Python Errors

When you write Python code, mistakes are normal. Python shows **errors** to explain what went wrong. Think of them as hints, not failures. By reading the error message, you can usually figure out the problem.

Here are some common beginner errors:

- **SyntaxError**
- **NameError**
- **TypeError**
- **Formatting Issues**

Learning to read and fix these errors will make your coding faster and less frustrating.

Understanding the `SyntaxError` in Python

Python expects code to follow a very strict syntax. If something is missing (like a parenthesis, colon, or quote), Python cannot understand the code and raises a `SyntaxError`.

Example 1: Missing Closing Parenthesis in `print()`

```
print('Hello'
```

Error:

```
SyntaxError: unexpected EOF while parsing
```

Explanation: The `print()` function is missing a closing parenthesis `)`.

Correct version:

```
print('Hello')
```

Example 2: Missing Parenthesis in `input()`

```
name = input('What is your name?'
```

Error:

```
SyntaxError: unexpected EOF while parsing
```

Explanation: The `input()` function needs both opening and closing parentheses. Since the `)` is missing, **Correct version:**

```
name = input('What is your name?')
```

Example 3: Combining Multiple Lines with a Missing Parenthesis

```
print('Welcome to the Classroom Error Demo')
name = input('What is your name?'
print('Nice to meet you', name)
age = input('How old are you? ')
print('You are', age, 'years old')
```

Error:

```
SyntaxError: '(' was never closed
```

Explanation: The second line is missing a `)`.

```
print('Welcome to the Classroom Error Demo')
name = input('What is your name?')
print('Nice to meet you', name)
age = input('How old are you? ')
print('You are', age, 'years old')
```

Quick Tip: If you see a `SyntaxError`, always check:

- Are all parentheses `()`, brackets `[]`, and braces `{}` properly closed?
- Are all quotes (`"`, `'`) in pairs?

Understanding the `NameError` in Python

A **NameError** happens when you try to use a variable or function **before it has been created or defined**. Python looks at the name you wrote (like `age`, `country`, or `name`) and checks if it already exists in memory.

Example 1: Using a Variable Before Defining It

```
print(age)
```

Error:

```
NameError: name 'age' is not defined
```

Explanation:

The variable `age` has never been created before you tried to print it.

Correct version:

```
age = 25
print(age)
```

Example 2: Using the Wrong Variable Name

```
city = 'Paris'
print(country)
```

Error:

```
NameError: name 'country' is not defined
```

Explanation:

You defined `city`, but then tried to print `country`. Since `country` doesn’t exist, Python raises a NameError.

Correct version:

```
city = 'Paris'
print(city)
```

Example 3: Using a Variable Before Assignment

```
 print('Welcome to the Classroom Error Demo')
 print('Nice to meet you', name)
 name = input('What is your name? ')
 print('You are', name)
```

Error:

```
NameError: name 'name' is not defined
```

Explanation:

You tried to use `name` before assigning it a value. Variables must be created **before** they are used.

Correct version:

```
print('Welcome to the Classroom Error Demo')
name = input('What is your name? ')
print('Nice to meet you', name)
print('You are', name)`
```

Quick Tips to Fix a `NameError`

- Check if you **spelled the variable name correctly**.
- Make sure the variable is **assigned a value before you use it**.
- Remember that Python is **case-sensitive** (`Name` ≠ `name`).

Understanding the `TypeError` in Python

A **TypeError** happens when you try to perform an operation on data of the **wrong type**.
For example:

- You can’t add a **string** and an **integer** directly.
- You can’t subtract numbers from **strings**.

Example 1: Subtracting a Number from a String

```
num = '7'
print(num - 2)
```

Error:

```
TypeError: unsupported operand type(s) for -: 'str' and 'int`
```

Explanation:
`num` is `'7'`, a string. You can’t subtract a number from a string.

Correct versions:

```
#Convert the string to a number before subtracting
num = '7'
print(int(num) - 2)
```

Example 3: Input is Always a String

```
age = input('How old are you? ')
print('Next year you will be ', age + 1)
```

Error:

```
TypeError: can only concatenate str (not "int") to str
```

Explanation:

The `input()` function always returns a **string**, even if the user types a number.

Here, `age` is a string, but you tried to do `"age" + 1` (string + integer).

Correct versions:**

```
#Convert to int before adding
age = int(input('How old are you? '))
print('Next year you will be ' + (age + 1))
```

Quick Tips to Fix a `TypeError`

- Convert strings to numbers with `int()` or `float()` when doing math.
- Convert numbers to strings with `str()` when combining with text.

Understanding Formatting (Output Issues) in Python

Sometimes your code **runs without errors**, but the **output looks wrong** — not what you intended.

This usually happens because of **string formatting mistakes**.

Example 1: Forgetting the `f` in an f-string

```
age = 20
print('Your age is {age}')
```

Output:

```
Your age is {age}
```

Why?

Without the `f`, Python just prints the text **literally**, including `{age}`.

Correct version:

```
print(f'Your age is {age}')
```

Example 2: Missing `f` Again

```
food = 'pizza'
print('I like {food} too!')
```

Output:

```
I like {food} too!
```

Correct version:

```
print(f'I like {food} too!')
```

Quick Tips to Fix Formatting Issues

- Remember: `print(f'...{variable}...')` inserts values inside `{}`.

Practice Questions

Instructions: **fix the broken program** and explain the errors you find.

1. **Run the code.**

2. **Read the error message** (or look at the wrong output).

3. **Write down what the error is** and why it happened.

    - If the code runs but the output looks wrong, mark the type as **"Formatting"**.

4. **Fix the error in the code.**

--Example how you you should write down the error you found in the code:--

Error/Fix 1:

- **Type:** NameError
- **Line:** 3
- **Why did it happen:** There were two = signs (==) when we only needed one (=).

Error/Fix 2:

- **Type:** SyntaxError
- **Line:** 7
- **Why did it happen:** The variable name had spaces.

Question_1:

```
print('Welcome to class')

student == input('What is your name? ')

print('Hello {student}, glad you're here!')

print(f'Enjoy the lesson, {Student}')
```

Question_2:

```
print('Welcome to the music app!')

genre = input('What music genre do you like? ')

print(F'Nice choice, {genre}')

artist name = input('Who is your favorite artist? ')

print('Your favorite artist is {artist_name}')
```

Question_3:

```
print('Hi there!')

hobby = input('What's your hobby? ')

print(f'Great! {hobby} is fun to do.')

print('I also enjoy {hobby}')
```

Question_4:

```
print('Welcome to Fitness Tracker!')

userName == input('Enter your name: ')

print(f'Hello {username}, let's get started!')

steps_today = input('How many steps did you walk today?')

goal steps = 10000

progress = steps_today / goal_steps

print('You completed {progress} of your goal!')
```

Question_5:

```
name  =  input('What is your name?'

print('Nice to meet you', name)

print('Your city is', city)

city  =  input('What city do you live in? ')

age  =  input('How old are you? ')

print('Next year you will be ',   age +  1)

score  =  95

print('Your score is {score}')

print('Hello',name,'welcome to the class')
```