7940 Lab1-2

STN: MaFengzhi STO: 20401043

What is a conflict in Git? What is the cause of such conflict?

When there are more than one person or machine updating the repo, it is possible that a conflict will be raised. When two users tries to work at the same time, whoever push first will success. The conflict will appear even if the two users make the exact same changes on the source.

How can we resolve a conflict?

When push fail, try to pull again first.

1) If the local change and the remote changes are in different files, it will merge automatically.
2) If the changes in on the same file, you need to resolve the conflict manually and commit/merge again.
3) The last resort is to

discard your current commit by reset; or force push.

What practices can be do to avoid having a conflict?

Conflict can be avoided each time we pull before commit.

Branching is a good methodology to manage a project. After completion of a feature, the developer will issue a pull request (PR). Her work will be tested and reviewed and ultimately merged to the master branch.

What is the purpose of return in python?

Python defines a function using the keyword def. A function is very useful. Normally a function include some parameters and a return value (but not always). A function can return a value so that you can off-load some calculation to it. You can also pass a list into a function or return that from a function.

In general, you can get the calculation result you want through return.

Where can we define a function parameter? How can we use it?

Python defines a function using the keyword def. A function is very useful. Normally a function include some parameters and a return value (but not always).

By calling the function you write, fill in the required parameter values to reference the function. Just like this:

```
# call the function convert_number
# convert all elements (except the first one) into number and return it as a list
def convert_number(l):
    a = []
    for i in range(1, len(l)):
        a.append(int(l[i]))
    return a

y = convert_number(text[0])

print("y")
print(y)
```

Copy the code that you have written for today's lab.

```
import json
import requests
site="https://api.npoint.io/2b57052af2060e84dc86"
# Your code goes here
# Trying to load JSON into text
r = requests.get(site)
print(r.json())
text = r.json()['users']
# Debug
for i in text:
    print("parse " + str(i))

# call the function convert_number
# convert all elements (except the first one) into number and return it
as a list
def convert_number(l):
    a = []
    for i in range(1, len(l)):
        a.append(int(l[i]))
    return a

y = convert_number(text[0])
print("y")
print(y)

# call the function replace_number
# replace all number 1 by the number 10 in the function
def replace_number(number_list, being_replace, to_replace):
    for i in range(len(number_list)):
        if(number_list[i] == being_replace):
            number_list[i] = to_replace
    return number_list
z = replace_number(number_list = y, being_replace = 1, to_replace = 10)
print("z")
print(z)
sum = 0
for i in z:
    sum = sum + i
    print("sum = " + str(sum) + "; i =" + str(i))
print ("Total = " + str(sum))
```

```python
def allFactor(n):
    if n == 0: return [0]
    if n <=3: return [1]
    tmp = n
    rlist = [1]
    i = 2
    while i <= tmp:
        if tmp % i == 0:
            if i != rlist[-1]:
                rlist.append(i)
            tmp = tmp // i
            i = 2
            continue
        i += 1
    return rlist if n != rlist[-1] else rlist[:-1]


# Write a function that prints all factors of the given parameter x
x = 10
def print_factor(x):
  # your code here
    print(allFactor(x))


# Write a program that be able to find all factors of the numbers in
the list l
l = [52633, 8137, 1024, 999]
# your code here
def print_allFactor(l):
    for i in range(len(l)):
        print(l[i], ':' , allFactor(l[i]))
```