Functions - Rest Parameters, Object & Array Handling

🧠 1. Function Parameters vs Arguments

js

CopyEdit

```
function add(a, b) {  // a and b are called parameters
  return a + b;
}

add(2, 3); // 2 and 3 are arguments
```

📦 2. Return Single Value from Function

js

CopyEdit

```
function calculateCartPrice(num1) {
  return num1;
}

console.log(calculateCartPrice(2)); // 2
```

🧠 *It just returns the first value passed to it.*

📦 3. Rest Parameters (...num1)

js

CopyEdit

```
function calculateCartPrice(...num1) {
  return num1;
}

console.log(calculateCartPrice(200, 400, 600)); // [200, 400, 600]
```

💡 **Rest parameter** collects multiple values into an array.

🧾 4. Rest After Specific Parameters

js

CopyEdit

```
function calculateCartPrice(val1, val2, ...num1) {
  return num1;
}

console.log(calculateCartPrice(200, 400, 600)); // [600]
```

📌 First 2 values go to val1 and val2, rest go to num1.

👤 5. Passing Object to Function

js

CopyEdit

```
const user = {
  username: "pranay",
  price: 199
};

function handleObject(anyObject) {
  console.log(`Username is ${anyObject.username} and price is ${anyObject.price}`);
}

handleObject(user);
```

✅ You can also pass object directly:

js

CopyEdit

```
handleObject({ username: "codeguy", price: 399 });
```

📋 6. Passing Array to Function

js

CopyEdit

```
const myNewArray = [200, 400, 600, 800];
```

```
function returnSecondValue(getArray) {
  return getArray[1];
}

console.log(returnSecondValue(myNewArray)); // 400
```

💡 You can access array values using index inside function.

🔁 **Real Life Example:**

🛒 Imagine you're buying items on Amazon:

- calculateCartPrice(199, 299, 399) → All prices go in a cart (array).
- handleObject({ username: "John", price: 999 }) → Represents a user checking out.