localhost:8888/notebooks/first%20project

jupyter first project Last Checkpoint: 1 hour ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)

[20]:
```
def sum_of_numbers(a , b):
    return a + b

a , b = 5, 9
print(f"The sum of {a} and {b} is {sum_of_two_numbers(a, b)}.")
```

```
The sum of 5 and 9 is 14.
```

[21]:
```
def odd_or_even(number):
    return "Even" if number % 2 == 0 else "odd"

num = 4
print (f"The number {num} is { odd_or_even(num)}.")
```

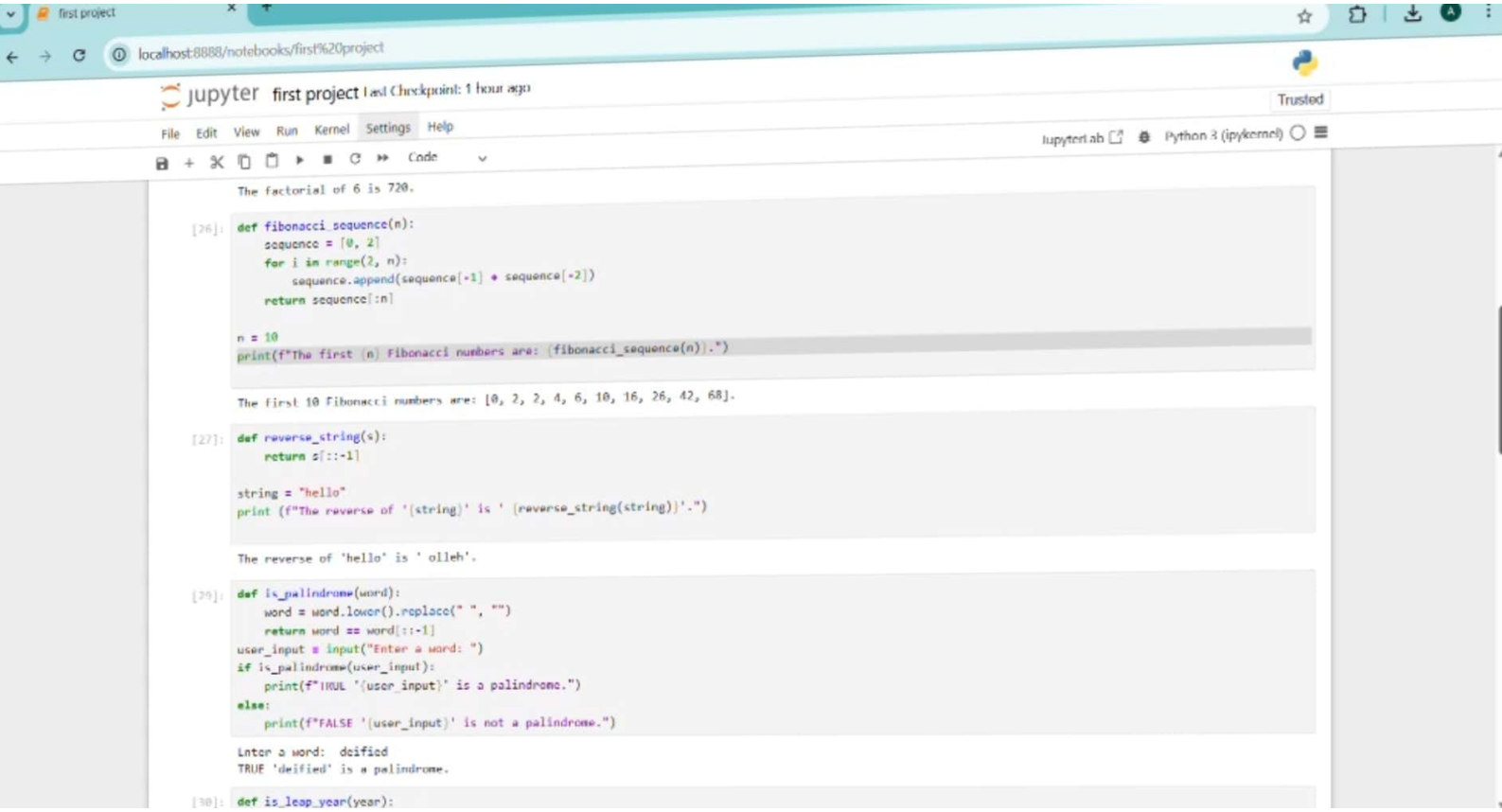
```
The number 4 is Even.
```

[23]:
```
def factorial(n):
    result = 1
    for i in range(1, n +1):
        result *= i
    return result

n = 6
print(f"The factorial of {n} is {factorial(n)}.")
```

```
The factorial of 6 is 720.
```

