Description

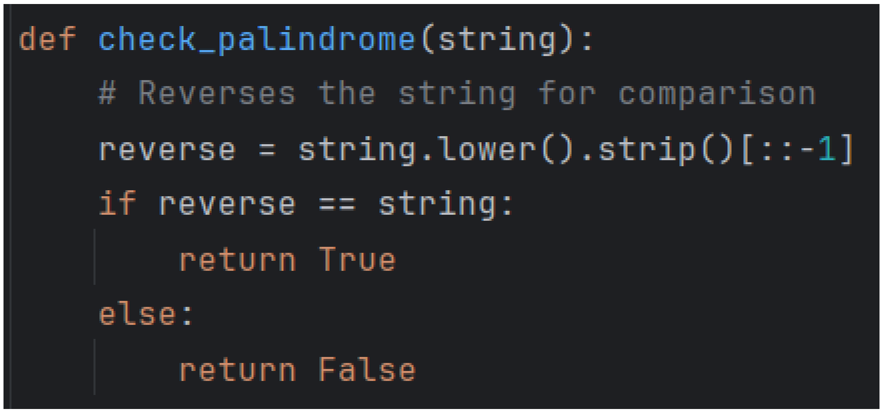
The program consists of various string-related functions that alert and return different outputs depending on the function used.

Principles

- KISS: Keep code simple and “stupid” for people to read
- DRY code: Don’t Repeat Yourself; avoid duplication
- Documentation: Leave comments to explain variables and functions

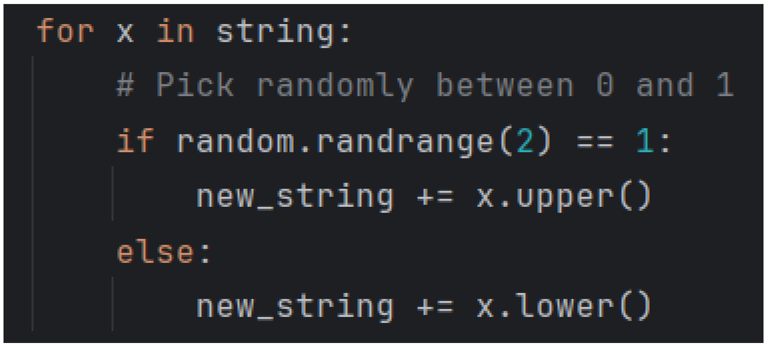
Correct Implementation

The program is generally minimal in code and complexity, using simple terms to match with individual functions and variables.



