```
  # good to know:
  gcloud projects list
  gcloud beta billing accounts list

  gcloud config get-value project # output will be (unset)
if unset
  # NO PROBLEM IF UNSET: the --project option usually
solves all related issues

  # to set a project:
  gcloud config set project PROJECT_ID

  # project number is not project id
  gcloud beta billing projects link <project id> --billing-
account <your billing account id>

  # OK, if a main.py exists in .
  gcloud functions deploy <GCF name> --runtime python39 --
trigger-http --allow-unauthenticated --source . --entry-
point <python function name> --project <project id>

  # how to access?, via browser:
  url: https://<region>-<project id>.cloudfunctions.net/
<GCF name>

  # BAD - it will fail without a project set
  gcloud functions deploy <GCF name> --runtime nodejs14 --
trigger-http --allow-unauthenticated --entry-point <JS
function name>

  # SOL: state the project!
  gcloud functions deploy <GCF name> --runtime nodejs14 --
trigger-http --allow-unauthenticated --entry-point <JS
function name> --project <project id>

  --region=YOUR_REGION # not needed

  # how to access?, via browser:
  url: https://<region>-<project id>.cloudfunctions.net/
<GCF name>
```

```
{
    "name": "your-package-name",
    "version": "1.0.0",
    "description": "",
    "main": "my_js_with_gcfs.js",
    "scripts": {
      "start": "node my_js_with_gcfs.js"
    },
    "dependencies": {
    }
}


/*
this file is assumed to be named index.js
if another file name is desired, it should be explicitly
mentioned in a package.json file


The simplest package.json file (when using index.js) is:
{
   "name": "my-cloud-function",
   "version": "0.0.1"
}


If the file is (for example) named my_js_with_gcfs.js


{
   "name": "your-package-name",
   "version": "1.0.0",
   "description": "",
   "main": "my_js_with_gcfs.js",
   "scripts": {
     "start": "node my_js_with_gcfs.js"
    },
   "dependencies": {
    }
}
*/


/*
Google Cloud Functions in JS files, use Express.js
conventions.
Express.js is a web application framework for Node.js
The structure and properties of the req object are similar
to the request object in Express.js
```

```
*/


// this is using the arrow notation
exports.currentTime = (req, res) => {
    res.send(`The current time is: ${new
Date().toLocaleTimeString()}`);
};


// this is not using the arrow notation
// to deploy:
// gcloud functions deploy gcf_alt --runtime nodejs14 --
trigger-http --allow-unauthenticated --entry-point
gcfThatWillBeTriggeredByHttpCall --project <project id>
// notice that the deploy command does NOT state the source
file - it looks for it in the package.json file
function gcfThatWillBeTriggeredByHttpCall(
    p_request, // this param will bind to the HTTP request,
by the GCF PaaS
    p_response // this param will bind to the HTTP response,
by the GCF PaaS
){
    var now = new Date()


    var str_current_date = now.toDateString()
    var str_current_time = now.toLocaleTimeString()


    // backticks for str evaluation were introduced in ES6
(ECMAScript 2015)
    var str_with_evaluation_via_backticks = `Date: $
{str_current_date} | Time: ${str_current_time} (v2)`


    p_response.send(str_with_evaluation_via_backticks)
}//gcfThatWillBeTriggeredByHttpCall


exports.gcfThatWillBeTriggeredByHttpCall =
gcfThatWillBeTriggeredByHttpCall
```

```
   def hello_world(request):
       """Responds to any HTTP request.
       Args:
           request (flask.Request): HTTP request object.
       Returns:
           The response text or any set of values that can
be turned into a
           Response object using `make_response`
           https://flask.palletsprojects.com/en/1.1.x/api/
#flask.make_response
       """
       request_json = request.get_json(silent=True)
       request_args = request.args


       if request_json and 'name' in request_json:
           name = request_json['name']
       elif request_args and 'name' in request_args:
           name = request_args['name']
       else:
           name = 'World'
       return 'Hello {}!'.format(name)
   # def hello_world


   """
   gcloud functions deploy whatargs --runtime python39 --
trigger-http --allow-unauthenticated --entry-point
gcf_another --source . --project <project id>
   """


   def gcf_another(
       p_request
   ):
       # p_request.get_json(silent=True)
       request_args = p_request.args


       b_check:bool = request_args #and 'name' in
request_args
       if (b_check):
           str_response:str = "<html><body>"
           str_response+="<p>Here are the args I got:</p>"
           str_response+="<ul>"


           for arg in request_args:
               str_response+=f"<li>{arg}</li>"
```

```
        # for
        str_response+="</ul></body></html"
        return str_response
    else:
        return "No request arguments received!"
    # if-else
# def gcf_another
```