

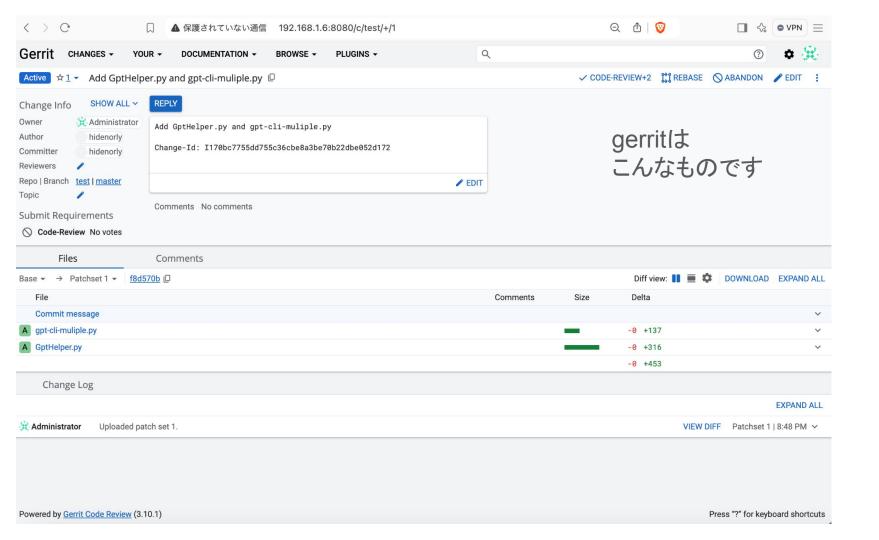
と流行のアレ

Anyone to know gerrit?

Anyone to use gerrit?

Which tool are you using for code review?





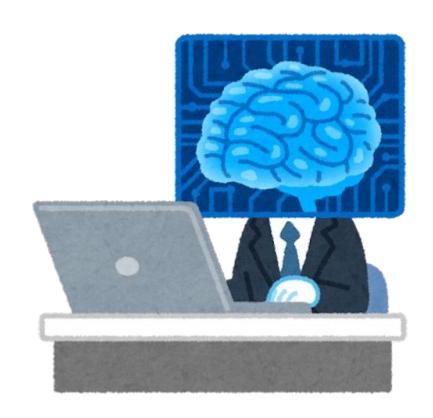
```
def read_prompt_json(path):
42
43
           system_prompt = ""
           user_prompt = ""
44
45
           result = {}
46
47
           if path and os.path.isfile(path):
     Draft
                                                                               8:50 PM ^
     Please change to whether the file exists or not.
     Ot should be changed to os.path.exists() instead.
    Resolved
                                                                           CANCEL SAVE
               with open(path, 'r', encoding='UTF-8') as f:
48
                 result = json.load(f)
49
                 if "system_prompt" in result:
50
51
                   system_prompt = result["system_prompt"]
52
                 if "user_prompt" in result:
                   user_prompt = result["user_prompt"]
53
54
55
           if system_prompt or user_prompt:
56
               return system_prompt, user_prompt
57
           else:
58
               return result, None
59
```

