dataset_sample_js contains:
- original_code_functions: found or created JavaScript functions that form the basis from which the dataset is created.
- obfuscated_code_functions: obfuscator.io; ByteHide.
- deobfuscated_code_functions: (relative.im: obfuscator.io; ByteHide); (js-beautify: obfuscator.io; ByteHide); (obf-io.deobfuscate.io: obfuscator.io; ByteHide).

Save dataset_sample_js in Downloads/Scaricati.

Download obfuscator.io (https://github.com/javascript-obfuscator/javascript-obfuscator terminal: npm install --save-dev javascript-obfuscator) on the virtual machine (VirtualBox) with ubuntu, save node_modules in a folder called obfuscator.io.

Download relative.im (https://github.com/relative/synchrony terminal: sudo su - npm install --global deobfuscator) on the virtual machine (VirtualBox) with ubuntu.

Download js-beautify (https://github.com/beautifier/js-beautify terminal: npm install js-beautify@next) on the virtual machine (VirtualBox) with ubuntu, save node_modules in a folder called js-beautify@next.

Download obf-io.deobfuscate.io (https://www.npmjs.com/package/obfuscator-io terminal: npm i obfuscator-io) on the virtual machine (VirtualBox) with ubuntu, save node_modules in a folder called obf-io.deobfuscate.io.

OBFUSCATED via ByteHide:
https://www.bytehide.com/products/shield-obfuscator/javascript

BASH SCRIPT TO OBFUSCATE:

- - obfuscator.io - -

__general terminal

```
for f in `ls ~/Scaricati/dataset_sample_js/original_code_functions/*.js`; do ~/obfuscator.io/node_modules/javascript-obfuscator/bin/./javascript-obfuscator $f; done; for h in `ls ~/Scaricati/dataset_sample_js/original_code_functions/*obfuscated.js`; do mv $h ~/Scaricati/dataset_sample_js/obfuscated_code_functions/obfuscator.io; done
```

BASH SCRIPT TO DEOBFUSCATE:

- - relative.im - -

- - obfuscator.io - -

__general terminal

```
for f in `ls ~/Scaricati/dataset_sample_js/obfuscated_code_functions/obfuscator.io/*-obfuscated.js`; do synchrony deobfuscate $f; done; for h in `ls ~/Scaricati/dataset_sample_js/obfuscated_code_functions/obfuscator.io/*-obfuscated.cleaned.js`; do mv $h
```

```
~/Scaricati/dataset_sample_js/deobfuscated_code_functions/relative.im/obfuscator.io; done
```

- - ByteHide - -

__general terminal

```
for f in `ls ~/Scaricati/dataset_sample_js/obfuscated_code_functions/ByteHide/*.js`; do synchrony deobfuscate $f; done; for h in `ls ~/Scaricati/dataset_sample_js/obfuscated_code_functions/ByteHide/*-shield.cleaned.js`; do mv $h ~/Scaricati/dataset_sample_js/deobfuscated_code_functions/relative.im/ByteHide; done
```

- - js-beautify - -

- - obfuscator.io - -

__general terminal with path ~/Scaricati/dataset_sample_js/obfuscated_code_functions/obfuscator.io

```
for f in `ls *-obfuscated.js`; do ~/js-beautify@next/node_modules/js-beautify/js/bin/./js-beautify.js -f $f > ~/Scaricati/dataset_sample_js/deobfuscated_code_functions/js-beautify/obfuscator.io/beautify-$f; done;
```

- - ByteHide - -

__general terminal with path ~/Scaricati/dataset_sample_js/obfuscated_code_functions/ByteHide

```
for f in `ls *-shield.js`; do ~/js-beautify@next/node_modules/js-beautify/js/bin/./js-beautify.js -f $f > ~/Scaricati/dataset_sample_js/deobfuscated_code_functions/js-beautify/ByteHide/beautify-$f; done;
```

- - obf-io.deobfuscate.io - -

Path : ~/obf-io.deobfuscate.io/node_modules/obfuscator-io

In the folder dist :

create a file called originalrun.js and save here the original content of run.js.

in the run.js file on line 16 change 'input/source.js' to 'input/obfuscator.js'.

in the run.js file on line 17 change 'output/output.js' to 'output/deobf.js'

In the folder input :

copy the source.js file and paste it out of the folder and then create a new empty file named obfuscator.js.

