2. DRY - Don't Repeat Yourself.

Every piece of knowledge or logic should have a single, unambiguous representation in the system.

- Avoid duplicating logic.
- Avoid copying the same code in multiple places.

Why DRY-

code duplication causes:-

→multiple places ~~to~~ update for one change.

- Inconsistent behavior.
- Higher bug probablity.

DRY help achieve:-

- Maintaiablity, consitency, Faster changes., cleaner code bases.

DRY reduces maintance cost by ensuring changes are made in one Place.

Wrong Design

```
double discount for User (double Price){
  return Price * 0.9;
}
double discount for Admin (double price){
  return price * 0.9;
}
```

Correct Design

```
double calculate Discount (double price){
 return price * 0.9;
}
```

- centralize validation in one place.
- Reuse across layers.

Eg: One law book,
not seperate rules written in every office.

Use when

- you see copy-pasted code.
- Business rules repeat.
- Constants appear in multiple places
- Same algorithm appears in multiple classes.