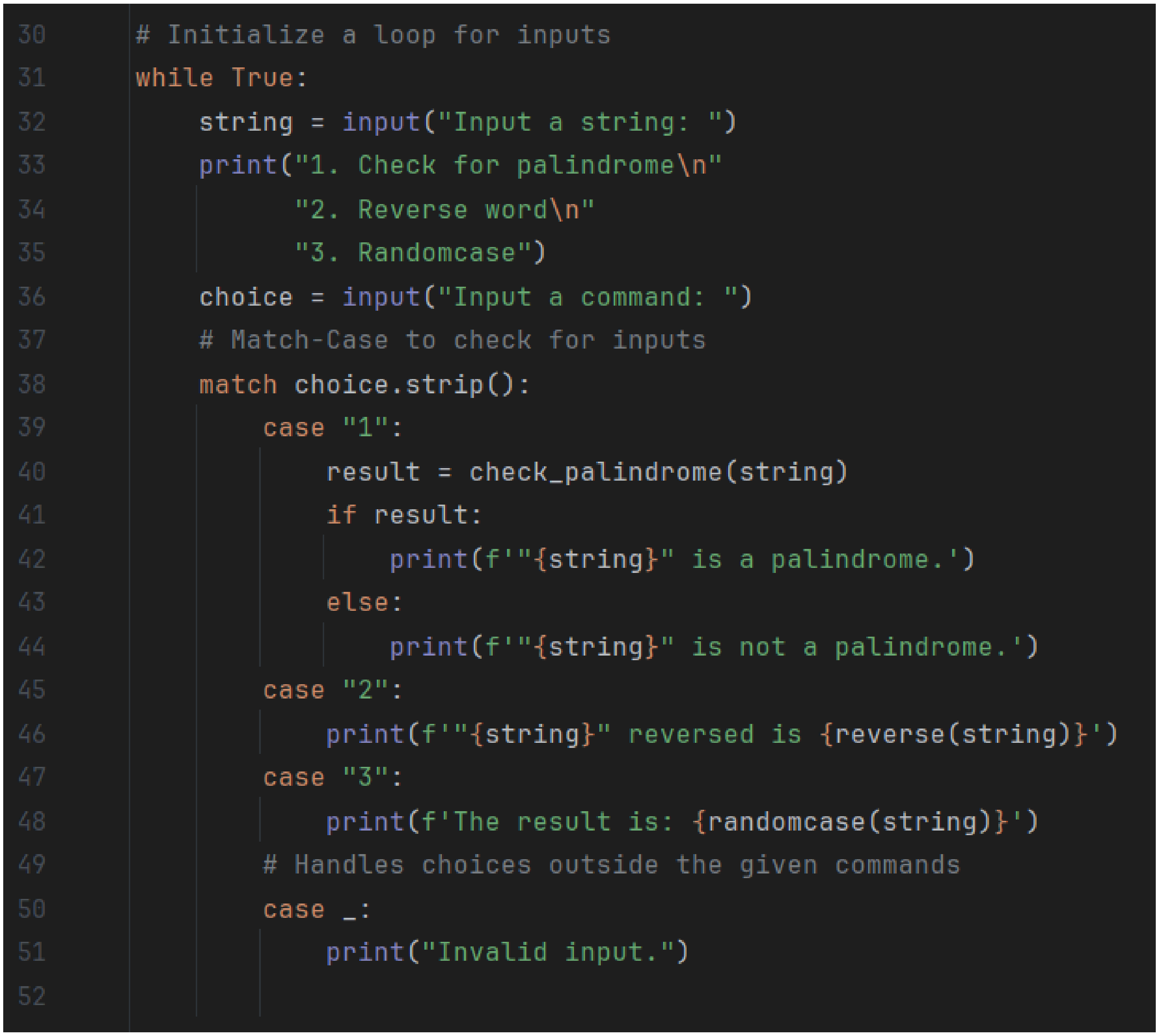
```python
def check_palindrome(string):
    # Reverses the string for comparison
    reverse = string.lower().strip()[::-1]
    if reverse == string:
        return True
    else:
        return False
```

The programs avoids duplication by using loops whenever possible.



```python
for x in string:
    # Pick randomly between 0 and 1
    if random.randrange(2) == 1:
        new_string += x.upper()
    else:
        new_string += x.lower()
```

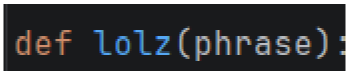
Comments are placed in appropriate locations, avoiding overuse and redundancy.



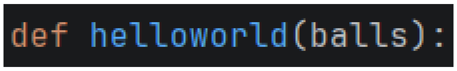
```
# Initialize a loop for inputs
while True:
    string = input("Input a string: ")
    print("1. Check for palindrome\n"
          "2. Reverse word\n"
          "3. Randomcase")
    choice = input("Input a command: ")
    # Match-Case to check for inputs
    match choice.strip():
        case "1":
            result = check_palindrome(string)
            if result:
                print(f'"{string}" is a palindrome.')
            else:
                print(f'"{string}" is not a palindrome.')
        case "2":
            print(f'"{string}" reversed is {reverse(string)}')
        case "3":
            print(f'The result is: {randomcase(string)}')
        # Handles choices outside the given commands
        case _:
            print("Invalid input.")

```

Incorrect Implementation

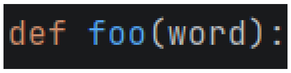
The function names are not coherent nor describe what they actually do



```
def lolz(phrase):
```

