how to add a feature

1. The first thing you need to do is create the feature .A feature is a python function that follows a set of rules . The function must get the request / response object . Optionally you can have up to 2 more user inputs . If you want to get user input just add it as a parameter to the function . The function must return the object after the changes . save the function in features.py . See example in the image below.

```
def find_and_replace(obj_responce, option1, option2):
    """
    search in the html for a key word. (option 1)
    if the work is found replace it with the other work (option 2)
    input : the response object , the keyword to change (option 1) and the keyword to change to (option 2)
    output : the response object
    """
    if "text/html" in obj_responce.content_type and type(obj_responce.content) == type(
            "".encode(encoding='UTF-8')):
        obj_responce.content = obj_responce.content.decode(encoding='UTF-8').replace(option1, option2).encode(
            encoding='UTF-8')
    return obj_responce
```

2. after making the function you need to add it to 2 more places . The 1st place is in the adblock.py . if the feature edits the request then add it to the REQUEST_FEATURES or the RESPONSE_FEATURES . the format of the features dictionaries are :

   “the name to call the feature”:the name of the function,

```
REQUEST_FEATURES = {  # your request feature come here <----------

}

RESPONSE_FEATURES = {  # your response feature come here <----------
    'css': css_feature,
    "inject img": inject_img,
    "block img by condition": block_img_by_condition,
    "find and replace": find_and_replace
}
```

3. The second place is in gui_adblocker.py . add the name you put in “ ” in the dictionary in the FEATURES list .

```
FEATURES = [
    # the list of filters for the proxy.
    # #to add a filter just add it here and make sure its in the right place in
    # adblock.py in the REQUEST_FEATURES or RESPONSE_FEATURES
    "none",
    "css",
    "inject img",
    "block img by condition",
    "find and replace"
]
```

That is it . you added your feature to the program :)

note : how the response / request object looks like .

```
class response:
    """
    A class used to represent a HTTP request
    ...

    Attributes
    ----------
    content :  bytes
        the content of the request .
    content_type : str
        the content type header in the HTTP response
    content_length: str
        the content length header  in the HTTP response
    response_code: str
        the response code of the request -
            200 - ok
            302 - redirect
            etc
    ---------

    Note: the response_code and content_length are str types that represent an int
    """
```