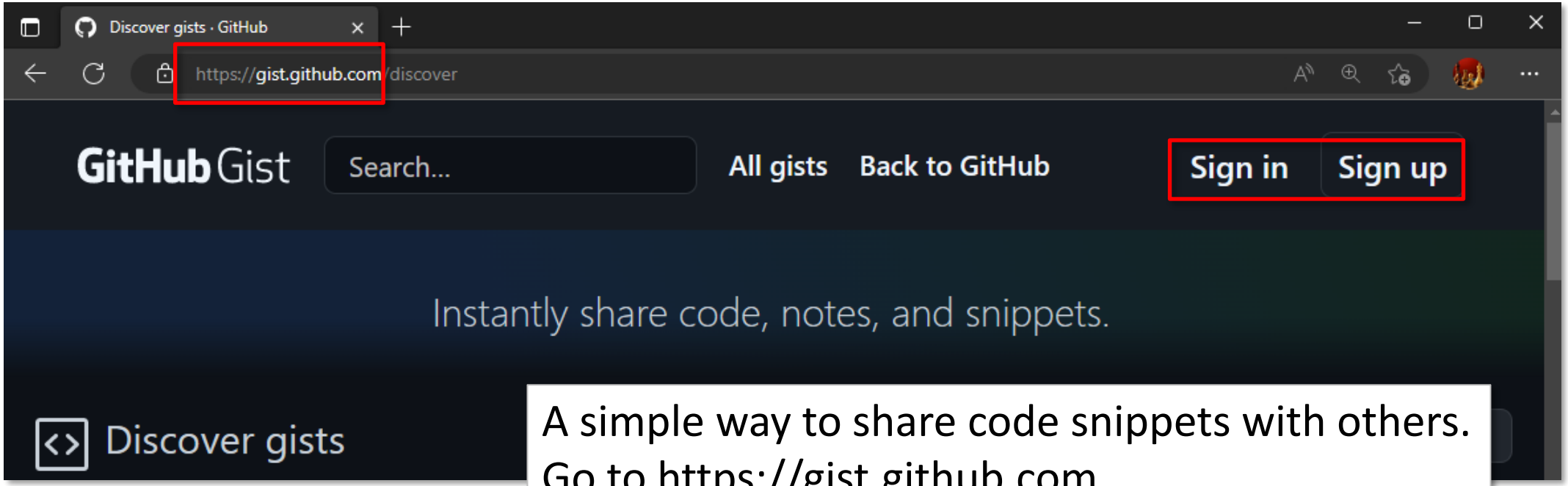
LAMBDA: ROBUSTNESS & REUSABILITY

GISTS

GETTING STARTED

GISTS

GETTING STARTED



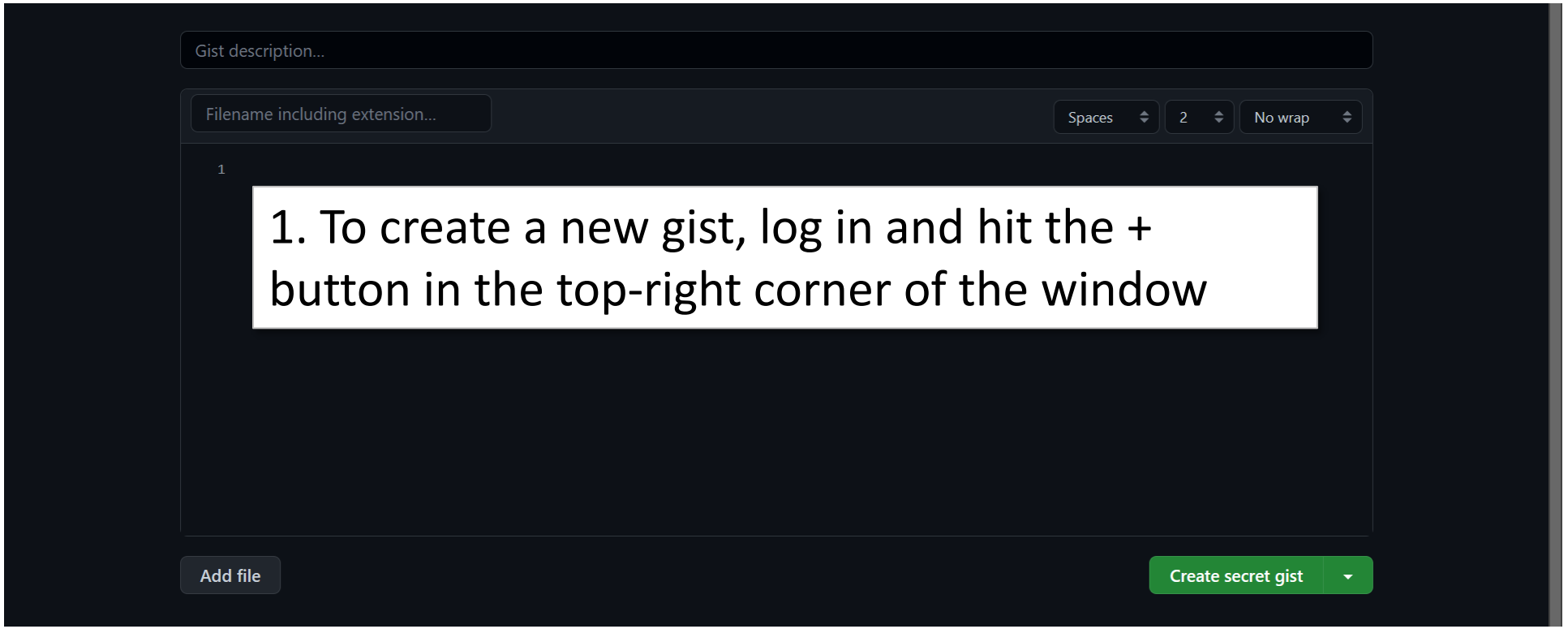
A simple way to share code snippets with others.
Go to https://gist.github.com
Sign in with a github account (or create one)

