# codeStudy-Hamon

This is a repo for the code study and unit tests for Hamon Technologies code test

## Functions and Operations

The function named `s` takes a single argument string as input and returns a dictionary `r` containing the count of each character present in the input string.

Here are the step by operations in the function

- Initialize the empty dictionary `r = {}` to store the count of characters.
- Using `for` loop it iteracts through each character `i` in the input string `s`.
- Inside the loop it checks if i is already present as a key in `r`. If it is, it increments the count of the character stored in the value by 1 (ie: `r[i] += 1`). If not, it creates a new key-value pair as key the character and value the count to a default `1`.
- After the iterating the function returns the dictionary `r` containing the count of characters.

## Suggestions

- **Adding a docstring**:Adding a docstring at beginning of the function can be used to understand the purpose, parameters and return value. This improves readability and maintainability.
- **Descriptive variable names**: Instead of using a single letter variable name like `i`, `s` can consider much more desscriptive name like `each`, `character`. This can improve code readability.
- **Use defaultdict**: The collections module in python provide a `defaultdict` class, which automatically initializes the value for a missing key with a default value. This can eliminate the checking part.

```
from collections import defaultdict

def f(s):
    '''
    The function returns the number of occurrences of each character in the
given string s.
    :param s: str
    :return: dict
    '''
    r = defaultdict(int)
    for char in s:
        r[char] += 1
    return r
```

- **Use Counter**: The collection module also provides a `Counter` class, for counting the characters in a string. Using `Counter` can make the code more efficient.

```
from collections import Counter

def f(s):
    '''
    The function returns the number of occurrences of each character in the
given string s.
    :param s: str
    :return: dict
    '''
    return Counter(s)
```

## Running the code tests

To run the tests, run the following command from the terminal

```
python test_utils.py
```