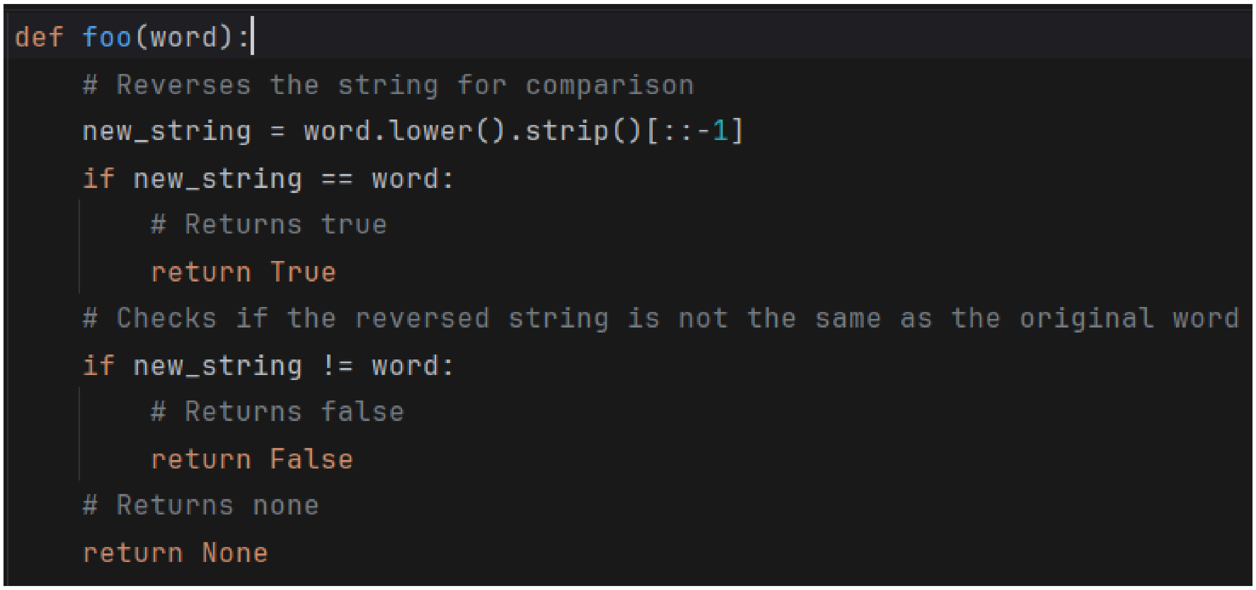


```
def helloworld(balls):
```



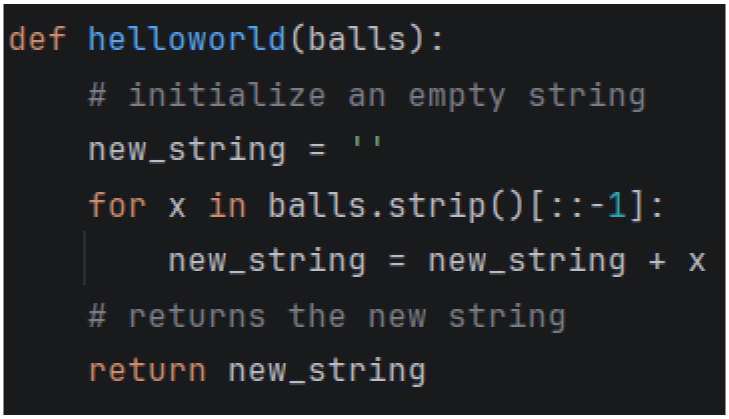
```
def foo(word):
```

Comments are unnecessary and redundant



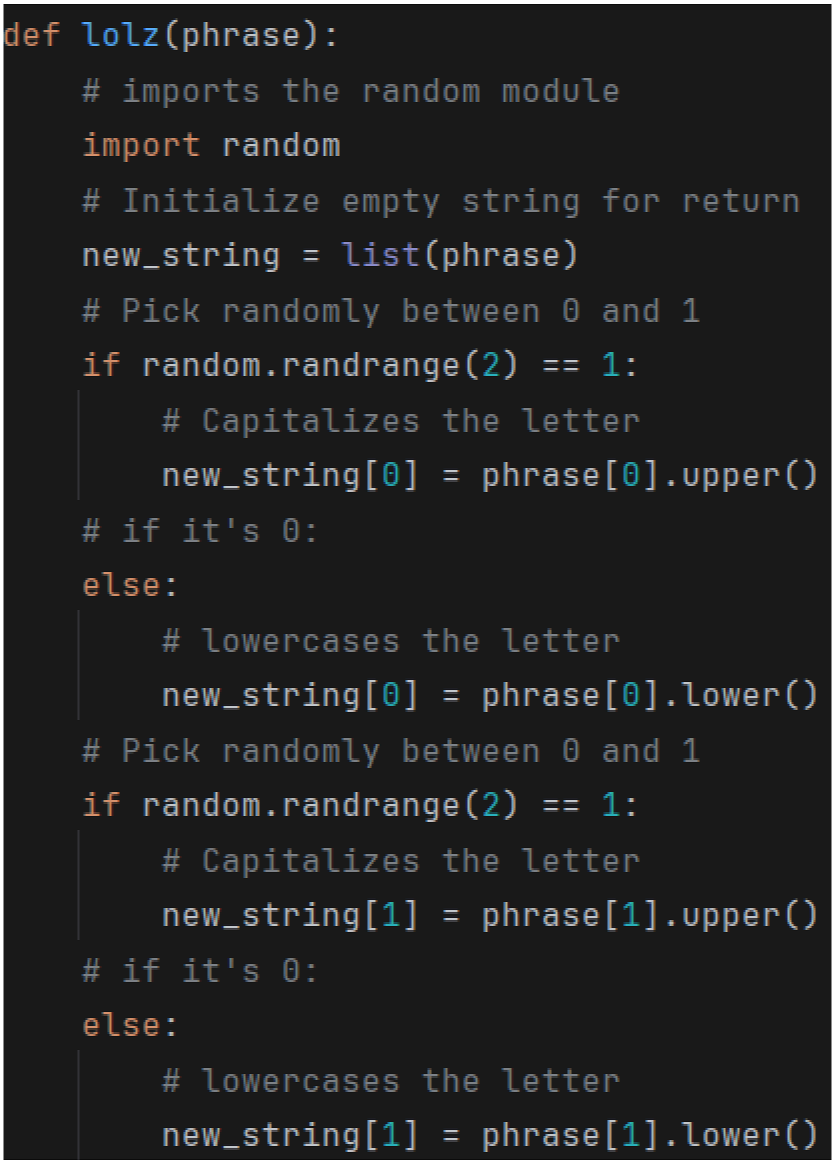
```python
def foo(word):
    # Reverses the string for comparison
    new_string = word.lower().strip()[::-1]
    if new_string == word:
        # Returns true
        return True
    # Checks if the reversed string is not the same as the original word