GISTS

CREATING A GIST

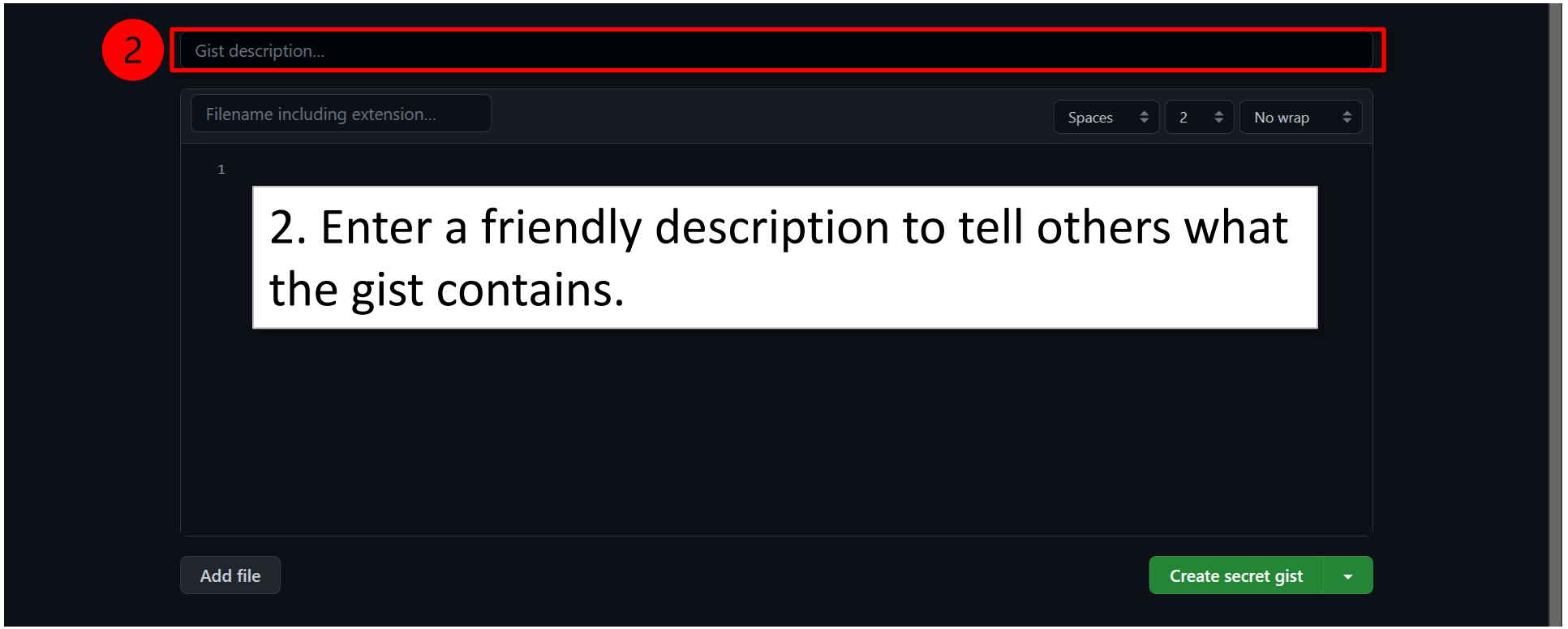
GISTS

CREATING A GIST



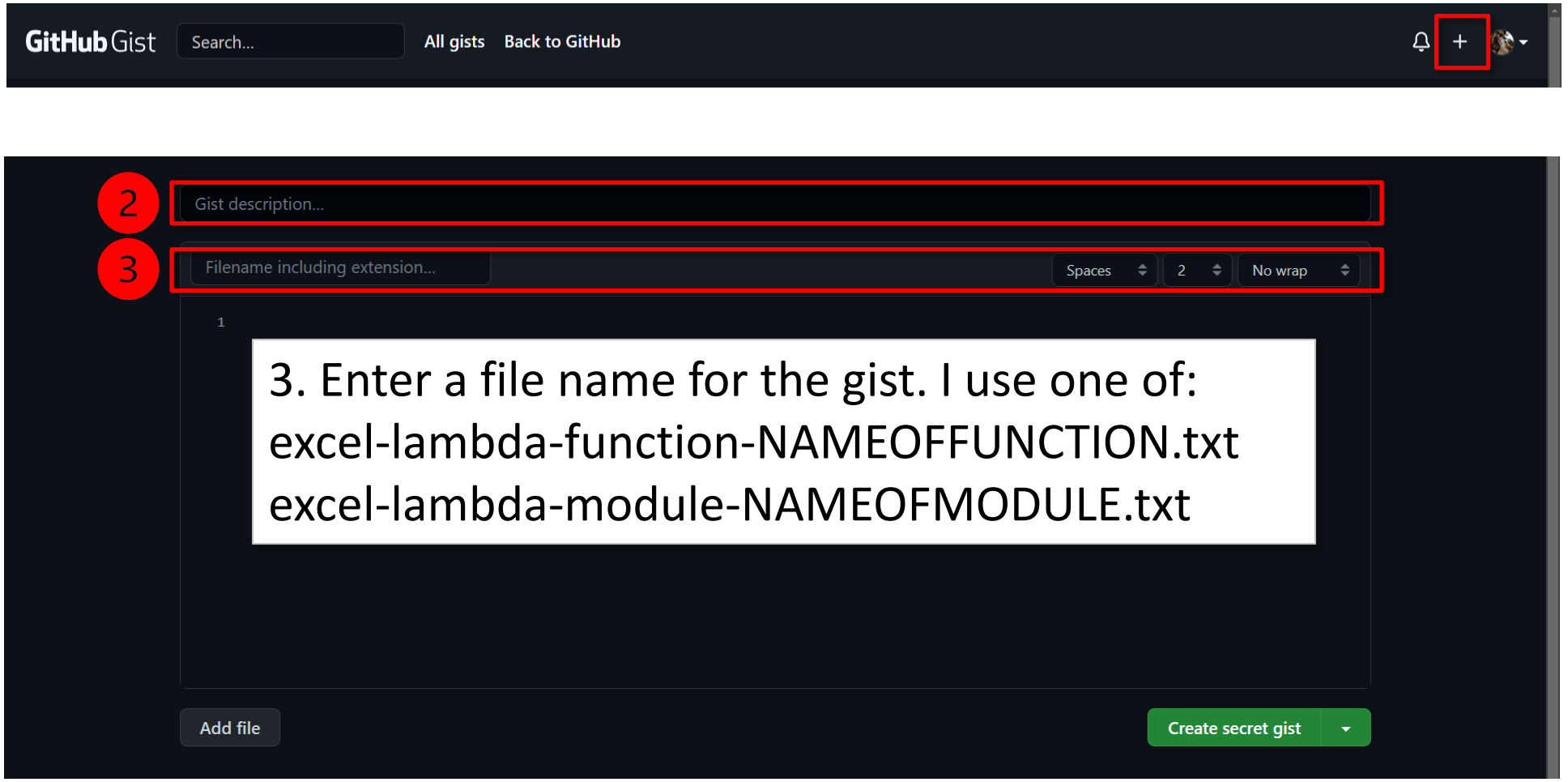
GISTS

CREATING A GIST



