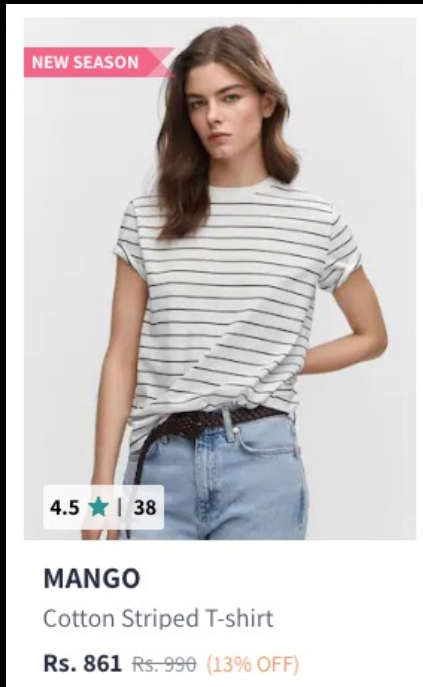


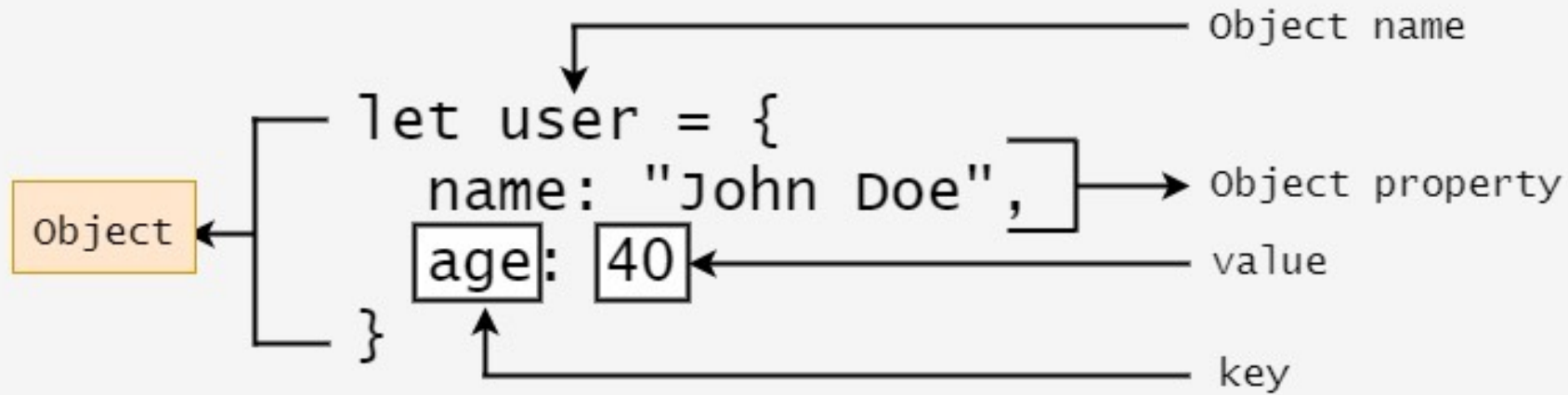
What is an Object?



```
let product = {  
  company: 'Mango',  
  item_name: 'Cotton striped t-shirt',  
  price: 861  
};
```

1. **Groups** multiple **values** together in **key-value** pairs.
2. **How to Define:** Use **{}** to enclose **properties**.
3. **Example:** product {name, price}
4. **Dot Notation:** Use **.** **operator** to access values.
5. **Key Benefit:** Organizes **related data** under a **single name**.

Object Syntax?



