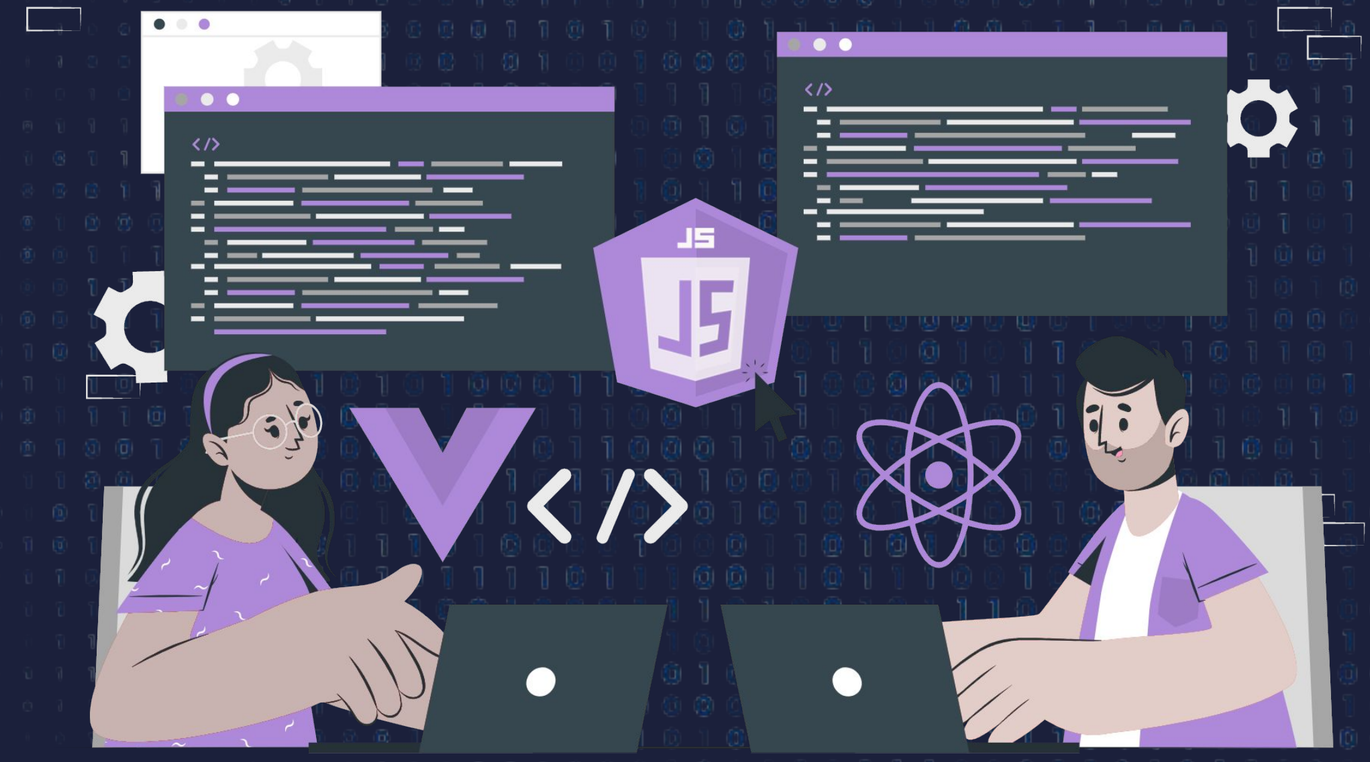
PW SKILLS

First Program in Closure and Understanding Closures



Topics

1. Example to demonstrate closure
2. How closures maintain the lexical environment variables even after the parent function is removed
3. Example 1
4. Examples 2

Example to demonstrate closure

```
function outer() {
  const a = 10;

  function inner() {
    const b = 20;
    console.log(a + b);
  }

  return inner;
}

const closureFn = outer();
closureFn(); // outputs 30
```

How closures maintain the lexical environment variables even after the parent function is removed

In JavaScript, when a function is created, it includes a reference to the outer environment where the function was defined. This reference is stored in a closure object along with the inner function. When the inner function is called, it can access the variables in the outer environment through this closure, even if those variables have since been modified or removed. This is because the closure retains a snapshot of the outer environment at the time the closure was created.

Example 1:

```
function parent(){

const count = 0;

function child(){

console.log(count++);

}

return child;
}

}

const increment = parent()
increment() //1
increment() //2
increment() //3
```

Example 2:

```
function createPerson(name) {
  const age = 0;

  return {
    setName: function(newName) {
      name = newName;
    },
    setAge: function(newAge) {
      age = newAge;
    },
    getInfo: function() {
      return name + ' is ' + age + ' years old.';
    }
  }
}

const person = createPerson('John');

person.setAge(30);
console.log(person.getInfo()); // logs 'John is 30 years old.'
```

THANK YOU