@staticmethod

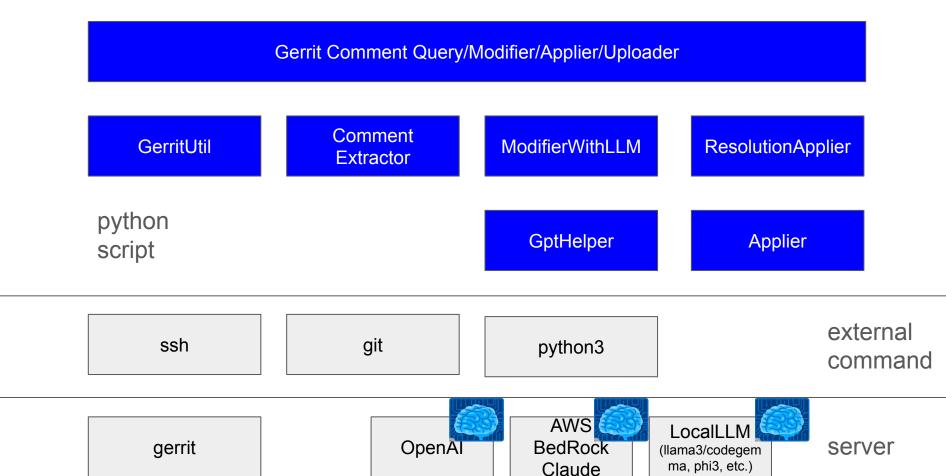
41

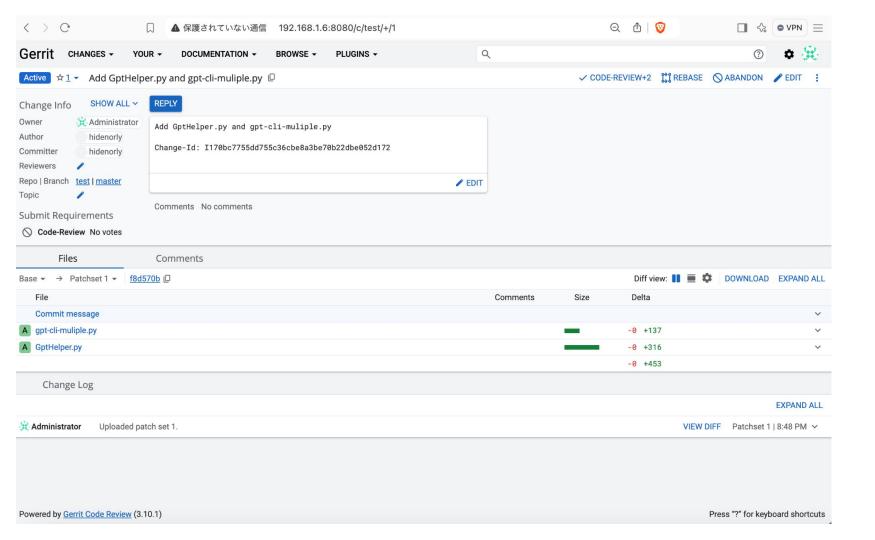
で、はやりのアレって?

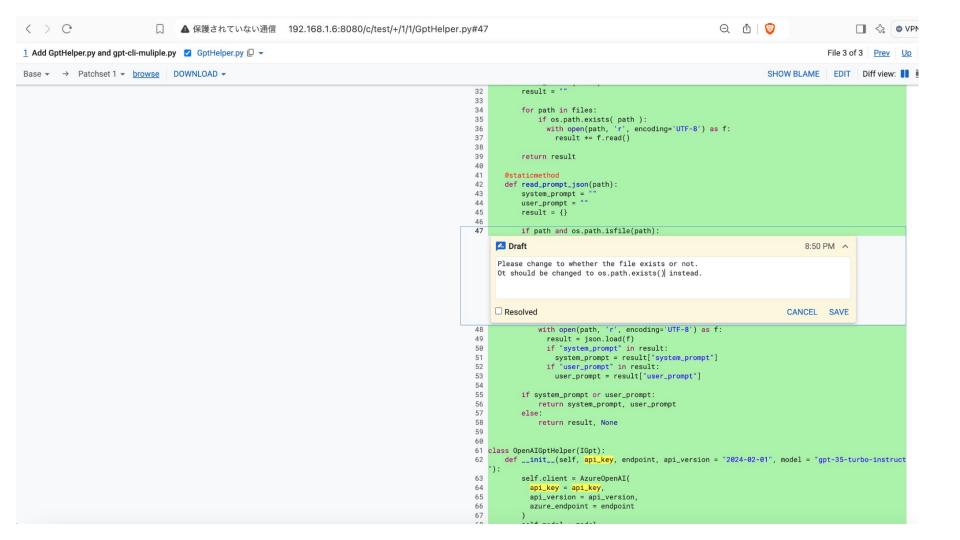
で、はやりのアレって?

