

# Functions

?????????

```
def _gpt_parse_images(
    image_infos: List[Tuple[str, List[str]]],
    prompt_dict: Optional[Dict] = None,
    output_dir: str = './',
    api_key: Optional[str] = None,
    base_url: Optional[str] = None,
    model: str = 'gpt-4o',
    verbose: bool = False,
    gpt_worker: int = 1,
    **args
) -> str:
    """
    Parse images to markdown content.
    """
```

## Print and Log

```
def print_f(*msg):
    '''print and log!'''
    # import datetime for timestamps
    import datetime as dt
    # convert input arguments to strings for concatenation
    message = []
    for m in msg:
        message.append(str(m))
    message = ' '.join(message)
    # append to the log file
    with open('/tmp/test.log','a') as log:
        log.write(f'{dt.datetime.now()} | {message}\n')
    # print the message using the copy of the original print function to stdout
    print(message)
```

```
print_f('Test Message')
```

## Sendmail via SMTP

```
def send_message(body, subject, to_addr):
    import smtplib
    from email.message import EmailMessage
    smtp_user = "your-smtp-user"
    smtp_pass = "your-smtp-pass"
    smtp_server = "smtp-relay.your.server"
    smtp_port = "587"

    msg = EmailMessage()
    msg['Subject'] = subject
    msg['From'] = smtp_user
    msg['To'] = to_addr
    msg.set_content(body)

    with smtplib.SMTP(smtp_server, smtp_port) as smtp:
        smtp.login(smtp_user, smtp_pass)
        smtp.send_message(msg)

debug = send_message("This is plain TEXT email", "Test from SMTP", "alang.hsu@gmail.com")
print(debug)
```

## Check Disk Usage

```
import shutil
import sys

def check_disk_usage(disk, min_absolute, min_percent):
    """Returns True if there is enough free disk space, false otherwise."""
    du = shutil.disk_usage(disk)
    # Calculate the percentage of free space
    percent_free = 100 * du.free / du.total
    # Calculate how many free gigabytes
    gigabytes_free = du.free / 2**30
    if percent_free < min_percent or gigabytes_free < min_absolute:
        return False
    return True
```

```
# Check for at least 2 GB and 10% free
if not check_disk_usage("/", 2, 10):
    print("ERROR: Not enough disk space")
    sys.exit(1)

print("Everything ok")
sys.exit(0)
```

## Check Internet

```
import socket

def check_no_network():
    """Returns True if it fails to resolve Google's URL, False otherwise."""
    try:
        socket.gethostbyname("www.google.com")
        return False
    except:
        return True
```

---

Revision #5  
Created 6 February 2023 17:35:37 by Admin  
Updated 2 January 2025 14:10:43 by Admin