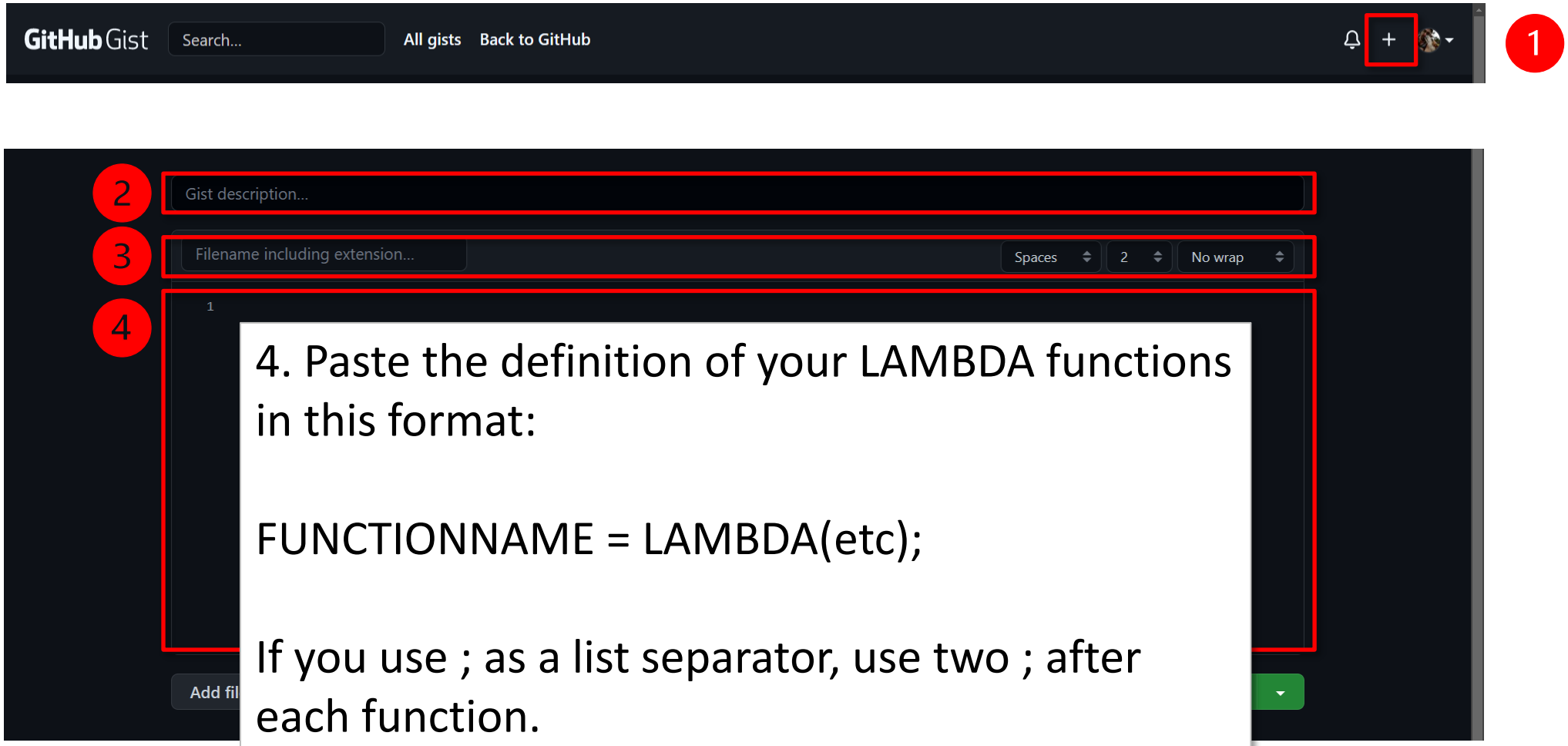
GISTS

CREATING A GIST



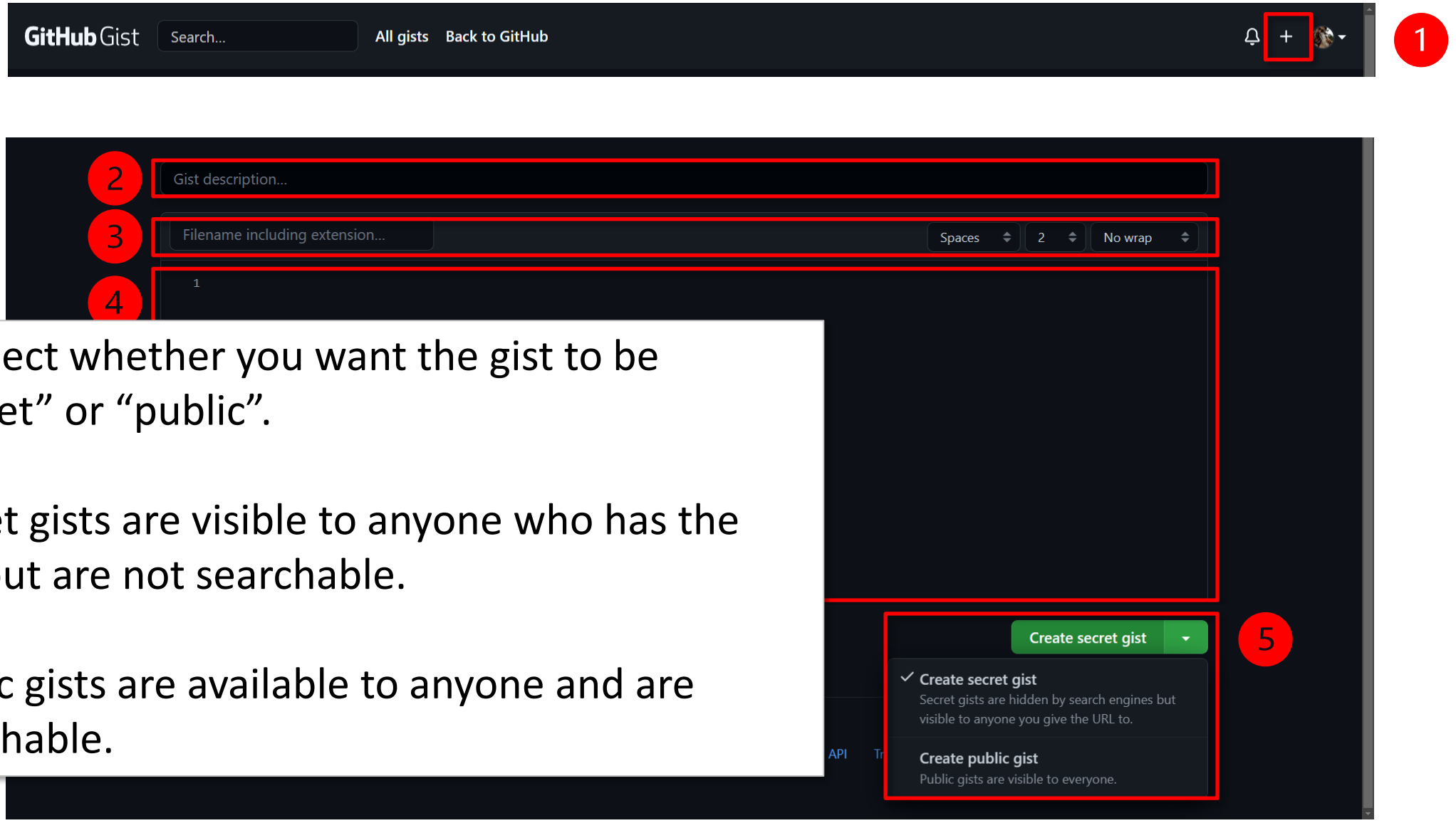
GISTS

CREATING A GIST



GISTS

CREATING A GIST



5. Select whether you want the gist to be “secret” or “public”.

Secret gists are visible to anyone who has the link but are not searchable.

Public gists are available to anyone and are searchable.