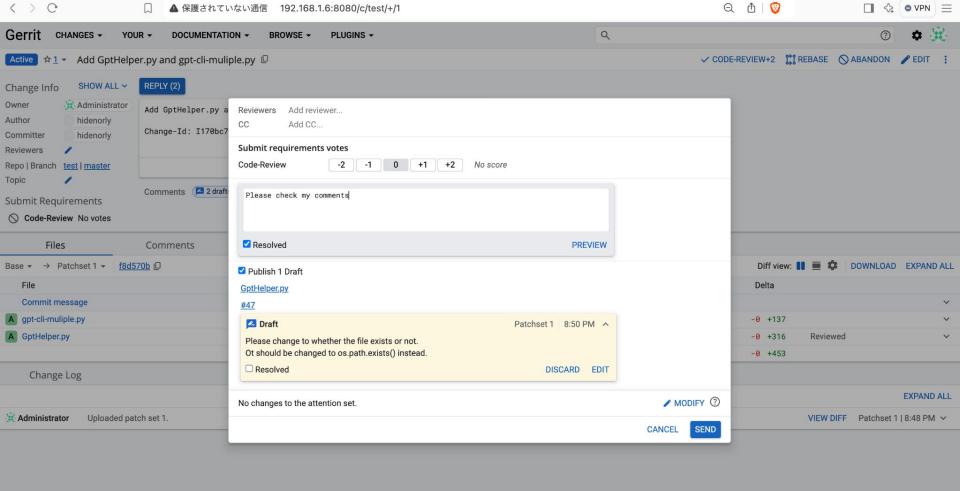


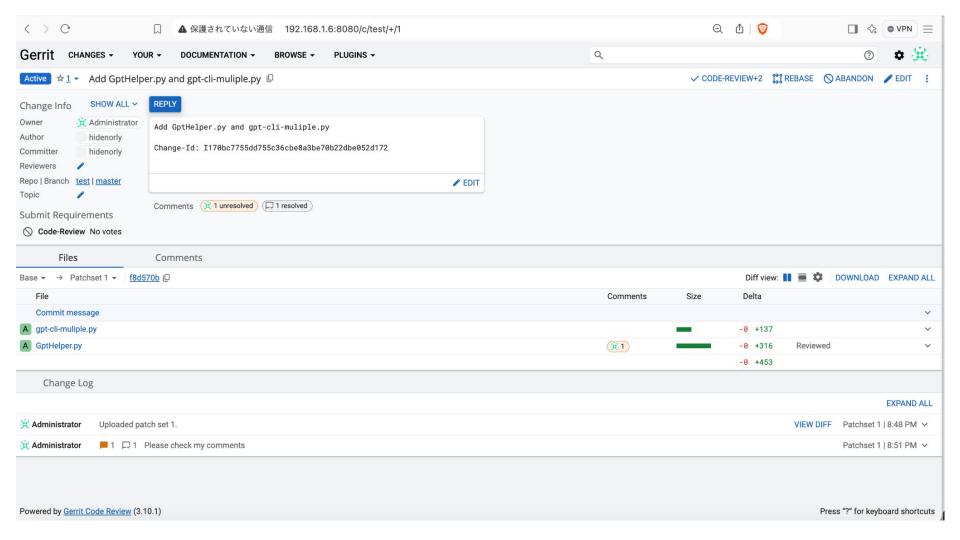
青箱が今回作った物











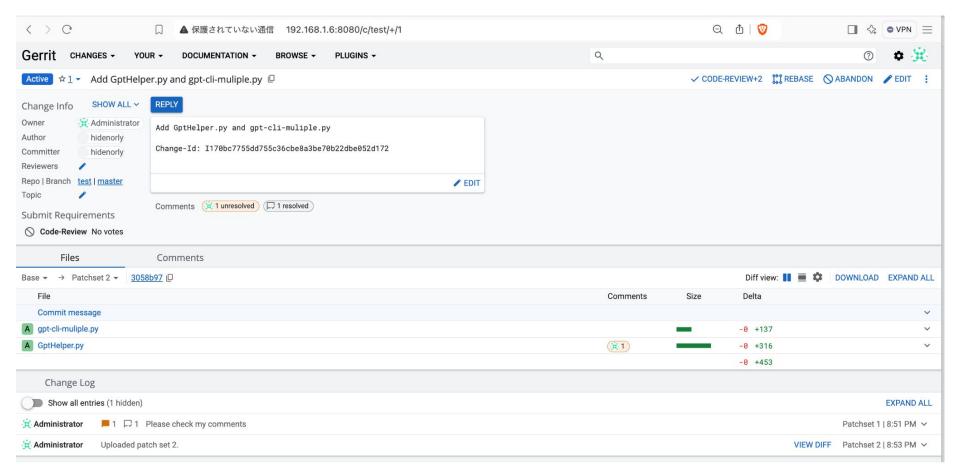
```
harold@m2mba gerrit-util % python3 gerrit_comment_modifier_applier.py -t gerrit -n 1 -r -a -m 5 -c -p git_comment_modifier.json --connection=ssh -u
project:test
branch:master
number:1
subject:Add GptHelper.py and gpt-cli-muliple.py_connect(SocketClient.java:243)
status: NEW
url:http://48b80fbef164/c/test/+/1r.mail.send.SmtpEmailSender.open(SmtpEmailSender.iava:407)
Created: 2024-09-30 20:48:20
project_dir:test
Last Updated:2024-09-30 20:51:14
patchset1_ssh:qit clone ssh://gerrit/test -b master; cd test; git pull ssh://gerrit/test refs/changes/1/1/1 --rebase in shooks HookFactory; hooks projectCreatedHook resolved to /var/gerrit/test
patchset1_repo:repo download test 1/1
comments:{'/PATCHSET_LEVEL': {0: [{'message': 'Please check my comments', 'reviewer': {'name': 'Administrator', 'email': 'admin@example.com', 'username': 'admin"}}]}, 'GptHelper.py': {47: [{'message': 'Please change to whether the file exists
or not.\nOt should be changed to os.path.exists() instead.'; 'reviewer': {'name': 'Administrator', 'email': 'admin@example.com', 'username': 'admin'}}]}}
current_patchset_ssh:git clone ssh://gerrit/test -b master; cd test; git pull ssh://gerrit/test refs/changes/01/1/1 --rebase
Cloning into 'test' ...
remote: Counting objects: 2, done
remote: Finding sources: 100% (2/2)
remote: Total 2 (delta 0), reused 0 (delta 0) ject="test" request="REST /changes/*/revisions/*/review"