[26]:
```
def fibonacci_sequence(n):
```


first project
localhost:8888/notebooks/first%20project
jupyter first project Last Checkpoint: 1 hour ago
Trusted
File Edit View Run Kernel Settings Help
JupyterLab
Python 3 (ipykernel)
Code
The factorial of 6 is 720.
[26]:
def fibonacci_sequence(n):
sequence = [0, 2]
for i in range(2, n):
sequence.append(sequence[-1] + sequence[-2])
return sequence[:n]
n = 10
print(f"The first {n} Fibonacci numbers are: {fibonacci_sequence(n)}.")
The first 10 Fibonacci numbers are: [0, 2, 2, 4, 6, 10, 16, 26, 42, 68].
[27]:
def reverse_string(s):
return s[::-1]
string = "hello"
print (f"The reverse of '{string}' is ' {reverse_string(string)}'.")
The reverse of 'hello' is ' olleh'.
[29]:
def is_palindrome(word):
word = word.lower().replace(" ", "")
return word == word[::-1]
user_input = input("Enter a word: ")
if is_palindrome(user_input):
print(f"TRUE '{user_input}' is a palindrome.")
else:
print(f"FALSE '{user_input}' is not a palindrome.")
Enter a word: deified
TRUE 'deified' is a palindrome.
[30]:
def is_leap_year(year):

localhost:8888/notebooks/first%20project

jupyter first project Last Checkpoint: 1 hour ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)

[30]:
```
def is_leap_year(year):
    return year % 2 == 0 and (year % 100 != 0 or year % 200 == 0)

year = 2024
print(f"Is {year} a leap year? {is_leap_year(year)}.")
```