CREATING A GIST

DEMO

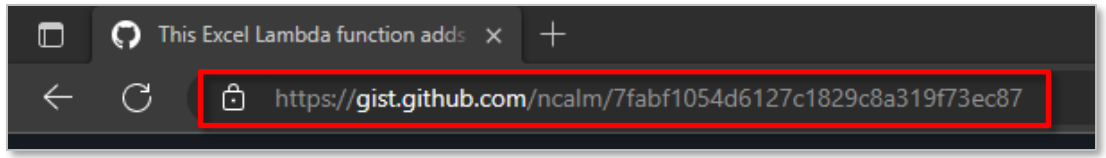
1. Creating a gist
2. Adding more than one file
3. Secret vs. Public

GISTS

USING A GIST

GISTS

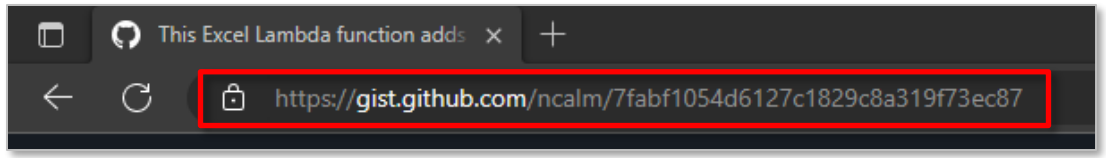
USING A GIST



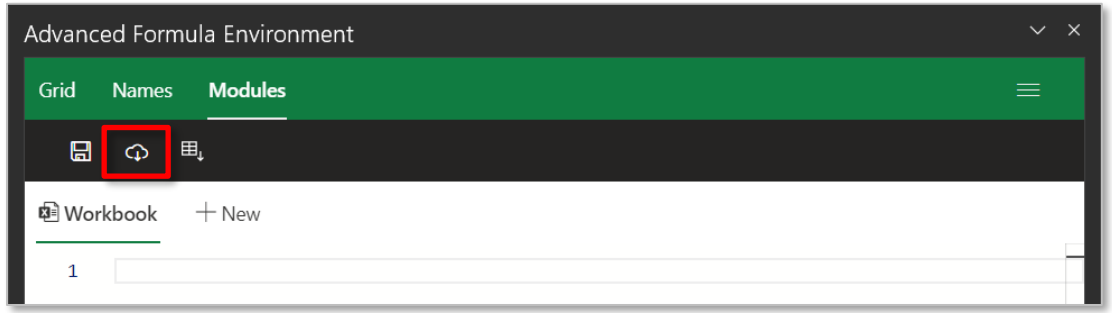
1. Copy the URL of the gist you want to use.

GISTS

USING A GIST



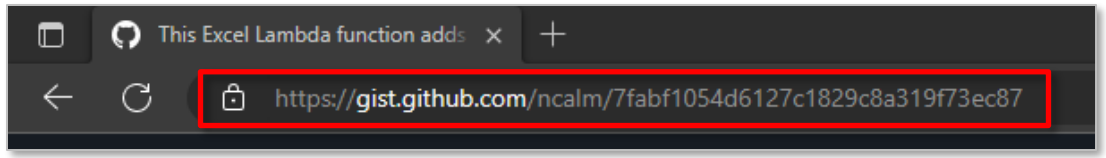
1. Copy the URL of the gist you want to use.



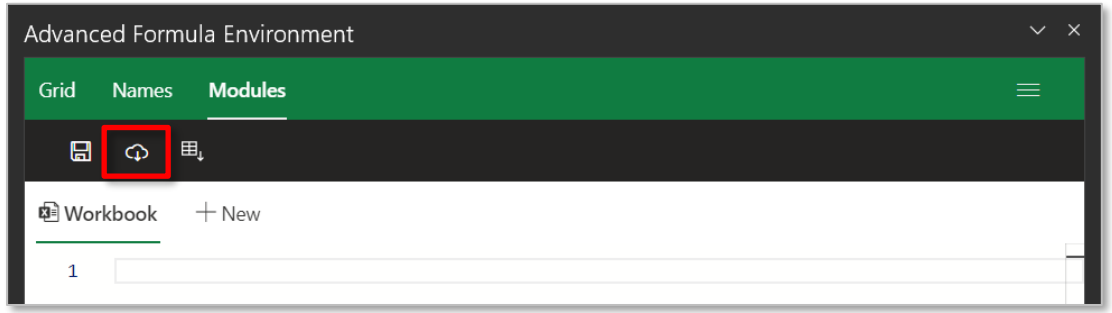
2. Use the “Import from URL” button in the Advanced Formula Environment

GISTS

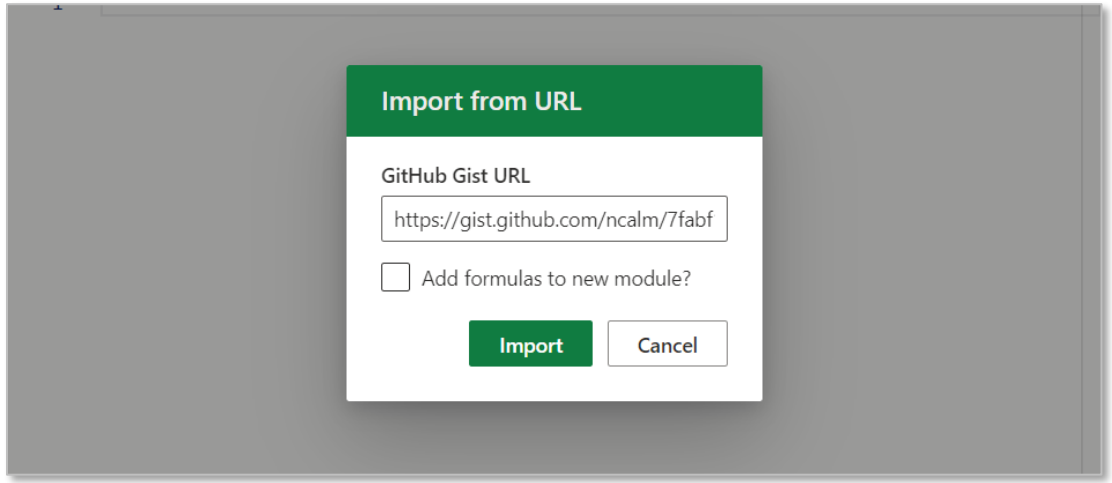
USING A GIST



1. Copy the URL of the gist you want to use.



2. Use the “Import from URL” button in the Advanced Formula Environment



3. Paste the URL and optionally add the formulas to a new module. If not selected, the gist will be imported into the currently active module.