Receiving objects: 100% (2/2); done. receive-pack /test (admin)] INFO com.google.gerrit.server.git.MultiProgressMonitor: Processing changes: refs: 1, updated: 1 [CONTEXT ratelimit period="1"]
remote: Counting objects: 5, done
remote: Finding sources: 100% (4/4)
remote: Total 4 (delta 0), reused 4 (delta 0)
Unpacking objects: 100% (4/4), 4.88 KiB | 1.63 MiB/s, done.
From ssh://gerrit/test
 * branch
                  refs/changes/01/1/1 -> FETCH_HEAD
Updatina 1c0db8c..f8d570b
Fast-forward
GptHelper.pv
                  1 316
apt-cli-muliple.pv | 137
2 files changed, 453 insertions(+)
create mode 100644 GptHelper.py
create mode 100644 apt-cli-muliple.py
GptHelper.py
absolute_pos=47:comment=Please change to whether the file exists or not.
Ot should be changed to os.path.exists() instead.:the_line=
                                                              with open(path, 'r', encoding='UTF-8') as f:
relative_pos=5
Please check the following reviewer's comment and please change the code to satisfy with the reviewer's mention. And please output the modified code.
Reviewer's comment:
Please change to whether the file exists or not.
Ot should be changed to os.path.exists() instead.
Mentioned code (the comment is intended to the #5 line):
       system_prompt = ""
      user_prompt = ""
       result = {}
       if path and os.path.isfile(path):
          with open(path, 'r', encoding='UTF-8') as f:
            result = json.load(f)
            if "system_prompt" in result:
             system_prompt = result["system_prompt"]
            if "user_prompt" in result:
```

