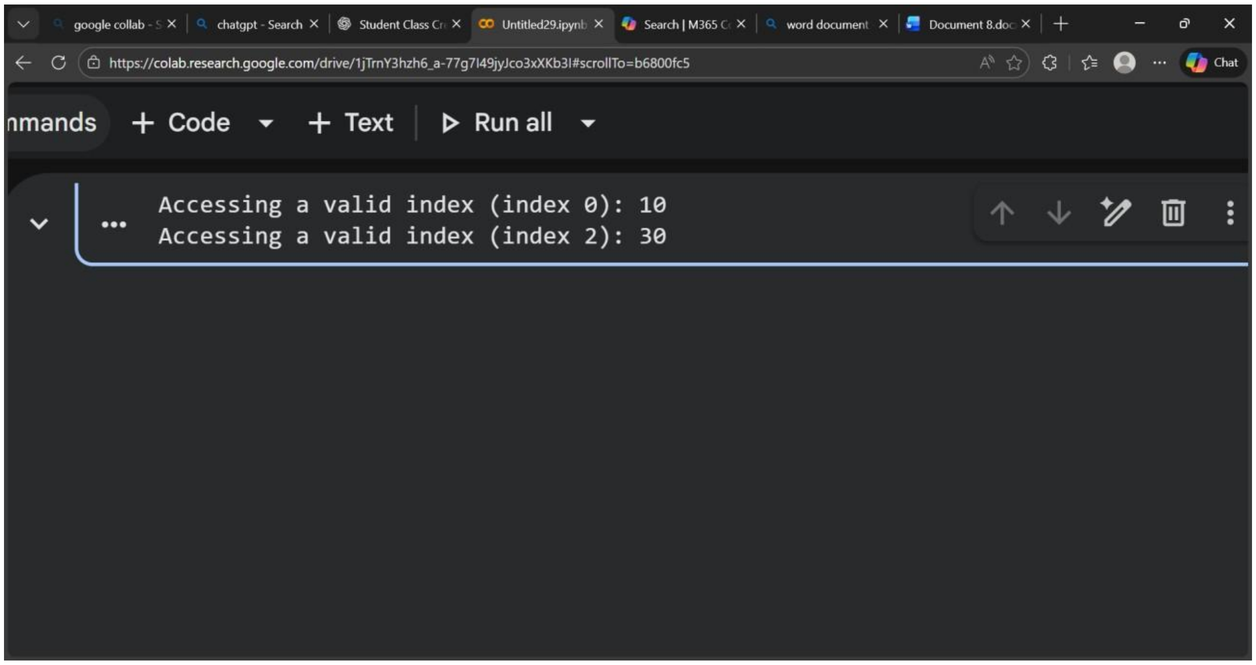
- accelerate 1.11.0 -> 1.12.0
- astropy 7.1.1 -> 7.2.0
- bigframes 2.28.0 -> 2.31.0
- cachetools 5.5.2 -> 6.2.4

Corrected Code:



```
numbers = [10, 20, 30]

# Attempt to access an element safely using try-except
try:
    print(f"Attempting to access index 5: {numbers[5]}")
except IndexError:
    print("Error: Index out of bounds! The list does not have an element at thi

# Example of valid access:
print(f"\nAccessing a valid index (index 0): {numbers[0]}")
print(f"Accessing a valid index (index 2): {numbers[2]}")
```

Output:



```
Accessing a valid index (index 0): 10
Accessing a valid index (index 2): 30
```

Explanation: The program tried to access an index that does not exist in the list, causing an `IndexError`.

Using `len()` to check bounds prevents the program from crashing.