USING A GIST

DEMO

1. Search of gists (user, filename)
2. Import to existing module
3. Import to new module
4. Import of multi-file gist
5. Import with conflict
6. Import of “malformed URL”

TAKEAWAYS

1. Use a consistent file naming convention
2. Only use one file in each gist
3. “Secret” does not mean “Private”

GISTS

REQUEST

Posted in Excel

OP • Just now • Submitted by you

Advanced Formula Environment: Allow import from any URL and from local text files

Currently the Advanced Formula environment only allows import from specifically formatted gist URLs. Any other URL is reported as "malformed or not supported".

This is a barrier to effective sharing and managing modules in repositories.

1. Allow import of text file contents from other URLs (e.g. directly from github repo, other website or a specific file URL within a multi-file gist).

2. Allow import of multi-file gists with option of having each file go to the same or separate modules.

3. Allow import of local text files (e.g. local branch of remote github repo) with an option to save changes to the local file. I believe import of local files was in the first release of AFE but has since been removed.

Thank you!

Open Windows Developer Tools

Please vote for expanded import capabilities in AFE:

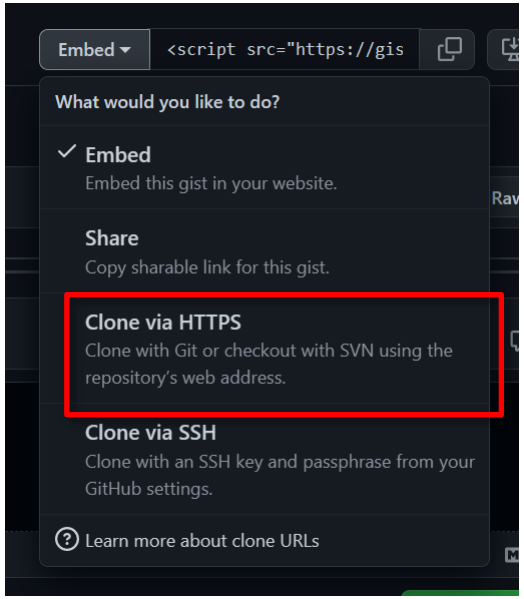
GISTS

USING GIT

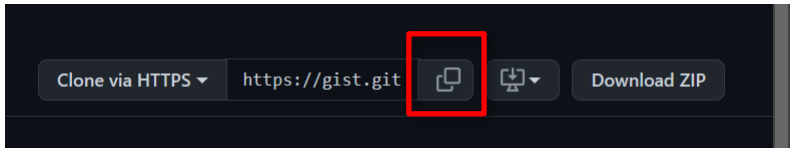
A gist is a git repository

GISTS

USING GIT



1. Select “Clone via HTTPS” from the “Embed” menu of a gist to get the repository URL



2. Copy the repository URL

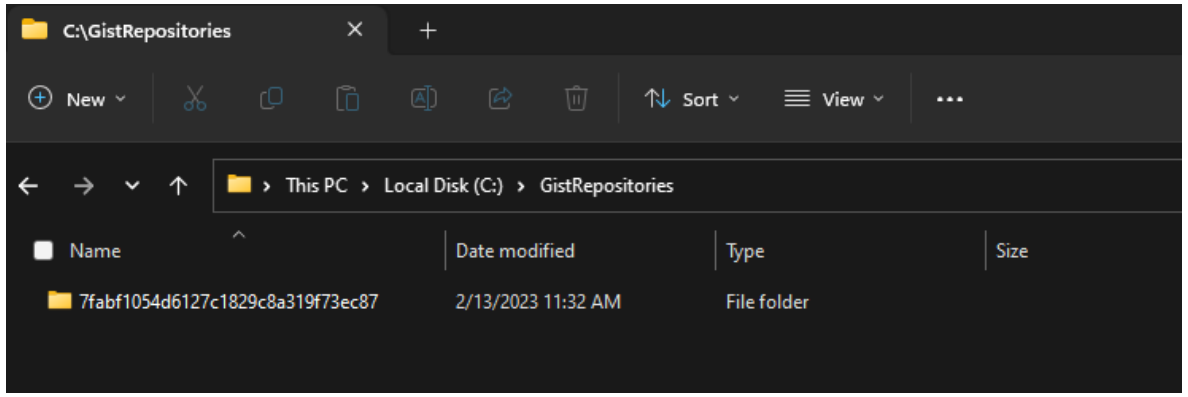
GISTS

USING GIT

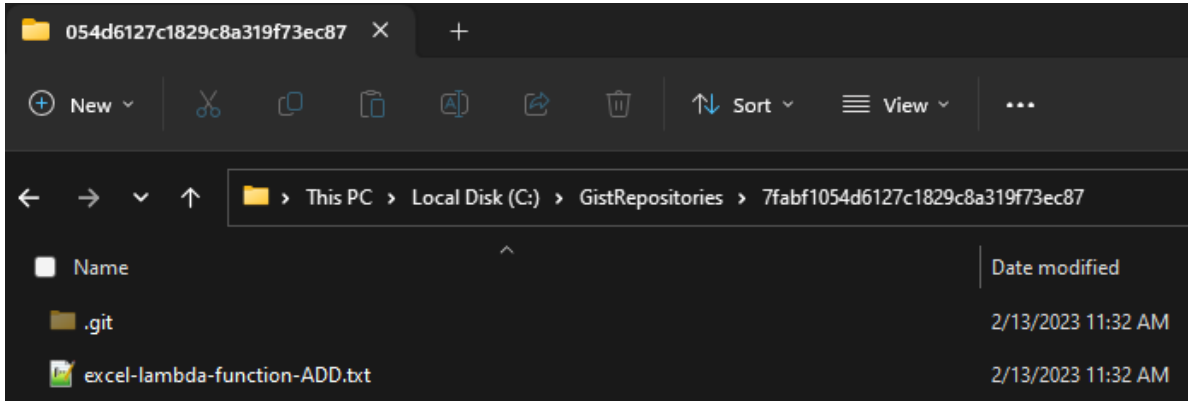
```
cd "C:\GistRepositories"

/c/GistRepositories
git clone https://gist.github.com/7fabf1054d6127c1829c8a319f73ec87.git
```

3 Using your preferred method of interacting with git (shown here is “git bash”), clone the gist to a folder on your computer



4 A sub-folder named with the gist identifier will be added to the folder



5 The file(s) in your gist are now on your computer

GISTS

USING GIT

DEMO

1. **Clone a gist**
2. **Edit an existing gist locally**