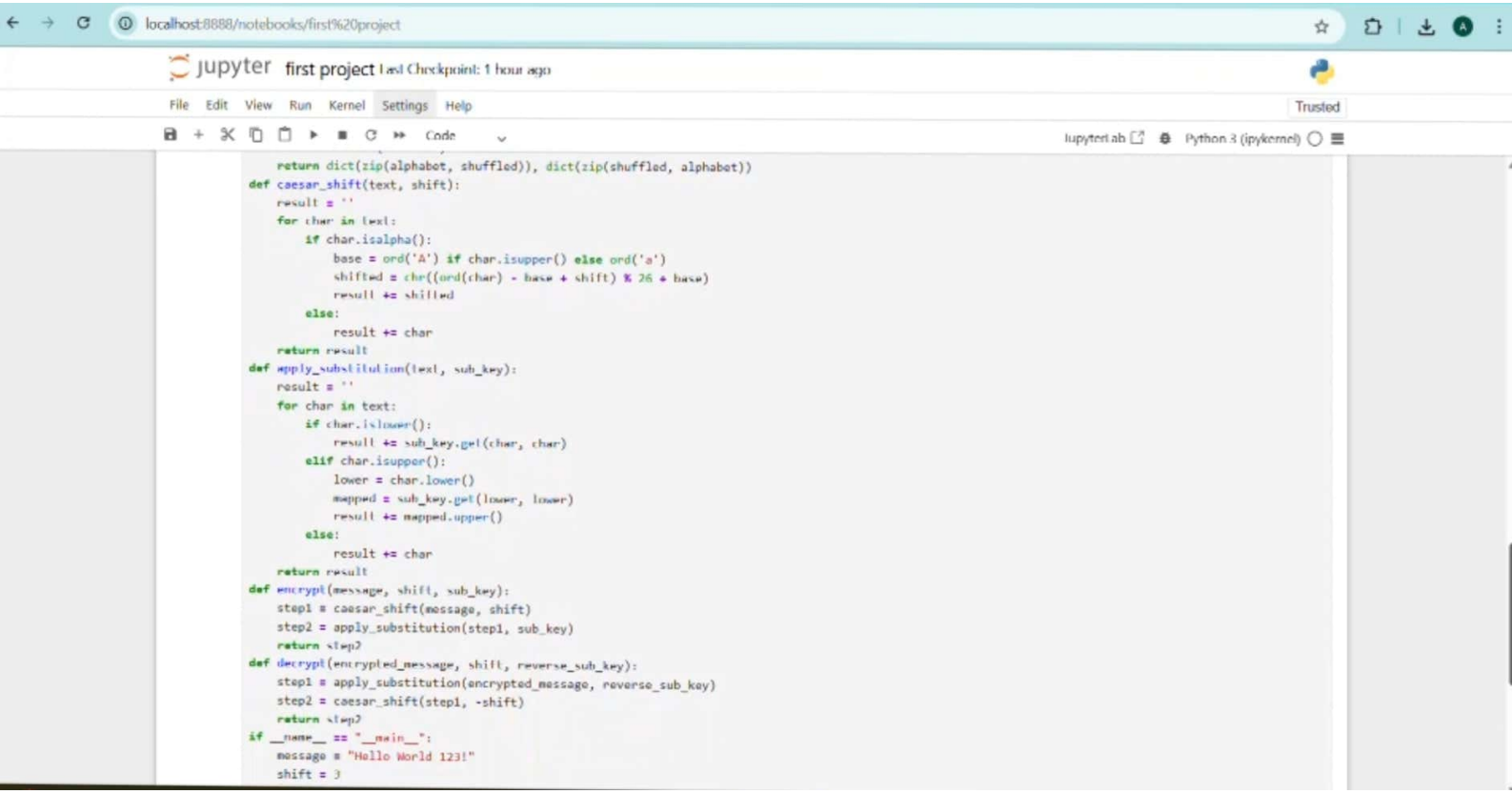
```
Is 2024 a leap year? True.
```

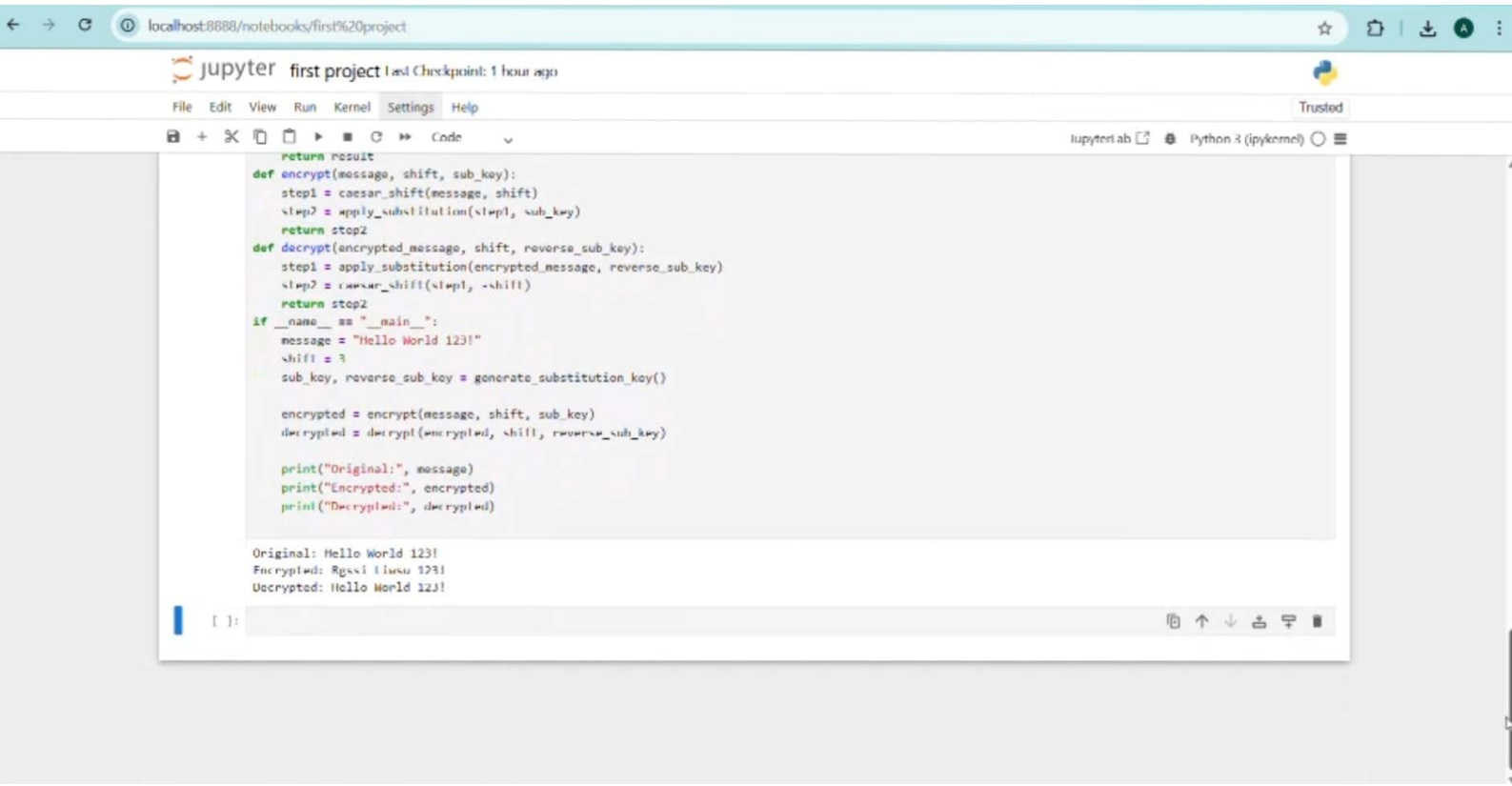
[31]:
```
def is_armstrong_number(n):
    digits = str(n)
    return n == sum(int(d) ** len(digits) for d in digits)

num = 153
print(f"Is {num} an Armstrong number? {is_armstrong_number(num)}.")
```

```
Is 153 an Armstrong number? True.
```

