Assignment 1: Git and GitHub Page
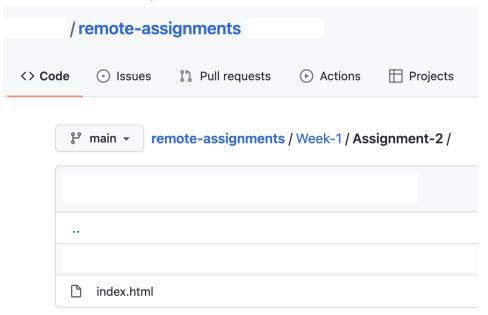
Follow this document to host your own website on Github Page and take steps below to hand in your assignments:

1. Create a new repository named remote-assignments in your GitHub account.
2. Create folders in this repository for assignments of each week.
   a. Create folder Week-n for week N. (N is an integer)
   b. Under folder Week-n, create folder Assignment-M for all the files you like to submit in assignment M (M is an integer)
3. Manage and submit your work to GitHub by any Git tool.
4. For example of Week-1, Assignment-2:
   a. Your **Github Page link** should look like:
      https://username.github.io/remote-assignments/Week-1/Assignment-2/
   b. Your **Github repository** should look like:

Reminder: this assignment is very important for coming tasks in the next 3 weeks. You have to complete it anyway!
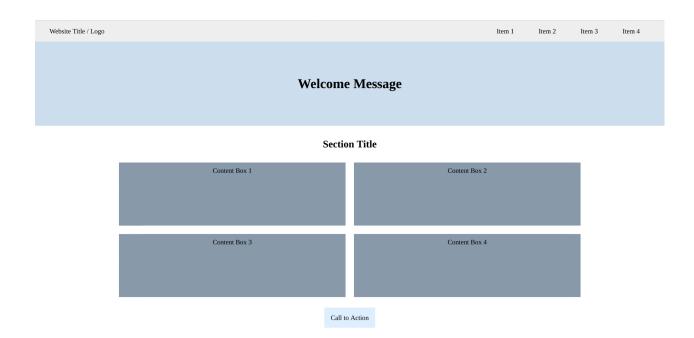
Assignment 2: Page Layout in pure HTML/CSS
Apply any HTML, CSS techniques <u>without</u> Bootstrap, JQuery or other libraries to accomplish following page layout requests.

Basic Principle: You can use Chrome Device Mode to get detail information about the web page.

Key Points:
1. Website Title / Logo in the navigator should align to the left.
2. Menu Items in the navigator should align to the right.
3. Total width of content boxes should be fixed in 1200px when screen width > 1200px.

Website Title / Logo		Item 1	Item 2	Item 3	Item 4

Welcome Message

Section Title

Content Box 1	Content Box 2
Content Box 3	Content Box 4

Call to Action

Assignment 3: Python Basic
Your job is to complete the following two functions.

1. find_max: find the max value of an array of numbers.
2. find_position: find the first position of the target number inside an array of numbers. The position should be counted starting from 0, if you can't find the target, please return -1.

Reminder: you cannot use those built-in functions like max() and index() to complete this assignment, please implement it by yourself.

```python
def find_max(numbers):
    # your code here

def find_position(numbers, target):
    # your code here

print(find_max([1, 2, 4, 5]) ); # should print 5
print(find_max([5, 2, 7, 1, 6]) ); # should print 7

print(find_position([5, 2, 7, 1, 6], 5)) # should print 0
print(find_position([5, 2, 7, 1, 6], 7)) # should print 2
print(find_position([5, 2, 7, 7, 7, 1, 6], 7)) # should print 2 (the first one)
print(find_position([5, 2, 7, 1, 6], 8)) # should print -1
```

Assignment 4: Algorithm Practice (Advanced Optional)

We created a function in Assignment 3 which can find the position of the target number inside an array of numbers. Actually, if the array was **Sorted** already, there is a beautiful algorithm called **Binary Search** which can do this job efficiently. You can try to lookup these keywords and learn the concept behind this algorithm and write the code by yourself.

For simplicity, you can assume that there are no duplicate numbers in the given array. It could be a bit of challenge if you haven't learned any algorithm before.

```python
def binary_search_position(numbers, target):
    # your code here

print(binary_search_position([1, 2, 5, 6, 7], 1)) # should print 0
print(binary_search_position([1, 2, 5, 6, 7], 6)) # should print 3
```