

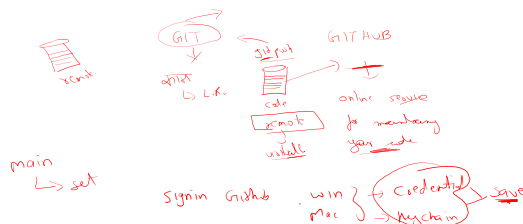
GITHUB } H.W-

WCA T → xccap
→ resume } H.W-
→ finish

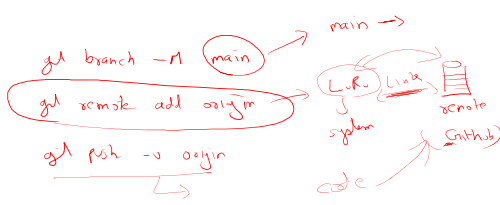
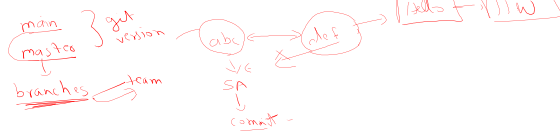
github.com/ create account



- 1) git add . / filename
- 2) git commit -m " " (with a note 'LR → Remote server (github)')
- 3) git push



git branch -M main } folders
-M main } location
3 other tasks



WCA T

- 1) -rs → spaces remove
- 2) -rd → new line remove
- 3) -rsc → special characters remove
- 4) -S → sequential
- 5) -sn →

1 Hello
2 My
3 Name

SN } non empty lines
1 Hell
2 I
3 A n
4 .
5 Abhijit

```
const fs = require('fs');
let arguments = process.argv.slice(2);

let flags = [];
let filenames = [];
let secondaryArguments = [];

for(let i of arguments) {
  if(i[0] == "-") {
    flags.push(i);
  } else if(i[0] == "s") {
    secondaryArguments.push(i.slice(1));
  } else {
    filenames.push(i);
  }
}
```

arguments }
File }
hol
hol
hol

```
for(let file of filenames) {
  let fileData = fs.readFileSync(file, 'utf-8');
  for(let flag of flags) {
    if(flag == "-rs") {
      fileData = removeAll(fileData, " ");
    }
    if(flag == "-rd") {
      fileData = removeAll(fileData, "\n");
    }
    if(flag == "-rsc") {
      for(let secondaryArgument of secondaryArguments) {
        fileData = removeAll(fileData, secondaryArgument);
      }
    }
    console.log(fileData);
  }
}
```

traverse file
\$ hol
\$ hol
\$ hol
\$ hol

1 Hello
2 I
3 am

1 Hello
2 I
3 Am

-rd → remove extra line

1 Hello
2 -
3 -
4 I am new here
5 -
6 what do you do

```
function removeExtraLine(fileData) {
  let contentArr = fileData.split("\n");
  let data = [];
  for(let i=0; i<contentArr.length; i++) {
    if(contentArr[i] == "" && contentArr[i+1] != "") {
      contentArr[i] = null;
    }
    if(contentArr[i] != "" && contentArr[i+1] == null) {
      contentArr[i] = null;
    }
  }
  for(let i=0; i<contentArr.length; i++) {
    if(contentArr[i] != null) {
      data.push(contentArr[i]);
    }
  }
  return data;
}
```

1 Hello
2 .
3
4 How
5
6
7
8
9 Are
10
11 go
12
13
14

[Hell, , How, , Are, , go, ,
↑ ↑ ↑ ↑ ↑
9 10 11 12 13
carr

Hell, → How, →
Are, → go, →