    if new_string != word:
        # Returns false
        return False
    # Returns none
    return None
```

The function could be simplified further, the for loop is unnecessary

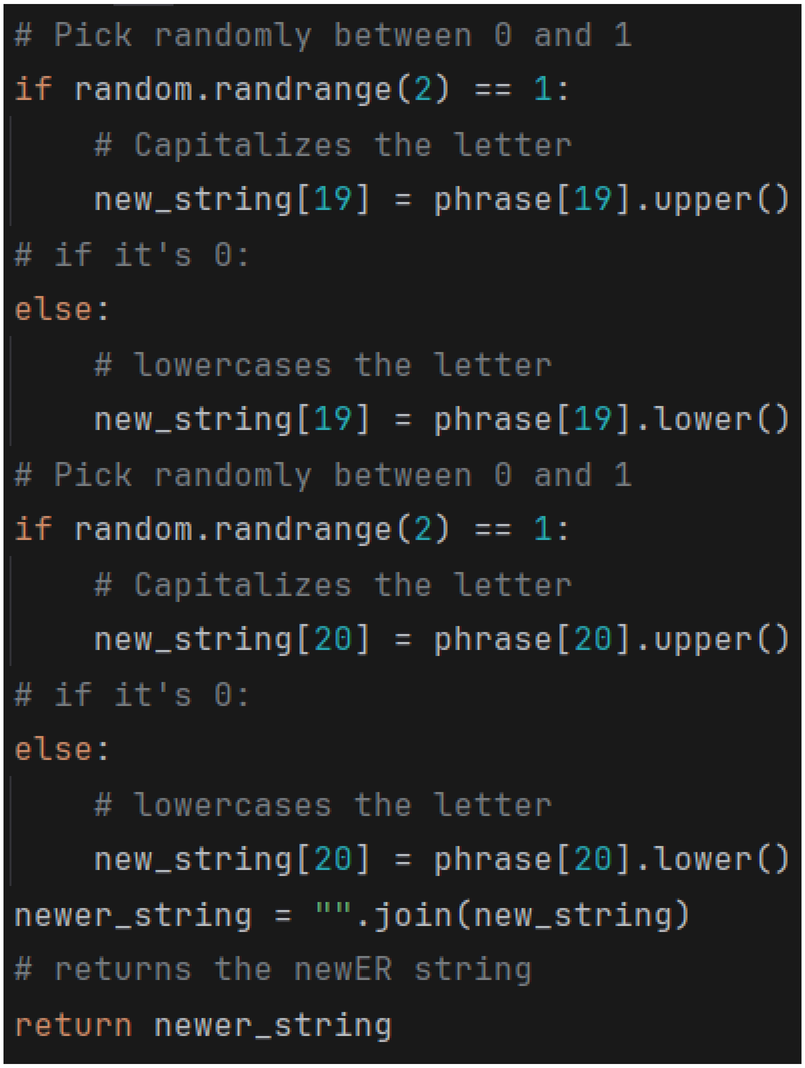


```python
def helloworld(balls):
    # initialize an empty string
    new_string = ''
    for x in balls.strip()[::-1]:
        new_string = new_string + x
    # returns the new string
    return new_string
```

