ASSIGNEMENT

Code.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;
contract StringReturn {
  string public zoneName = "Your Zone Name";
 function getZoneName() public view returns (string memory) {
return zoneName;
}
 }
```

Program description

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;
```

- This section includes SPDX License Identifier, which specifies the license for the code (MIT in this case).
- **`pragma solidity ^0.8.0;`** specifies the version of the Solidity compiler to be used. In this case, it's version 0.8.0.

```
contract StringReturn {
    string public zoneName = "Your Zone Name";
```

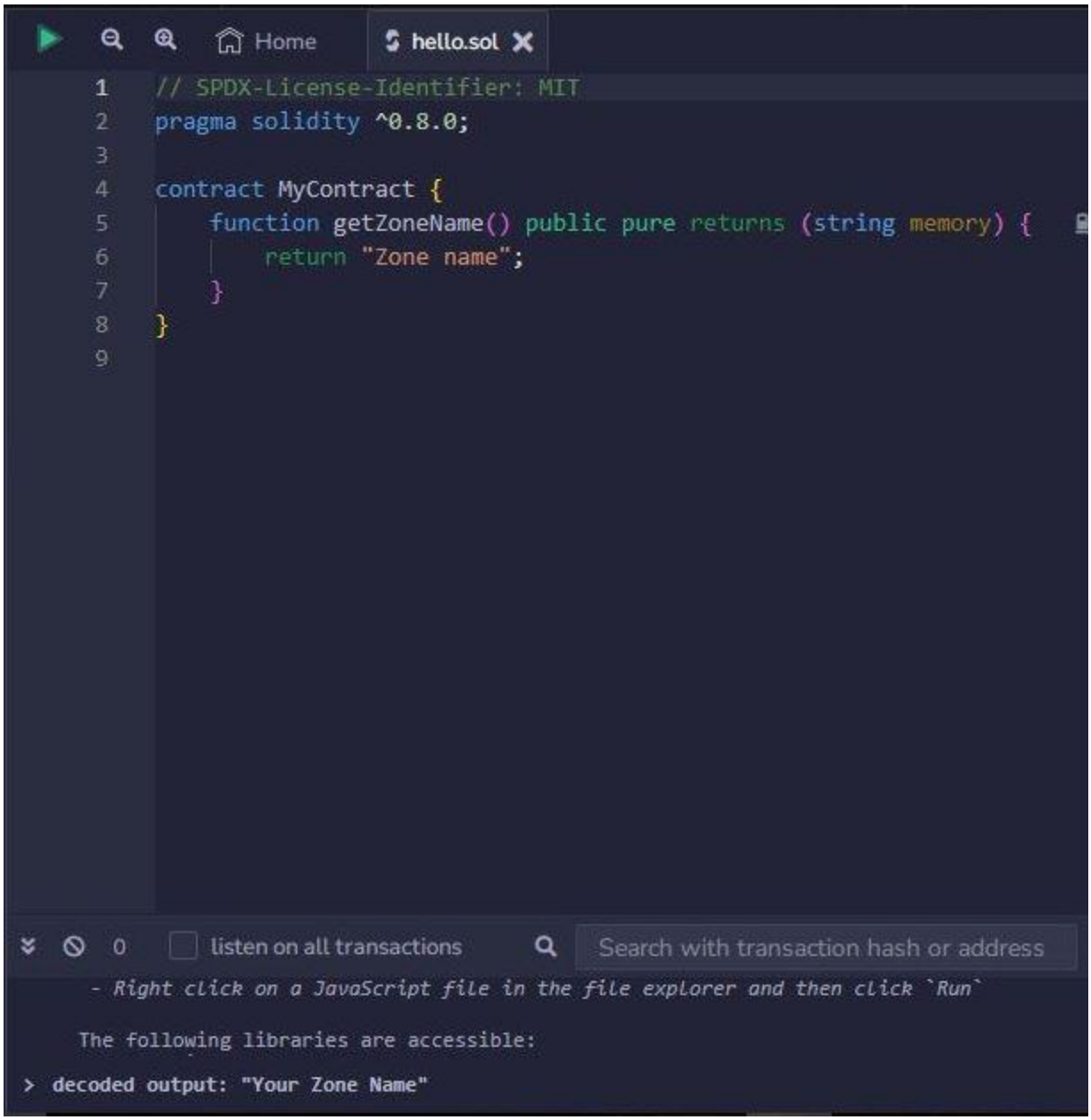
- This code defines a Solidity smart contract named **`StringReturn`**. Contracts are the building blocks of Ethereum applications.
- Inside the contract, there is a state variable **`zoneName`** which is of type **`string`**. It is declared as **`public`**, which means it can be read from outside the contract. The initial value of **`zoneName`** is set to "Your Zone Name".

```
    function getZoneName() public view returns (string memory) {
        return zoneName;
    }
```

- The **`getZoneName`** function is a public function, which means it can be called from outside the contract.
- It is marked as **`view`**, indicating that it doesn't modify the state of the contract and is read-only.
- This function returns a **`string`** type, and the **`memory`** keyword indicates that the returned value will be temporary in memory.
- Inside the function, it simply returns the value of the **`zoneName`** state variable.

Output:

decoded output: "Your Zone Name"


Home
hello.sol
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;
contract MyContract {
function getZoneName() public pure returns (string memory) {
return "Zone name";
}
}
0
listen on all transactions
Search with transaction hash or address
- Right click on a JavaScript file in the file explorer and then click `Run`
The following libraries are accessible:
> decoded output: "Your Zone Name"