week 1 assignment

Git & Github

2.

git status : Show the working tree status. To see any files not be tracked by git or in staging area need to be committed.

git add : Add file contents to the index. Add changed files from working directory to staging area.

git commit : Record change to the repository. Use the files in staging area to create new commit.

git log : Show commit log. To see the history of the repo with author name, author email, date, and commit message.

git push [Repo_name] [Branch_name] : Update remote refs along with associated objects. To push your repo from local to remote.

git remote -v : List all the remotes that are associated with the current repo.

git branch : List, create, or delete branches. To see all the branches and which branch now we work on or create new branch. -d to delete the branch.

fork : Create a completely independent copy of git repo. If your want to get the full control of the repo and any changes and updates to the forked repo will not reflect the original repo.

3. Click + on the top right corner, select “New repository”, input repo name and other information, and click “Create repository”, and use `git remote add` and `git push` to upload the local project to the repo you just create on GitHub.

Basic

1.

```
let Pi = 3.14
```

2.

```
let x = 10, y = 20
let average = (x + y) / 2
```

3.

```
let averageDouble = Double(average)
```

(10/3) use 2 int to divide, so we can only get int value 3.

(10.0/3.0) use 2 double value to divide, so we can get double value 3.33333

4.

```
var flag: Bool = true
var inputString: String = "Hello world."
let bitsInBite: Int = 8
let averageScore: Double = 86.8
```

5.

```
var salary = 22000
salary += 28000
```

6.

`==`

7.

```
let a = 10, b = 3
let remainder = a % b
```

8. var for variables and let for constants, var can change the value when you want, but let can’t change the value.
9. Use meaningful name, and use capital letter to separate each word.
10. If you don’t specify the type of value, swift will automatically choose the type fitting the value you set.

11. First line we assign phoneNumber as Int value, so we will get error to assign string value at second line.

Collection

1.

```
var myFriends: [String] = []
```

2.

```
myFriends += ["Ian", "Bomi", "Kevin"]
```

3.

```
myFriends.append("Michael")
```

4.

```
myFriends.remove(at: 2)
myFriends.insert("Kevin", at: 0)
```

5.

```
for friend in myFriends{
    print(friend)
}
```

6. **Fatal error: Index out of range.**

Use `myFriends[5]` to get index 5 in array, but you only have 4 items in your array(0-3), so you will only get the error message.

7.

```
myFriends.first
```

8.

```
myFriends.last
```

9.

```
var myCountryNumber: [String:Int] = [:]
```

10.

```
myCountryNumber["US"] = 1
myCountryNumber["GP"] = 44
myCountryNumber["JP"] = 81
```

11.

```
myCountryNumber["GP"] = 0
```

12.

```
var emptyDictionary: [String:Int] = [:]
```

13.

```
dict[key] = nil
```

Control flow

1.

```
for i in lottoNumbers.count-3..<lottoNumbers.count{
    print(lottoNumbers[i])
}
```

2.

```
for i in 5...10{
    print(i)
}
```

```
for i in stride(from: 10, through: 6, by: -2){
    print(i)
}
```

3.

```
var i = 5
while i <= 10{
    print(i)
    i += 1
}
```

```
var i = 10
while i >= 6{
    print(i)
    i -= 2
}
```

4.

```
var i = 5
repeat{
    print(i)
    i += 1
}while i <= 10
```

```
var i = 10

repeat{
    print(i)
    i -= 2
}while i >= 6
```

5. repeat-while will at least execute code block once.

6.

```
if isRaining{
    print("It's raining, I don't want to work today.")
}else{
    print("Although it's sunny, I still don't want to work today.")
}
```

7.

```
switch jobLevel{
  case 1:
     print("Member")
  case 2:
    print("Team Leader")
  case 3:
    print("Manager")
  case 4:
    print("Director")
  default:
    print("We don't have this job")
}
```

Function

1.

```
func greet(person:String){
    print("Hello, \(person)")
}
```

2.

```
func multiply(a: Int, b: Int = 10){
        print(a * b)
}
```

3. The argument label is used when calling the function; each argument is written in the function call with its argument label before it. The parameter name is used in the implementation of the function. By default, parameters use their parameter name as their argument label.
4. function birthday : String
   function payment : Double