Comments and logic are extremely redundant, and the function is hardcoded. The lack of spacing in 175 lines also makes it very unreadable.

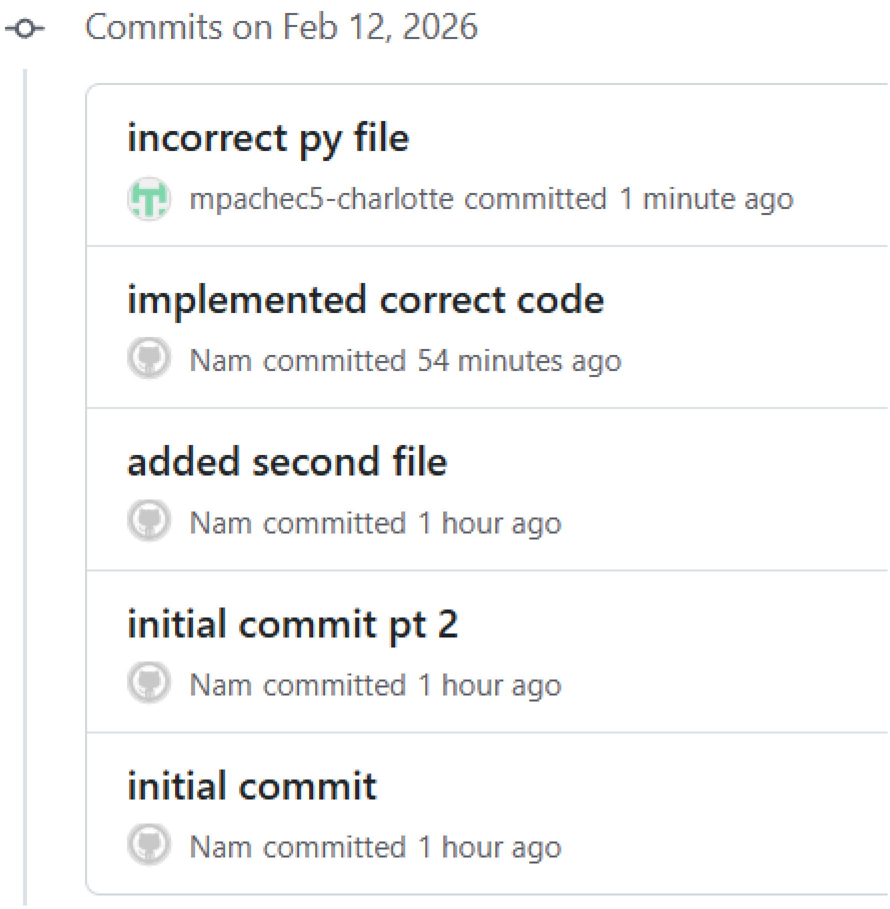


```python
def lolz(phrase):
    # imports the random module
    import random
    # Initialize empty string for return
    new_string = list(phrase)
    # Pick randomly between 0 and 1
    if random.randrange(2) == 1:
        # Capitalizes the letter
        new_string[0] = phrase[0].upper()
    # if it's 0:
    else:
        # lowercases the letter
        new_string[0] = phrase[0].lower()
    # Pick randomly between 0 and 1
    if random.randrange(2) == 1:
        # Capitalizes the letter
        new_string[1] = phrase[1].upper()
    # if it's 0:
    else:
        # lowercases the letter
        new_string[1] = phrase[1].lower()
```



```python
# Pick randomly between 0 and 1
if random.randrange(2) == 1:
    # Capitalizes the letter
    new_string[19] = phrase[19].upper()
# if it's 0:
else:
    # lowercases the letter
    new_string[19] = phrase[19].lower()
# Pick randomly between 0 and 1
if random.randrange(2) == 1:
    # Capitalizes the letter
    new_string[20] = phrase[20].upper()
# if it's 0:
else:
    # lowercases the letter
    new_string[20] = phrase[20].lower()
newer_string = "".join(new_string)
# returns the newER string
return newer_string
```

