Day 11 notes

Key Takeaway	Closures
Learning Date	@July 8, 2025
Module	**Module 2: Deep Dive into Functions & Objects**
Status	Done
Topic	✅ Day 11: Closures
Video Link	https://www.youtube.com/watch?v=IA7CGz3iHyI&list=PLIJrr73KDmRw2Fwwjt6cPC_tk5vcSICCu&index=11

Closures :

A closures is a function that allow to access to outer function even after the outer function has been executed.

```
// lets anoter example.......
function outerCount() {
  let count = 0;
  return function innerCount() {
    count++;
    console.log(count);
  };
}
const retVal = outerCount();
retVal();
retVal();
retVal();
```

“The
`count` variable retains its value and cannot be reinitialized.”

```
//___________Closures => Real World Example(Banking System)_______________
function createBankAccount(initialBalance) {
  let balance = initialBalance;

  return {
    deposite: (amount) => {
      balance = balance + amount;
      console.log("Deposited", amount, "Current balance", balance);
    },
    withdraw: (amount) => {
      if (amount > balance) {
        console.warn("Insifficient balance");
      } else {
        balance = balance - amount;
        console.log("Withdrawn", amount, "Current balance", balance);
      }
    },
    checkBalance: () => {
      console.log("Current Balance", balance);
    },
  };
}

const tapaScriptAccount = createBankAccount(500);
console.log(tapaScriptAccount.withdraw(200));
console.log(tapaScriptAccount.withdraw(50));
console.log(tapaScriptAccount.withdraw(20));
console.log(tapaScriptAccount.checkBalance());
```

Usefulness of closures :

- You can keep the variables private without exposing them.
- You can create a function factory
- You can stop variable pollution.
- You can keep a Variable alive between multiples calls