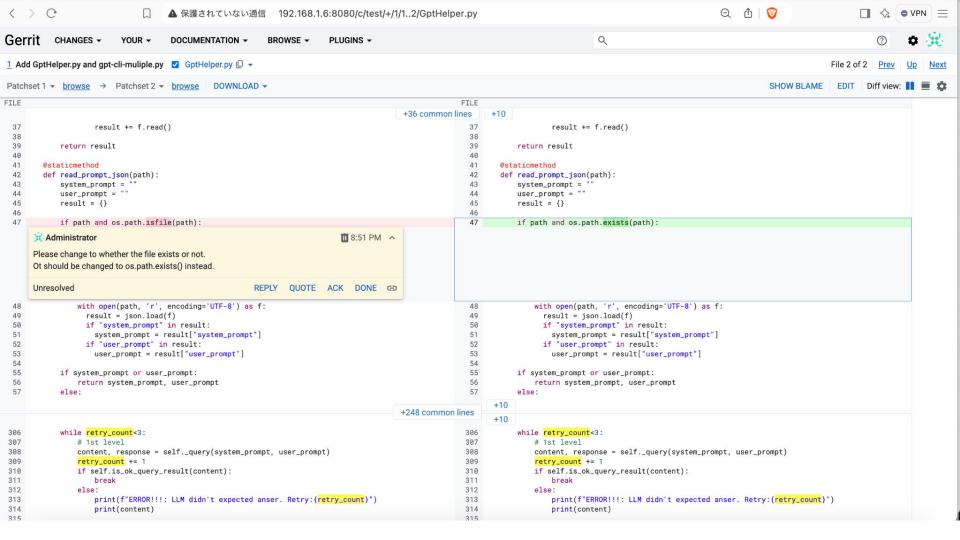
```
GptHelper.py
absolute_pos=47:comment=Please change to whether the file exists or not.
Ot should be changed to os.path.exists() instead.:the_line=
                                                                      with open(path, 'r', encoding='UTF-8') as f:
relative_pos=5
Please check the following reviewer's comment and please change the code to satisfy with the reviewer's mention. And please output the modified code.
Reviewer's comment:
Please change to whether the file exists or not.
Ot should be changed to os.path.exists() instead.
Mentioned code (the comment is intended to the #5 line):
       system_prompt = ""
       user_prompt = ""
       result = {}
       if path and os.path.isfile(path):
           with open(path, 'r', encoding='UTF-8') as f:
             result = json.load(f)
             if "system_prompt" in result:
               system_prompt = result["system_prompt"]
             if "user_prompt" in result:
Here's the modified code with the change requested by the reviewer:
```python
 system_prompt = ""
 user_prompt = ""
 result = {}
 if path and os.path.exists(path):
 with open(path, 'r', encoding='UTF-8') as f:
 result = json.load(f)
 if "system_prompt" in result:
 system_prompt = result["system_prompt"]
 if "user_prompt" in result:
The change is on line 5, where `os.path.isfile(path)` has been replaced with `os.path.exists(path)`. The `os.path.exists()` function checks if the given path (file or directory) exists, wher
eas `os.path.isfile()` specifically checks if the given path is a file. By using `os.path.exists()`, the code will work for both files and directories, which may be more flexible in case the
path refers to a directory instead of a file.
```

```
else:
 print(f"ERROR!!!: LLM didn't expected anser. Retry:{retry_count}")
 print(content)
 return content, response
[master 3058b97] Add GptHelper.py and gpt-cli-muliple.py
 Date: Mon Sep 30 20:48:15 2024 +0900
 2 files changed, 453 insertions(+)
 create mode 100644 GptHelper.py
 create mode 100644 gpt-cli-muliple.py
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 4.91 KiB | 4.91 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Processing changes: refs: 1, updated: 1, done
remote:
remote: SUCCESS
remote:
 http://48b80fbef164/c/test/+/1 Add GptHelper.py and gpt-cli-muliple.py
remote:
remote:
To ssh://gerrit/test
```

HEAD -> refs/for/master

[new reference]





