Object Type

- The concept of **object** into computer programming was introduced in early 1960 by "**Alan Kay**".
- It is used to encapsulate related type of data and logic.
- Data is defined in "Property".
- Logic is defined by using "Method/Function".
- 1967 **Johan Olay, Kristian Nygaard** introduced the concept of **Object-Oriented Programming** and formulated with "SIMULA 67".
- Object encapsulates related type of properties and functions into one reference memory.
- The object related members of configured within "{ }".
- TypeScript early versions don't have any pre-defined data type for handling object, it uses "any" as type for object.

```
let obj:any = {
        property:value,
        method: function() { }
```

```
}
obj.property;
obj.method();
```

- In early versions of JavaScript, we call object as “Pseudo Class”.
- In ES5 JavaScript introduced class-based programming.
- The latest versions of TypeScript introduced “object” as new type for handling objects.

```
let obj:object = {
        property:value,
        method:function(){}
}
```

- You can access the members of object within the object by using “this” keyword.
- You can access the members of object outside the object by using “object name”.
- By default, all objects in TypeScript have static behaviour.
- The properties of an object can handle any type of data, both primitive and non-primitive type.

- Object allows re-usability of properties and methods.

Ex:

```
let product:object = {
    Name: "Samsung TV",
    Price: 45600.55,
    InStock:true,
    ShippedTo:["Delhi","Hyd"],
    Qty:2,
    Total:function(){
        return this.Qty * this.Price;
    },
    Print:function(){
console.log(`Name=${this.Name}\nPrice=${this.Pri
ce}\nInStock=${(this.InStock)==true?"Available":
"Out of Stock"}\nQuantity=${this.Qty}\nShipped
To=${this.ShippedTo.toString()}\nTotal=${this.T
otal()}`);
    }
}
```

```
console.log(`--------TV Details-------`);
product.Print();
console.log(`-------Shoe Details-------`);
product.Name = "Nike Casuals";
product.Price = 2000.44;
product.Qty = 1;
product.ShippedTo = ["Mumbai","Chennai"];
product.InStock = false;
product.Print();
```

Array of Objects

- It is a collection of objects.
- There is no specific data type for Array of Object.
- You have to use “any” as type.
- The format of this data often represents “JSON”. JavaScript Object Notation.
- JSON formatted data is available offline.
- It saves round-trip [Make a request to server every time when data is required].
- It is native to client browser.
- It can run on any device.

Ex: Add a new file: products.json

```
[
   {
      "Name": "Samsung TV",
      "Price": 34000.44,
      "Category": "Electronics"
   },
   {
      "Name": "Nike Casuals",
      "Price": 4200.44,
      "Category": "Footwear"
   }
]
```

Ex: Configuring JSON type data and reading in TypeScript

```
let products:any = [
   {
      "Name": "Samsung TV",
```

```
        "Price": 34000.44,
        "Category": "Electronics"
    },
    {
        "Name": "Nike Casuals",
        "Price": 4200.44,
        "Category": "Footwear"
    }
];
for(var product of products)
{
    console.log(`${product.Name} -
${product.Price}`);
}
```

Ex: Destructing JSON type data.

```
let products:any = [
    {
        "Name": "Samsung TV",
        "Price": 34000.44,
```

```
        "Category": "Electronics"
    },
    {
        "Name": "Nike Casuals",
        "Price": 4200.44,
        "Category": "Footwear"
    }
];
let [tv, shoe] = products;
console.log(`-----TV Details------`)
console.log(`Name=${tv.Name}\nPrice=${tv.Price}
`);
console.log(`-----Shoe Details------`)
console.log(`Name=${shoe.Name}\nPrice=${shoe.
Price}`);
```

Ex: Filtering Data

```
let products:any = [
    {
```

```
    "Name": "Samsung TV",
    "Price": 34000.44,
    "Category": "Electronics"
},
{
    "Name": "Nike Casuals",
    "Price": 4200.44,
    "Category": "Footwear"
},
{
    "Name": "Mobile",
    "Price": 4200.44,
    "Category": "Electronics"
},
{
    "Name": "Jeans",
    "Price": 4200.44,
    "Category": "Fashion"
}
```

```
];
let electronics:any[] =
products.filter(function(product){
    return product.Category=="Electronics";
});
console.log(`Electronic Products:`);
for(var item of electronics)
{
    console.log(item.Name);
}
```

Ex: Find the total count of footwear products

```
let products:any = [
    {
        "Name": "Samsung TV",
        "Price": 34000.44,
        "Category": "Electronics"
    },
```

```
{
    "Name": "Nike Casuals",
    "Price": 4200.44,
    "Category": "Footwear"
},
{
    "Name": "Mobile",
    "Price": 4200.44,
    "Category": "Electronics"
},
{
    "Name": "Jeans",
    "Price": 4200.44,
    "Category": "Fashion"
},
{
    "Name": "Lee Boot",
    "Price": 4200.44,
    "Category": "Footwear"
```

```
    }
];
let footwearcount:number =
products.filter(function(product){return
product.Category=="Footwear"}).length;
console.log(`Total Footwear Products :
${footwearcount}`);
```