Github Proof



Commits on Feb 12, 2026

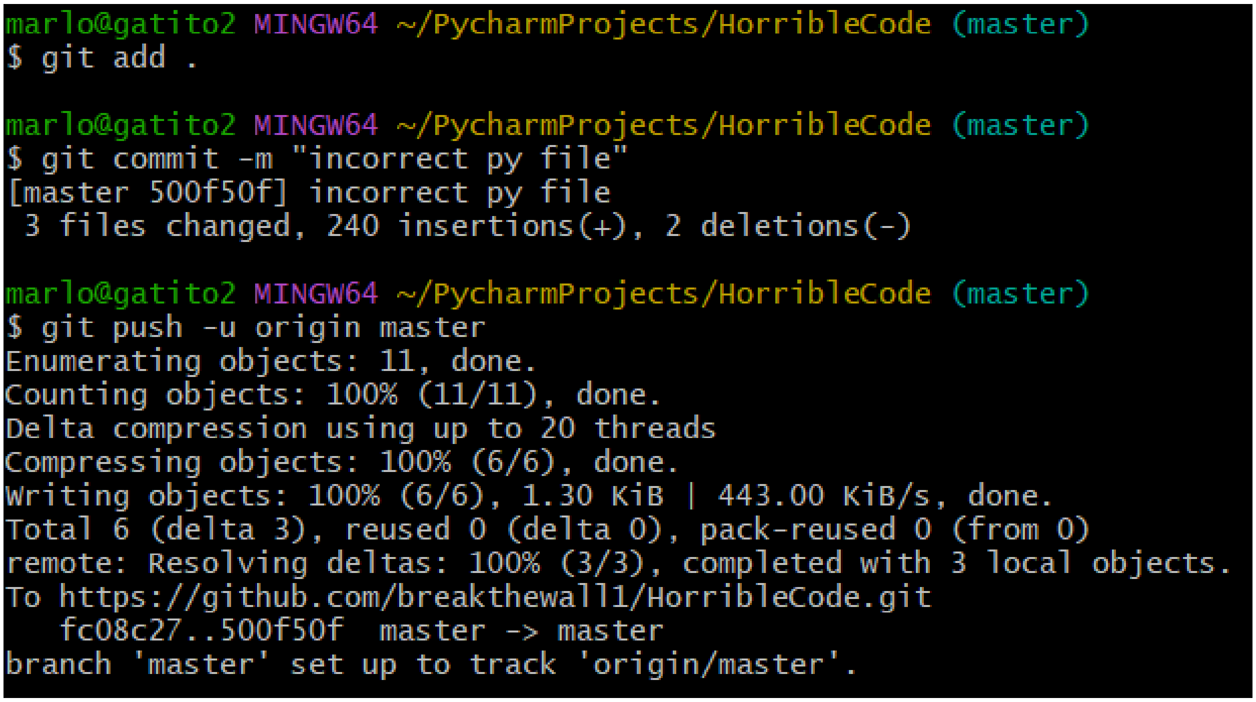
incorrect py file
mpachec5-charlotte committed 1 minute ago

implemented correct code
Nam committed 54 minutes ago

added second file
Nam committed 1 hour ago

initial commit pt 2
Nam committed 1 hour ago

initial commit
Nam committed 1 hour ago



```
marlo@gatito2 MINGW64 ~/PycharmProjects/HorribleCode (master)
$ git add .

marlo@gatito2 MINGW64 ~/PycharmProjects/HorribleCode (master)
$ git commit -m "incorrect py file"
[master 500f50f] incorrect py file
 3 files changed, 240 insertions(+), 2 deletions(-)

marlo@gatito2 MINGW64 ~/PycharmProjects/HorribleCode (master)
$ git push -u origin master
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 20 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.30 KiB | 443.00 KiB/s, done.
Total 6 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/breakthewall1/HorribleCode.git
   fc08c27..500f50f  master -> master
branch 'master' set up to track 'origin/master'.
```