In the folder output :

copy the output.js file and paste it out of the folder and then create a new empty file named deobf.js.

- - obfuscator.io - -

MOVE factorial.js in folder factorial1_2_3 __general terminal with path ~/obf-io.deobfuscate.io/node_modules/obfuscator-io

```
for f in `ls ~/Scaricati/dataset_sample_js/obfuscated_code_functions/obfuscator.io/*.js`; do cat $f > ~/obf-io.deobfuscate.io/node_modules/obfuscator-io/input/obfuscator.js; var=$(echo $f | xargs -n 1 basename); ~/obf-io.deobfuscate.io/node_modules/obfuscator-io/dist/./run.js; cp ~/obf-io.deobfuscate.io/node_modules/obfuscator-io/output/deobf.js ~/Scaricati/dataset_sample_js/deobfuscated_code_functions/obf-io.deobfuscate.io/obfuscator.io/$var; done;
```

EXTRACT from folder factorial1_2_3 : factorial.js

- - ByteHide - -

MOVE factorial.js in folder factorial

__general terminal with path ~/obf-io.deobfuscate.io/node_modules/obfuscator-io

```
for h in `ls ~/Scaricati/dataset_sample_js/obfuscated_code_functions/ByteHide/*.js`; do cat $h > ~/obf-io.deobfuscate.io/node_modules/obfuscator-io/input/obfuscator.js; var=$(echo $h | xargs -n 1 basename); ~/obf-io.deobfuscate.io/node_modules/obfuscator-io/dist/./run.js; cp ~/obf-io.deobfuscate.io/node_modules/obfuscator-io/output/deobf.js ~/Scaricati/dataset_sample_js/deobfuscated_code_functions/obf-io.deobfuscate.io/ByteHide/$var; done;
```

EXTRACT from folder factorial : factorial.js

BASH SCRIPT TO RUN: __general terminal

- - original code - -

```
for f in `ls ~/Scaricati/dataset_sample_js/original_code_functions/*.js`; do node $f; done;
```

- - obfuscated code obfuscator.io - -

```
for h in `ls ~/Scaricati/dataset_sample_js/obfuscated_code_functions/obfuscator.io/*-obfuscated.js`; do node $h; done;
```

- - obfuscated code ByteHide - -

```
for b in `ls ~/Scaricati/dataset_sample_js/obfuscated_code_functions/ByteHide/*.js`; do node $b; done;
```

- - deobfuscated code relative.im - -

- - obfuscator.io - -

```
        for k in `ls ~/Scaricati/dataset_sample_js/deobfuscated_code_functions/relative.im/obfuscator.io/*-obfuscated.cleaned.js`; do node $k; done;

        - - ByteHide - -
        for k in `ls ~/Scaricati/dataset_sample_js/deobfuscated_code_functions/relative.im/ByteHide/*.js`; do node $k; done;

- - deobfuscated code js-beautify - -
        - - obfuscator.io - -
        for z in `ls ~/Scaricati/dataset_sample_js/deobfuscated_code_functions/js-beautify/obfuscator.io/*.js`; do node $z; done;

        - - ByteHide - -
        for z in `ls ~/Scaricati/dataset_sample_js/deobfuscated_code_functions/js-beautify/ByteHide/*.js`; do node $z; done;

- - deobfuscated code obf-io.deobfuscate.io - -
        - - obfuscator.io - -
        for q in `ls ~/Scaricati/dataset_sample_js/deobfuscated_code_functions/obf-io.deobfuscate.io/obfuscator.io/*.js`; do node $q; done;

        - - ByteHide - -
        for q in `ls ~/Scaricati/dataset_sample_js/deobfuscated_code_functions/obf-io.deobfuscate.io/ByteHide/*.js`; do node $q; done;
```

When the dataset folders already had functions saved inside and I wanted to add new original, obfuscated and deobfuscated functions, I acted like this:
- I created new folders: **O_original_code_functions** where I saved the new original functions; **obfio** where I saved the result of the obfuscation, using the obfuscator.io tool, of the new original functions; **rel_obfio** where I saved the result of the deobfuscation, using the relative.im tool, of the original functions obfuscated via obfuscator.io, **js-beau_obfio** where I saved the result of the deobfuscation, using the js-beautify tool, of the original functions obfuscated via obfuscator.io, **deob_obfio** where I saved the result of the deobfuscation, using the obf-io.deobfuscate.io tool, of the original functions obfuscated via obfuscator.io.

- I saved all these folders empty in the **new_func** folder and when I want to add new functions to the dataset I take the folders, place them outside new_func, save the new functions in the respective folder and then through the following bash script the new functions are obfuscated and deobfuscated and placed in the respective folder:

<u>obfuscate:</u>

- obfuscator.io

THE RESULT OF THE OBFUSCATION WILL BE SAVED IN OBFIO

__general terminal

```
for f in `ls
~/Scaricati/dataset_webtools/dataset_sample_js/O_original_code_functions/*.js`; do
~/obfuscator.io/node_modules/javascript-obfuscator/bin/./javascript-obfuscator $f;
done; for h in `ls
~/Scaricati/dataset_webtools/dataset_sample_js/O_original_code_functions/*obfusc
ated.js`; do mv $h ~/Scaricati/dataset_webtools/dataset_sample_js/obfio; done
```

<u>deobfuscate:</u>

- relative.im

THE RESULT OF THE DEOBFUSCATION WILL BE SAVED IN REL_OBFIO

__general terminal

```
for f in `ls ~/Scaricati/dataset_webtools/dataset_sample_js/obfio/*-obfuscated.js`; do
synchrony deobfuscate $f; done; for h in `ls
~/Scaricati/dataset_webtools/dataset_sample_js/obfio/*-obfuscated.cleaned.js`; do
mv $h ~/Scaricati/dataset_webtools/dataset_sample_js/rel_obfio; done
```

- js-beautify

THE RESULT OF THE DEOBFUSCATION WILL BE SAVED IN JS-BEAU_OBFIO

__general terminal with path ~/Scaricati/dataset_webtools/dataset_sample_js/obfio

```
for f in `ls *-obfuscated.js`; do
~/js-beautify@next/node_modules/js-beautify/js/bin/./js-beautify.js -f $f >
~/Scaricati/dataset_webtools/dataset_sample_js/js-beau_obfio/beautify-$f; done;
```

- obf-io.deobfuscate.io

THE RESULT OF THE DEOBFUSCATION WILL BE SAVED IN DEOB_OBFIO

__general terminal with path ~/obf-io.deobfuscate.io/node_modules/obfuscator-io

```
for f in `ls ~/Scaricati/dataset_webtools/dataset_sample_js/obfio/*.js`; do cat $f >
~/obf-io.deobfuscate.io/node_modules/obfuscator-io/input/obfuscator.js; var=$(echo
$f | xargs -n 1 basename);
~/obf-io.deobfuscate.io/node_modules/obfuscator-io/dist/./run.js; cp
```

```
~/obf-io.deobfuscate.io/node_modules/obfuscator-io/output/deobf.js
~/Scaricati/dataset_webtools/dataset_sample_js/deob_obfio/$var; done;
```

when I saved all the new functions in the respective momentary folder and then through the bash script I obfuscated and deobfuscated these functions and saved in the respective momentary folders, then I selected all the functions in these folders and moved them to the respective folders of the final dataset, and finally I saved the temporary folders in new_func so that they can be used again when I want to add new functions to the dataset.