Homework

- Upload your solution to git, and copy the git link to Sakai Assignment.
- Draw a lexical environment diagram for the right code and show:
  - global lexical environment (LE)
  - LE for makeArmy()
  - LE for LE of the while loop
  - LE for army[0]
  - What will army[0] alert?
  - Can you fix the code?
  - How will the diagram change?

```
function makeArmy() {
    let shooters = [];
    let i = 0;
    while (i < 2) {
        let shooter = function() {
            alert(i);
        };
        shooters.push(shooter);
        i++;
    }
    return shooters;
}
let army = makeArmy();
army[0];
```

+ Global lexical environment (LE)

- Creation phase → Execution phase

Global Execution context: Creation → Execution
LE: { makeArmy: fn, output: Null, army: [fn, fn] }
this: window
makeArmy() army[0] = function() {alert(i)}

TDZ: ~~army~~

+ LE for makeArmy():

→ makeArmy() functional EC, Creation phase → Execution
LE: { arguments: {length: 0}, outer: global
shooters = [function() {alert(i)}]
~~i = 0~~ ~~i = 1~~ i = 2
this: window
return shooters;

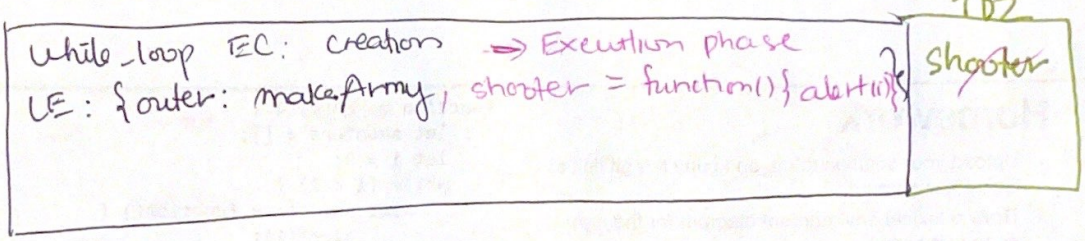
TDZ: ~~shooters~~ i

→ Execution phase: after while_loop, changed the condition to while (i < 2). The LE is being changed as below

+ LE for LE of the while_loop:

- Each iteration of while_loop has own LE, only use i = 0 as example to save time

→ Creation phase:



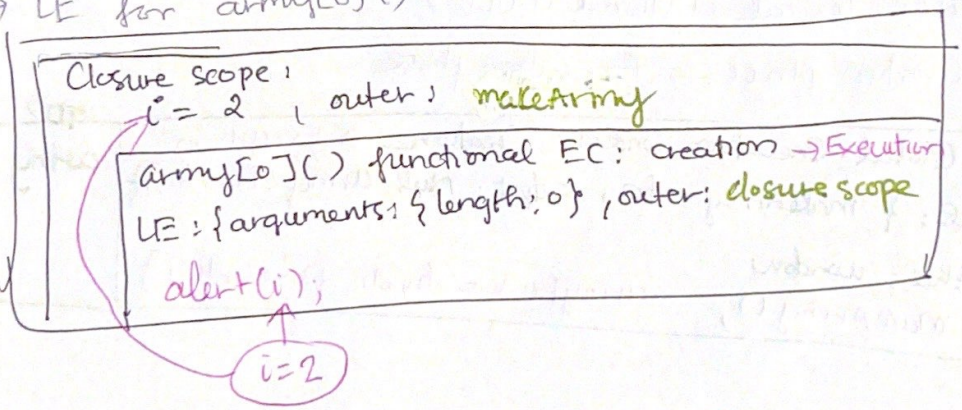
→ Execution phase:

+ shooters.push(shooter);

+ i++;

The two statements above will cause the changes in makeArmy() LE

→ LE for army[0]()



+ what will army[0]() alert? 2

+ Can you fix the code?

```
function makeArmy() {
    let shooters = [];
    let i = 0;
    while (i < 2) {
        let i = j;
        let shooter = function() {
            console.log(j);
        };
        shooters.push(shooter);
        i++;
    }
return shooters; }
```

```
let army = makeArmy();
army.forEach(f => f());
```