```
Gerrit comment AI helper
options:
 -h, --help
 show this help message and exit
 -t TARGET, --target TARGET
 Specify ssh target host
 -b BRANCH, --branch BRANCH
 Branch to query
 -s STATUS, --status STATUS
 Status to query (mergedlopen)
 Since when to query
 --since SINCE
 -n NUMBERS, --numbers NUMBERS
 Specify gerrit numbers with ,
 -w DOWNLOAD, --download DOWNLOAD
 Specify download path
 Specify if re-download anyway
 -r, --renew
 -m MARGINLINE, --marginline MARGINLINE
 Specify margin lines
 specify if you want to use calude3
 -c, --useclaude
 -g GPT, --gpt GPT
 specify openai or calude3 or openaicompatible
 -k APIKEY, --apikey APIKEY
 specify your API key or set it in AZURE_OPENAI_API_KEY env
 -y SECRETKEY, --secretkey SECRETKEY
 specify your secret key or set it in AWS_SECRET_ACCESS_KEY env (for claude3)
 -e ENDPOINT, --endpoint ENDPOINT
 specify your end point or set it in AZURE_OPENAI_ENDPOINT env
 -d DEPLOYMENT, --deployment DEPLOYMENT
 specify deployment name or set it in AZURE_OPENAI_DEPLOYMENT_NAME env
 -p PROMPTFILE, --promptfile PROMPTFILE
 specify prompt.json
 -a, --apply
 Specify if apply the modification for the conflicted file
 Specify if upload the the conflict resolved result
 -u, --upload
```

usage: gerrit\_comment\_modifier\_applier.py [-h] [-t TARGET] [-b BRANCH] [-s STATUS] [--since SINCE] [-n NUMBERS] [-w DOWNLOAD] [-r] [-m MARGINLINE] [-c] [-g GPT] [-k APIKEY] [-y SECRETKEY]

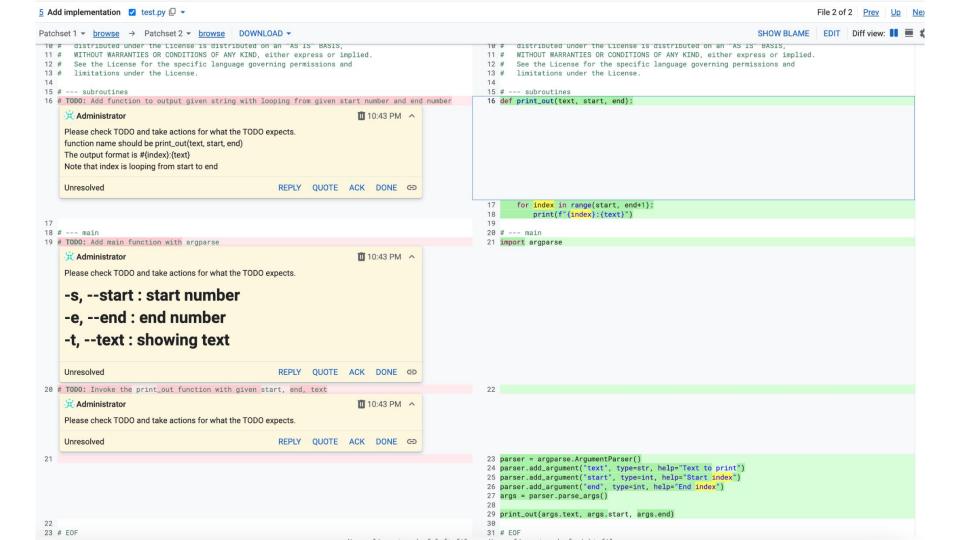
[-e ENDPOINT] [-d DEPLOYMENT] [-p PROMPTFILE] [-a] [-u]

```
△ 保護されていない通信 192.168.1.6:8080/plugins/gitiles/test/+/refs/changes/01/1/2/GptHelper.py
 result = ""
 32
 33
 for path in files:
 34
 if os.path.exists(path):
 35
 with open(path, 'r', encoding='UTF-8') as f:
 36
 result += f.read()
 37
 38
 return result
 39
 40
 @staticmethod
 41
 def read_prompt_json(path):
 42
 43
 system_prompt = ""
 user_prompt = ""
 44
 result = {}
 45
 46
 if path and os.path.exists(path):
 47
 with open(path, 'r', encoding='UTF-8') as f:
 48
 result = json.load(f)
 49
 if "system_prompt" in result:
 50
 system_prompt = result["system_prompt"]
 51
 if "user_prompt" in result:
 52
 user_prompt = result["user_prompt"]
 53
 54
 55
 if system_prompt or user_prompt:
 return system_prompt, user_prompt
 56
 else:
 57
 return result, None
 58
 59
```