3. **Add a new file to a gist**
4. **Add, Commit, Push to remote**

GISTS

USING GIT

Git command	description
git clone <gist URL>	Creates a local copy of the gist on your computer
(make edits locally)	
git add .	After making changes to local file(s), adds the changed/added/deleted files to the working queue
git commit –m “message”	Commits the changes to the local repository
git push origin master	Pushes any differences between local and remote to the remote repository

TAKEAWAYS

1. Clone a gist to your local computer
2. Make edits locally
3. Push changes to the URL

Download for git bash

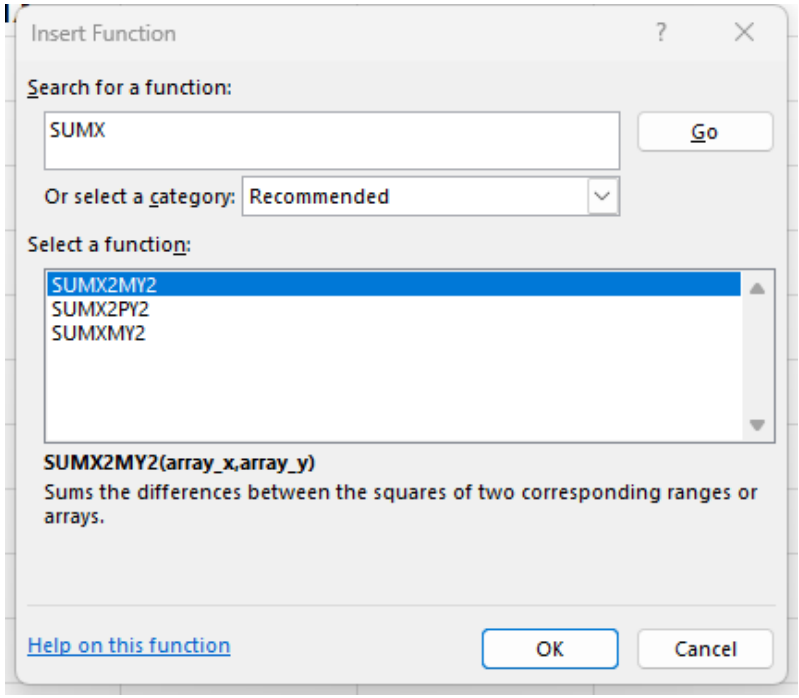
FUNCTION NAMES

NATIVE FUNCTIONS

FUNCTION NAMES

NATIVE FUNCTIONS

1. All Excel functions are capitalized
2. Punctuation is not used (except period e.g. *F.DIST.INV*)
3. Names don't conflict with references
4. Usually descriptive
5. Usually short
6. Not perfect



FUNCTION NAMES

LAMBDA FUNCTIONS

FUNCTION NAMES

LAMBDA FUNCTIONS

LAMBDA functions use the same syntax rules as Names

1. First character must be letter, underscore or backslash \
2. Remainder must be letters, numbers, periods or underscores
3. You can't use C, or R as a name
4. You can't use references
5. Spaces aren't allowed
6. Names aren't case-sensitive

```
1   //FIRST CHARACTER
2   //Yes:
3   MYFUNCTION = LAMBDA(1);
4   _MyFunction = LAMBDA(1);
5   \MYFUNCTION = LAMBDA(1);
6
7   //No:
8   1MYFUNCTION = LAMBDA(1);
9   .MYFUNCTION = LAMBDA(1);
10  [MYFUNCTION] = LAMBDA(1);
11
12  //REMAINING
13  //Yes:
14  MY.FUNCTION = LAMBDA(1);
15  _My.Function = LAMBDA(1);
16  \MY.FUNCTION = LAMBDA(1);
17
18  //No:
19  1MY\FUNCTION = LAMBDA(1);
20  .MY\FUNCTION = LAMBDA(1);
21  [MY\FUNCTION] = LAMBDA(1);
```

```
22
23  //Can't use C or R
24  C = LAMBDA(1);
25  R = LAMBDA(1);
26
27  //Can't use references:
28  A1 = LAMBDA(1);
29  AN1 = LAMBDA(1);
30  XFD1048576 = LAMBDA(1);
31
32  //Spaces aren't allowed
33  MY FUNCTION = LAMBDA(1);
34
35  //Names aren't case-sensitive
36  //So you can define:
37  COUNT = LAMBDA(1);
38
39  //But not this as well:
40  Count = LAMBDA(1);
```

