SOLID Principles in JavaScript

1. Single Responsibility Principle (SRP)

```
[Bad Example]:
class User {
  constructor(name, email) {
    this.name = name;
    this.email = email;
  }

  saveToDatabase() {
    // Code to save user to DB
  }
}

[Good Example]:
class User {
  constructor(name, email) {
    this.name = name;
    this.email = email;
  }
}

class UserDB {
  save(user) {
    // Code to save user to DB
  }
}
```

2. Open/Closed Principle (OCP)

```
[Bad Example]:
class Discount {
  getDiscount(type) {
    if (type === 'regular') return 10;
    if (type === 'premium') return 20;
  }
}

[Good Example]:
class Discount {
  getDiscount() {
    return 0;
  }
}

class RegularDiscount extends Discount {
  getDiscount() {
    return 10;
  }
}

class PremiumDiscount extends Discount {
  getDiscount() {
    return 20;
  }
}
```

3. Liskov Substitution Principle (LSP)

```
[Bad Example]:
class Bird {
  fly() {
    console.log("Flying");
  }
}

class Penguin extends Bird {
  fly() {
    throw new Error("Penguins can't fly!");
  }
}

[Good Example]:
class Bird {}

class FlyingBird extends Bird {
  fly() {
    console.log("Flying");
  }
}

class Penguin extends Bird {
  swim() {
    console.log("Swimming");
  }
}
```

4. Interface Segregation Principle (ISP)

```
[Bad Example]:
class Worker {
  work() {}
  eat() {}
}
```

```
class Robot extends Worker {
  eat() {
    throw new Error("Robots don't eat!");
  }
}

[Good Example]:
class Workable {
  work() {}
}

class Eatable {
  eat() {}
}

class Human extends Workable {
  work() {
    console.log("Working");
  }
}

Object.assign(Human.prototype, new Eatable());
```

5. Dependency Inversion Principle (DIP)

```
[Bad Example]:
class MySQL {
  connect() {
    console.log("Connected to MySQL");
  }
}

class App {
  constructor() {
    this.db = new MySQL();
  }

  init() {
    this.db.connect();
  }
}

[Good Example]:
class Database {
  connect() {}
}
```

```
class MySQL extends Database {
  connect() {
    console.log("Connected to MySQL");
  }
}

class App {
  constructor(database) {
    this.db = database;
  }

  init() {
    this.db.connect();
  }
}
```