```
diff --git a/GptHelper.py b/GptHelper.py
index 7a0edc7..f4b2adf 100644
--- a/GptHelper.py
+++ b/GptHelper.py
@@ -44,7 +44,7 @@ class IGpt: hooks.HookFactory : hooks.projectCreatedHook resolved to /var/gerr
 user_prompt = ""
 result = googlesource.gerrit.plugins.hooks.HookFactory : hooks.commitReceivedHook re
 if path and os.path.isfile(path):
 if path and os.path.exists(path):
 with open(path, 'r', encoding='UTF-8') as f:
 result = json.load(f)
 if "system_prompt" in result:
@@ -313,4 +313,4 @@ class GptQueryWithCheck:
 print(f"ERROR!!!: LLM didn't expected anser. Retry:{retry_count}")
 print(content)
```

```
Unless required by applicable law or agreed to in writing, software
 distributed under the License is distributed on an "AS IS" BASIS,
10 #
 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
11 #
12 #
 See the License for the specific language governing permissions and
13
 limitations under the License.
14
 # --- subroutines
16 # TODO: Add function to output given string with looping from given start number and end number
 Draft
 10:42 PM ^
 Please check TODO and take actions for what the TODO expects.
 function name should be print_out(text, start, end)
 The output format is #{index}:{text}
 Note that index is looping from start to end
 Resolved
 DISCARD
 EDIT
17
 # --- main
19 # TODO: Add main function with argparse
 Draft
 10:42 PM A
 Please check TODO and take actions for what the TODO expects.
 -s, --start : start number
 -e, --end: end number
 -t, --text : showing text
 Resolved
 DISCARD
 EDIT
20 # TODO: Invoke the print_out function with given start, end, text
 Draft
 10:42 PM ^
 Please check TODO and take actions for what the TODO expects.
 Resolved
 DISCARD
 EDIT
21
22
```

23 # EOF

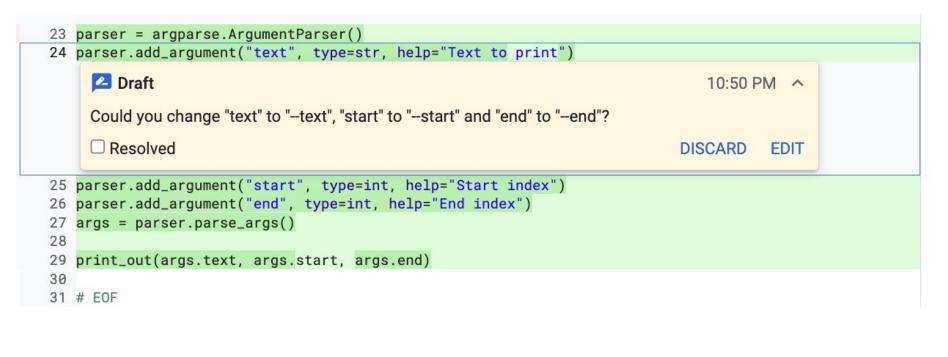


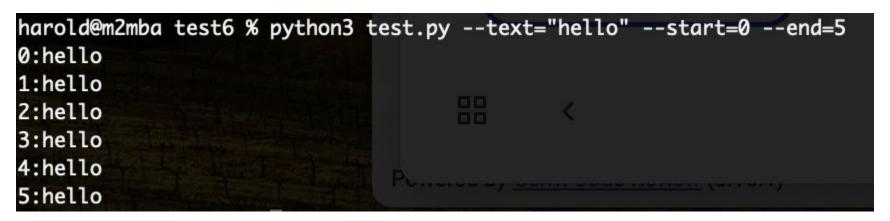
```
[harold@m2mba test6 % python3 test.py --text="hello" --start=0 --end=5
usage: test.py [-h] text start end
test.py: error: the following arguments are required: text, start, end
[harold@m2mba test6 % python3 test.py text="hello" start=0 end=5
usage: test.py [-h] text start end
test.py: error: argument start: invalid int value: 'start=0'
[harold@m2mba test6 % python3 test.py "hello" 0 5
0:hello
1:hello
2:hello
3:hello
```

Powered by Gerrit Code Review (3.10.1)

4:hello

5:hello





## Call for actions

- Please use it and feedback it
- Please contribute your improvement (pull req)
- Let's develop the gerrit plug-in together.
- Let's use & improve gerrit!