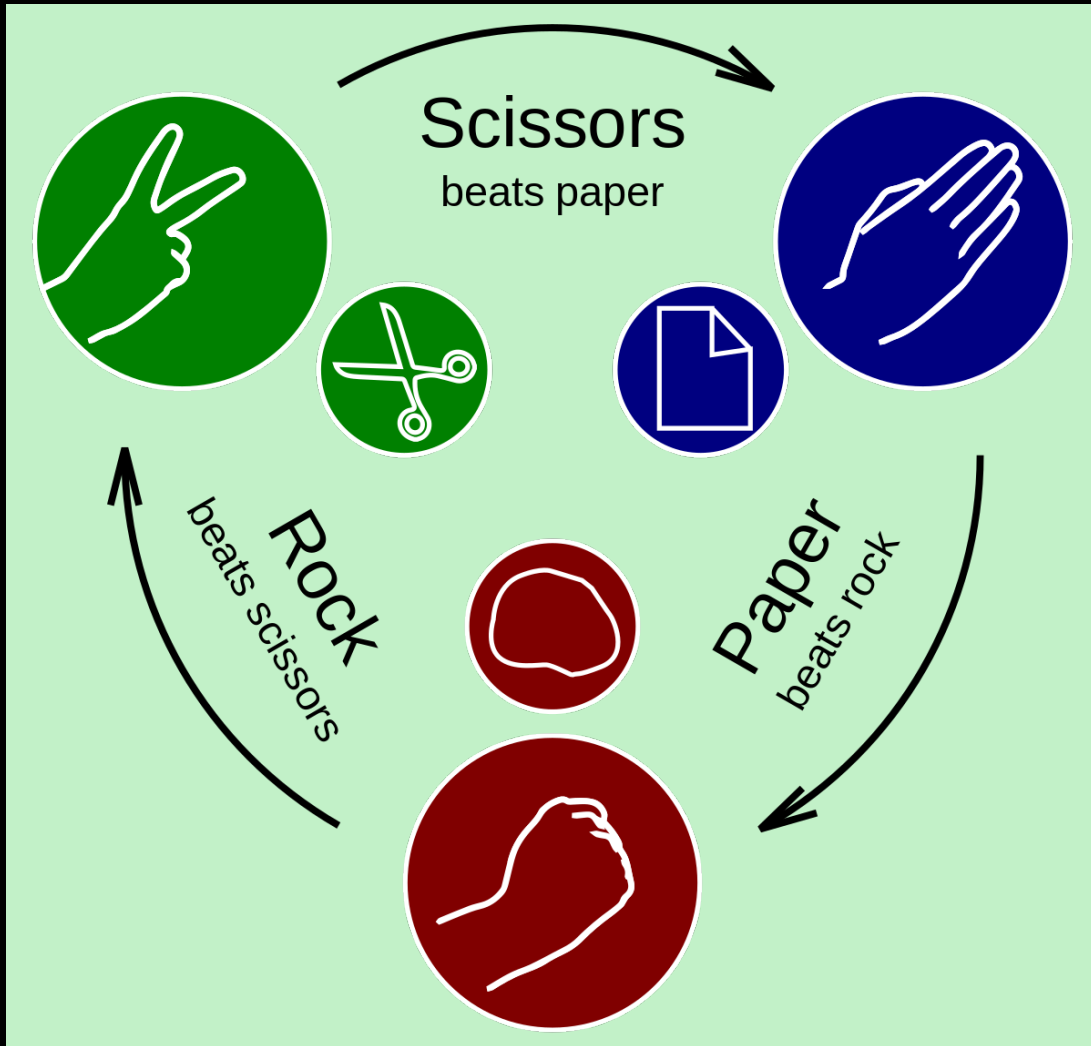
1. **Basic Structure:** Uses `{}` to enclose data.
2. **Rules:** **Property** and **value** separated by a **colon(:)**
3. **Comma:** Separates different **property-value** pairs.
4. **Example:** `{ name: "Laptop", price: 1000 }`

Accessing Objects



1. **Dot Notation:** Access **properties** using **.** **Operator** like **product.price**
2. **Bracket Notation:** Useful for **properties** with **special characters** **product["nick-name"]**. Variables can be used to access properties
3. **typeof** returns **object**.
4. **Values** can be **added** or **removed** to an object
5. **Delete** Values using **delete**

Project Rock-Paper-Scissor Game



Rock Paper Scissors Game

Click on one of the following to play the game:



Rock



Paper



Scissors

Create object for
maintaining Score

Inside Object

```
let product = {  
  company: 'Mango',  
  itemName: 'Cotton striped t-shirt',  
  price: 861,  
  rating: {  
    stars: 4.5,  
    noOfReviews: 87  
  },  
  displayPrice: function() {  
    return `$$${this.price.toFixed(2)}`;  
  }  
};
```



1. **Objects** can contain **Primitives** like **numbers** and **strings**.
2. **Objects** can contain **other objects** and are called Nested Objects.
3. **Functions** can be object properties.
4. **Functions** inside an object are called **methods**.
5. **Null Value: Intentionally** leaving a property **empty**.