*[34]:
```
import random
import string
def generate_substitution_key():
    alphabet = list(string.ascii_lowercase)
    shuffled = alphabet[:]
    random.shuffle(shuffled)
    return dict(zip(alphabet, shuffled)), dict(zip(shuffled, alphabet))
def caesar_shift(text, shift):
    result = ''
    for char in text:
        if char.isalpha():
            base = ord('A') if char.isupper() else ord('a')
            shifted = chr((ord(char) - base + shift) % 26 + base)
            result += shifted
        else:
            result += char
```


localhost:8888/notebooks/first%20project
jupyter first project Last Checkpoint: 1 hour ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
return dict(zip(alphabet, shuffled)), dict(zip(shuffled, alphabet))
def caesar_shift(text, shift):
result = ''
for char in text:
if char.isalpha():
base = ord('A') if char.isupper() else ord('a')
shifted = chr((ord(char) - base + shift) % 26 + base)
result += shifted
else:
result += char
return result
def apply_substitution(text, sub_key):
result = ''
for char in text:
if char.islower():
result += sub_key.get(char, char)
elif char.isupper():
lower = char.lower()
mapped = sub_key.get(lower, lower)
result += mapped.upper()
else:
result += char
return result
def encrypt(message, shift, sub_key):
step1 = caesar_shift(message, shift)
step2 = apply_substitution(step1, sub_key)
return step2
def decrypt(encrypted_message, shift, reverse_sub_key):
step1 = apply_substitution(encrypted_message, reverse_sub_key)
step2 = caesar_shift(step1, -shift)
return step2
if __name__ == "__main__":
message = "Hello World 123!"
shift = 3


localhost:8888/notebooks/first%20project
jupyter first project Last Checkpoint: 1 hour ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
return result
def encrypt(message, shift, sub_key):
step1 = caesar_shift(message, shift)
step2 = apply_substitution(step1, sub_key)
return step2
def decrypt(encrypted_message, shift, reverse_sub_key):
step1 = apply_substitution(encrypted_message, reverse_sub_key)
step2 = caesar_shift(step1, -shift)
return step2
if __name__ == "__main__":
message = "Hello World 123!"
shift = 3
sub_key, reverse_sub_key = generate_substitution_key()
encrypted = encrypt(message, shift, sub_key)
decrypted = decrypt(encrypted, shift, reverse_sub_key)
print("Original:", message)
print("Encrypted:", encrypted)
print("Decrypted:", decrypted)
Original: Hello World 123!
Encrypted: Rgssi Liwsu 123!
Decrypted: Hello World 123!
[ ]: