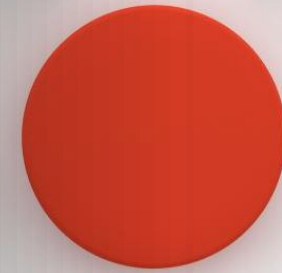


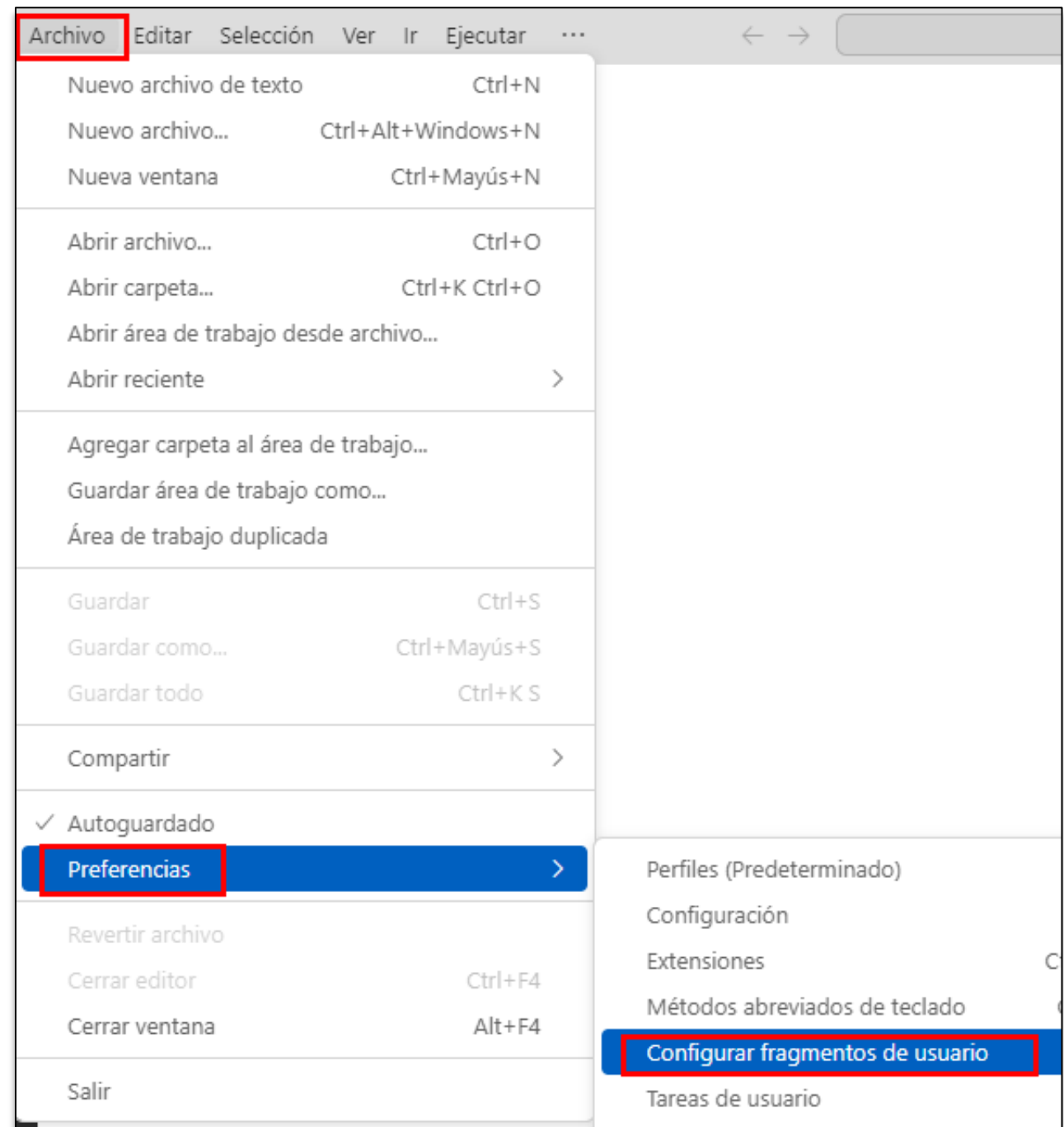


COMO CREAR / IMPORTAR SNIPPETS

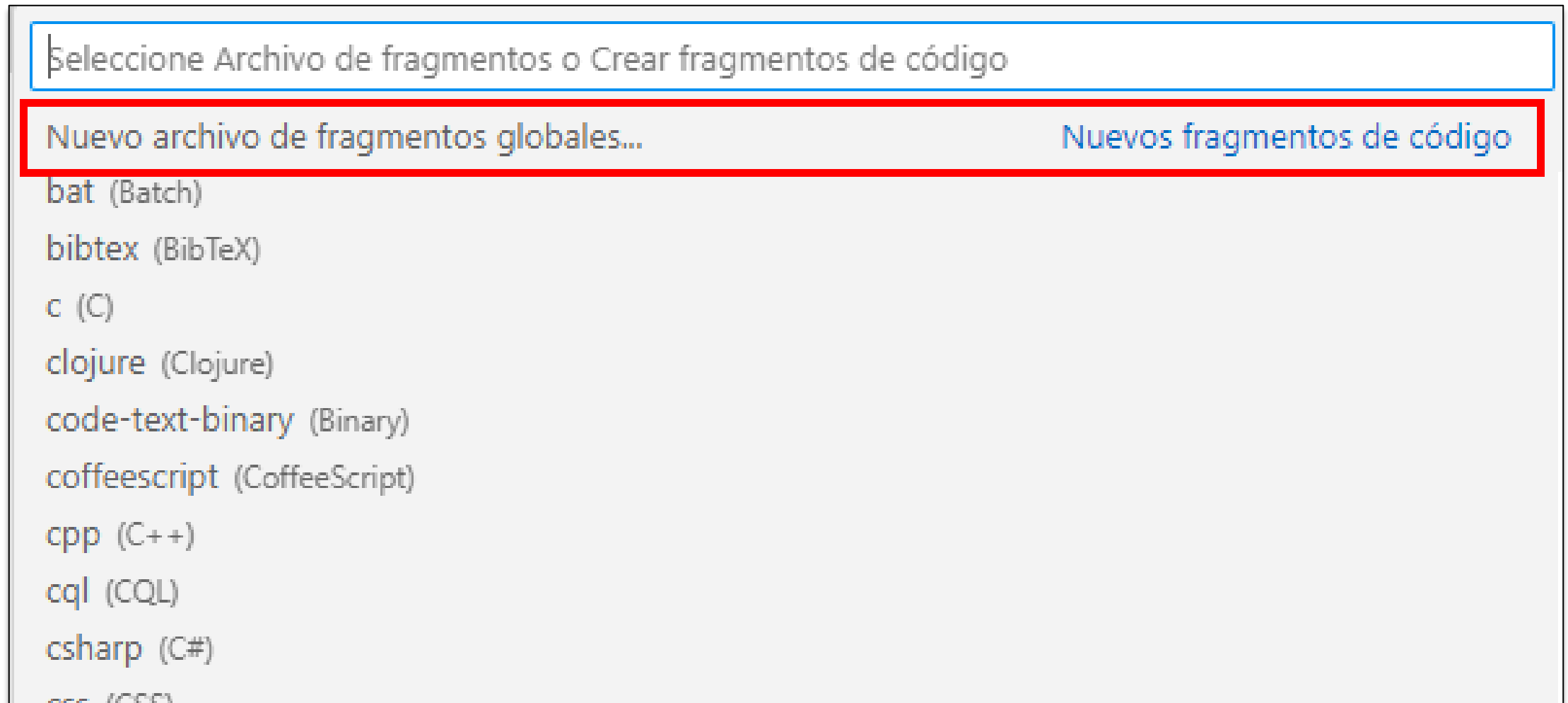
VISUAL STUDIO
CODE



- Nos vamos a **Archivo** → **Preferencias** → **Configurar fragmentos de usuario**



- Seleccionamos Nuevo archivo de fragmentos globales ...



- Hay que escribir el nombre del nuevo archivo: por ejemplo: javascript

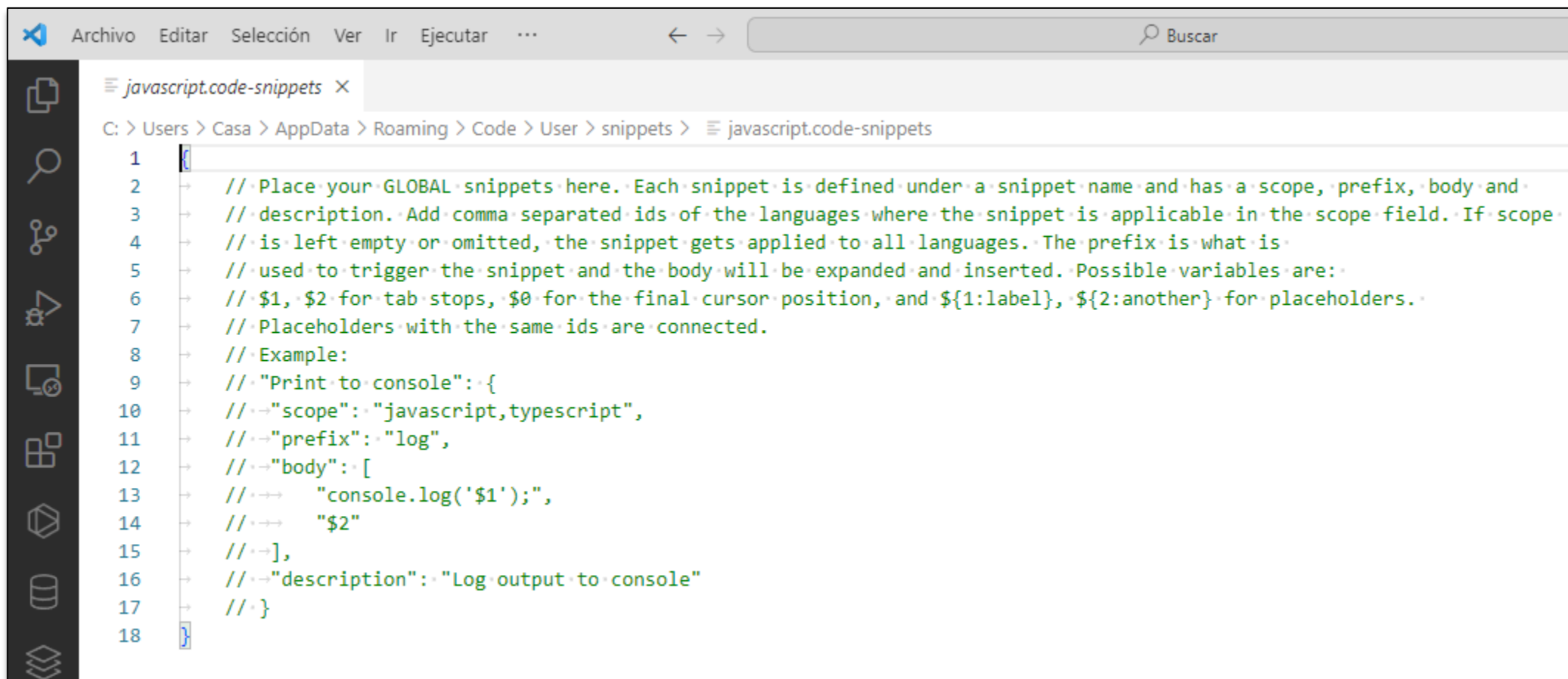
Escriba el nombre del archivo de fragmento de código

Presione "Entrar" para confirmar su entrada o "Esc" para cancelar

javascript

Presione "Entrar" para confirmar su entrada o "Esc" para cancelar

- Nos aparece el fichero entero comentado, excepto la llave de inicio y de fin



```
1 {
2  // Place your GLOBAL snippets here. Each snippet is defined under a snippet name and has a scope, prefix, body and
3  // description. Add comma separated ids of the languages where the snippet is applicable in the scope field. If scope
4  // is left empty or omitted, the snippet gets applied to all languages. The prefix is what is
5  // used to trigger the snippet and the body will be expanded and inserted. Possible variables are:
6  // $1, $2 for tab stops, $0 for the final cursor position, and ${1:label}, ${2:another} for placeholders.
7  // Placeholders with the same ids are connected.
8  // Example:
9  // "Print to console": {
10 //   "scope": "javascript,typescript",
11 //   "prefix": "log",
12 //   "body": [
13 //     "console.log('$1');",
14 //     "$2"
15 //   ],
16 //   "description": "Log output to console"
17 // }
18 }
```

- En el aula virtual tenemos un fichero de texto con los snippets ya preparados por Juan (¿dónde estás Juan?)



10b. FETCH



snippets para fetch



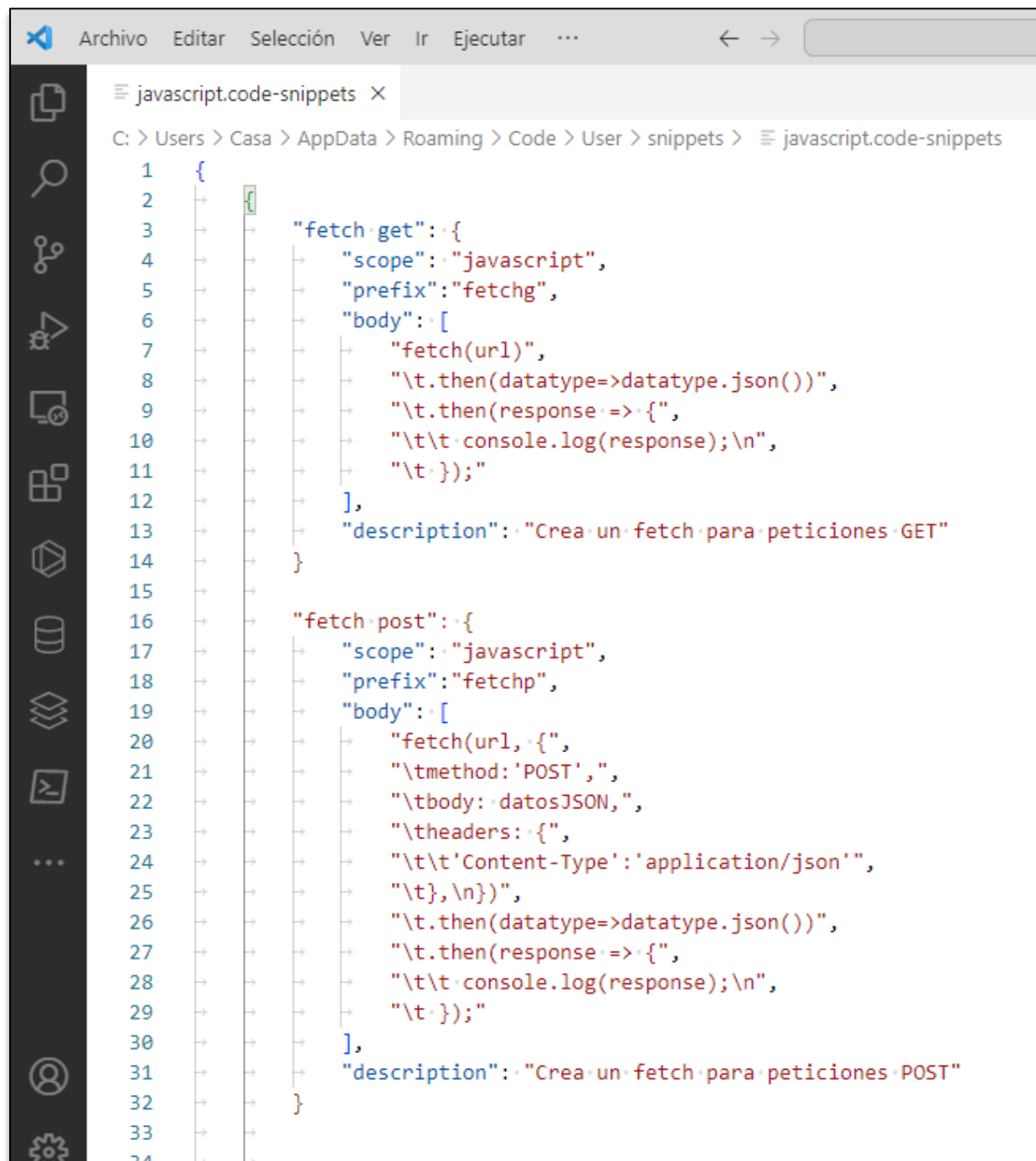
10c. MAP

- Abrimos dicho fichero ,
seleccionamos todo el
contenido y lo pegamos en
el fichero de vscode

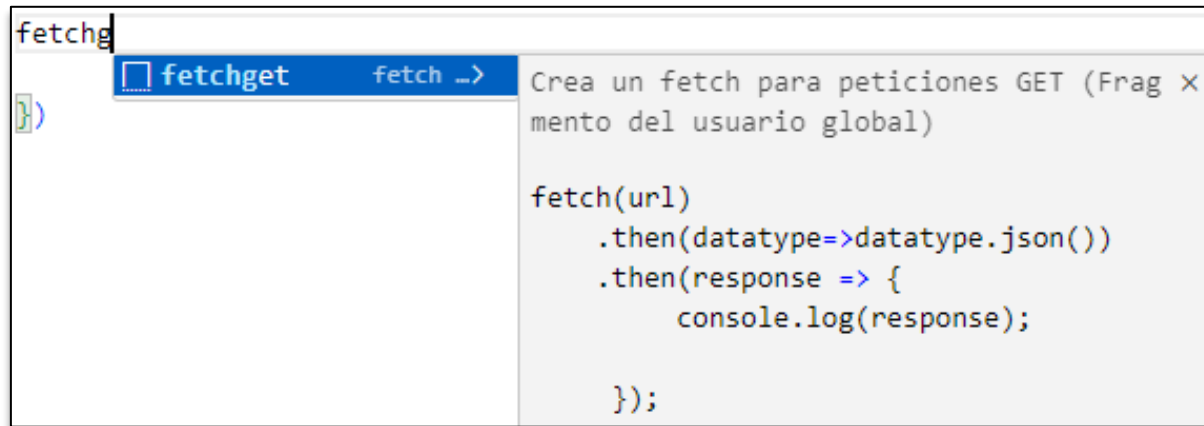
```
{
  "fetch get": {
    "scope": "javascript",
    "prefix": "fetchg",
    "body": [
      "fetch(url)",
      "\t.then(datatype=>datatype.json())",
      "\t.then(response => {",
      "\t\t\t console.log(response);\n",
      "\t });"
    ],
    "description": "Crea un fetch para peticiones GET"
  }
}

"fetch post": {
  "scope": "javascript",
  "prefix": "fetchp",
  "body": [
    "fetch(url, {",
    "\tmethod: 'POST'",
    "\tbody: datosJSON,",
    "\theaders: {",
    "\t\t'Content-Type': 'application/json'",
    "\t},\n})",
    "\t.then(datatype=>datatype.json())",
    "\t.then(response => {",
    "\t\t\t console.log(response);\n",
    "\t });"
  ],
  "description": "Crea un fetch para peticiones POST"
}

}
```

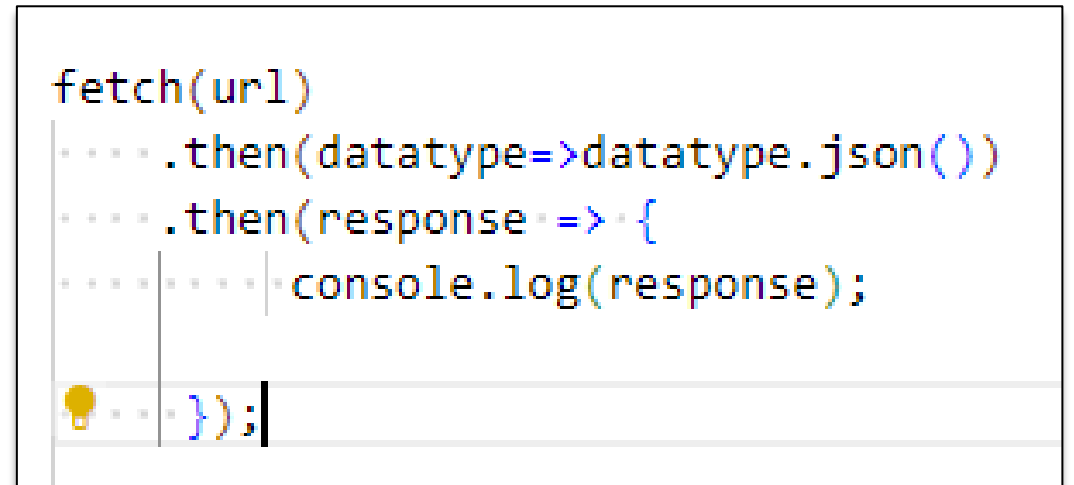


- Ya podemos usar los snippets, para ello debemos utilizar el valor guardado en el campo “**prefix**”.



A screenshot of an IDE's code completion menu. The text 'fetchg' is entered in the editor. A dropdown menu is open, showing a snippet named 'fetchget' with a 'fetch' prefix. The description of the snippet is 'Crea un fetch para peticiones GET (Frag x mento del usuario global)'. The code for the snippet is:

```
fetch(url)
  .then(datatype=>datatype.json())
  .then(response => {
    console.log(response);
  });
```



A screenshot of a code editor showing the 'fetch' snippet being applied. The code is:

```
fetch(url)
  .then(datatype=>datatype.json())
  .then(response => {
    console.log(response);
  });
```

- ssadfasdf

- ssadfasdf