DEMO

1. Duplication of native functions doesn't work
2. Grouping of like functions using a period

FUNCTION NAMES RECOMMENDATION

Function names:

1. Should be capitalized so they are easily distinguished from other names or LET variables
2. Should start with a letter so they are easy to type into the grid and other functions
3. Should use numbers sparingly and only to shorten the spelling of the number
4. Should use periods only to group like functions within a module (e.g. TEXT.SPLITTER)
5. Should not use underscores (mostly to make functions faster to type)
6. Should be meaningful. Don't use TEXT.SPLIT when only splitting on spaces. Use TEXT.SPLITONSPACE instead.

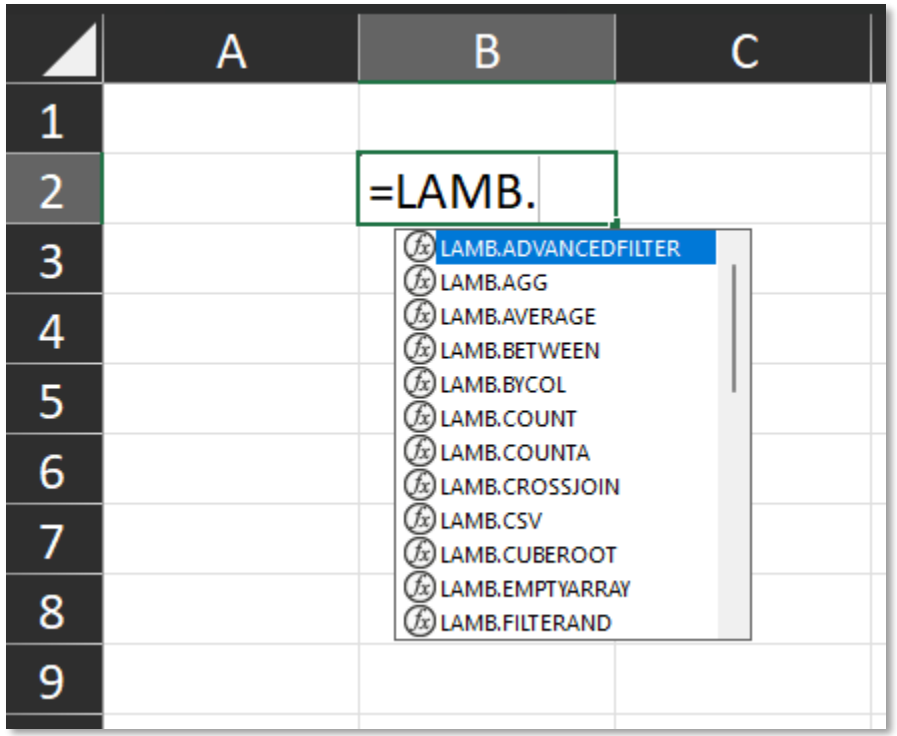
MODULES

WHY TO USE THEM

MODULES

WHY TO USE THEM

1. **Avoid confusing LAMBDAs with native functions**
2. **Keep separate functions with the same name but different domains**
3. **Easily group functions around specific topics**
4. **Keep test functions separate from development**
5. **Keep functions by other developers separate**
6. **Easier to find when typing formulas**



PARAMETERS

OPTIONAL PARAMS

PARAMETERS

OPTIONAL PARAMS

1. An optional parameter is enclosed in square brackets like this: [param]

2. A user can omit the parameter when using the function

3. If the user omits the parameter, ISOMITTED(param) returns TRUE

4. It’s up to you to control what happens when:
    1. An optional parameter is supplied
    2. An optional parameter is omitted

OPTIONAL PARAMS

DEMO

1. What happens when a non-optional parameter is omitted
2. Testing ISOMITTED
3. One-of two optional parameters
4. IFOMITTED

PARAMETERS

TYPES / DOMAINS

PARAMETERS

TYPES / DOMAINS

1. Usually, a parameter will expect a specific data type

2. Sometimes, a parameter may expect a limited domain of values
    1. E.g. a numeric parameter that expects only numbers between 1 and 5

3. Unexpected values may produce confusing results

4. It’s up to you to control what happens when:
    1. A value of the wrong data type is provided
    2. An out-of-range value is provided

DEMO

1. Testing data types
2. Domain test

PARAMETERS

TAKEAWAYS

1. **Non-optional params will cause #VALUE! if omitted**
2. **Use IFOMITTED for optional params**
3. **Test for correct types**
4. **Test for correct domains**

Vote to add IFOMITTED to Excel

BUILDING BLOCKS

BUILDING BLOCKS

DEMO

1. Components of DESCRIBE
    1. Lambda-ized native functions
    2. FUNCS
    3. STACKER
    4. AGG
    5. DESCRIBE

THANK YOU!

flexyourdata.com | youtube.com/@flexyourdata | linkedin.com/in/owenhprice