

1. <https://github.com/rvsp/typescript-oops/blob/master/Practice/Movie.md>

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    class movie
    {
      constructor(title,studio,rating)
      {
        this.title=title
        this.studio=studio
        if(rating==undefined||null)
        {
          this.rating="PG"
        }
        else
        {
          this.rating=rating
        }
      }
      getPG(arr)
      {
        console.log(arr)
        var pgResult=[];
        for(var i=0;i<arr.length;i++)
        {
          if(arr[i].rating=="PG")
            pgResult.push(arr[i])
        }
        console.log(pgResult)
      }
    }
    var film1=new movie("Casino Royale","Eon Productions","PG13")
    var film2=new movie("ABC Royale","Eon Productions")
    var film3=new movie("DEF Royale","Eon Productions","PG13")
    var film4=new movie("GHI Royale","Eon Productions","PG13")
    var film5=new movie("JKL Royale","Eon Productions")
    var film6=new movie("MNO Royale","Eon Productions","PG13")
    var film7=new movie("PQR Royale","Eon Productions","PG")
    const array=[film1,film2,film3,film4,film5,film6,film7];
```

```

        film1.getPG(array)
    </script>
</body>
</html>

```

OUTPUT:

The screenshot shows a web browser's developer console with two arrays of movie objects. The first array, labeled (7) [movie, movie, movie, movie, movie, movie, movie], contains 7 items. The second array, labeled (3) [movie, movie, movie], contains 3 items. Both arrays have a 'length' property and a '[[Prototype]]' property.

```

(7) [movie, movie, movie, movie, movie, movie, movie] i
  ▶ 0: movie {title: 'Casino Royale', studio: 'Eon Productions', rating: 'PG13'}
  ▶ 1: movie {title: 'ABC Royale', studio: 'Eon Productions', rating: 'PG'}
  ▶ 2: movie {title: 'DEF Royale', studio: 'Eon Productions', rating: 'PG13'}
  ▶ 3: movie {title: 'GHI Royale', studio: 'Eon Productions', rating: 'PG13'}
  ▶ 4: movie {title: 'JKL Royale', studio: 'Eon Productions', rating: 'PG'}
  ▶ 5: movie {title: 'MNO Royale', studio: 'Eon Productions', rating: 'PG13'}
  ▶ 6: movie {title: 'PQR Royale', studio: 'Eon Productions', rating: 'PG'}
  length: 7
  ▶ [[Prototype]]: Array(0)

(3) [movie, movie, movie] i
  ▶ 0: movie {title: 'ABC Royale', studio: 'Eon Productions', rating: 'PG'}
  ▶ 1: movie {title: 'JKL Royale', studio: 'Eon Productions', rating: 'PG'}
  ▶ 2: movie {title: 'PQR Royale', studio: 'Eon Productions', rating: 'PG'}
  length: 3
  ▶ [[Prototype]]: Array(0)

```

2. <https://github.com/rvsp/typescript-oops/blob/master/Practice/class-circle.md>

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>

    class Circle{
      constructor(radius, color) {
        this.radius = 1.0;
        this.color = "Red";
        if (typeof (radius) !== "undefined") {
          this.radius = radius;
        }
        if (typeof (color) !== "undefined") {
          this.color = color;
        }
      }
    }
  </script>

```

```

    }
    getRadius() {
        return this.radius;
    };
    setRadius(radius) {
        this.radius = radius;
    };
    getColor() {
        return this.color;
    };
    setColor(color) {
        this.color = color;
    };
    toString() {
        return "Radius: " + this.radius + " Color: " + this.color;
    };
    getArea() {
        return (this.radius * (Math.PI) * this.radius);
    };
    getCircumference() {
        return (2 * (Math.PI)* this.radius);
    };
    // area=pi*r*r
    //circum=2*pi*r
}

var c1 = new Circle();
console.log("Constructor with no params: " + c1.toString());
var c2 = new Circle(3.5);
console.log("Constructor with one param: " + c2.toString());
var c3 = new Circle(2.2, 'Yellow');
console.log("Constructor with two param: " + c3.toString());

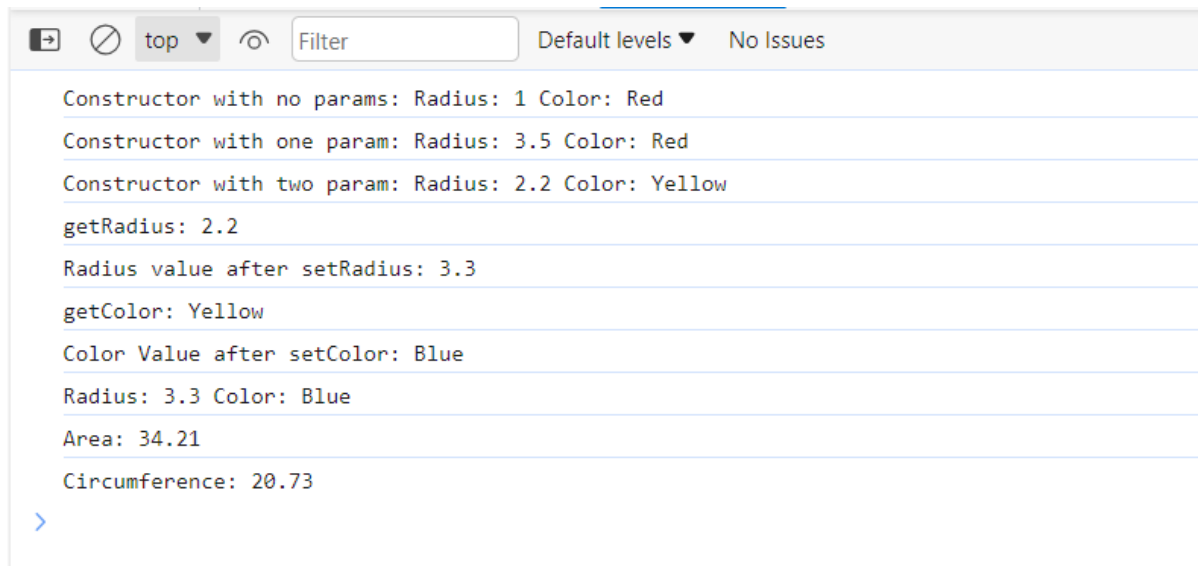
console.log("getRadius: " + c3.getRadius());
c3.setRadius(3.3);
console.log("Radius value after setRadius: " + c3.getRadius());

console.log("getColor: " + c3.getColor());
c3.setColor("Blue");
console.log("Color Value after setColor: " + c3.getColor());

console.log(c3.toString());
console.log("Area: " + c3.getArea().toFixed(2));
console.log("Circumference: " + c3.getCircumference().toFixed(2));
</script>
</body>
</html>

```

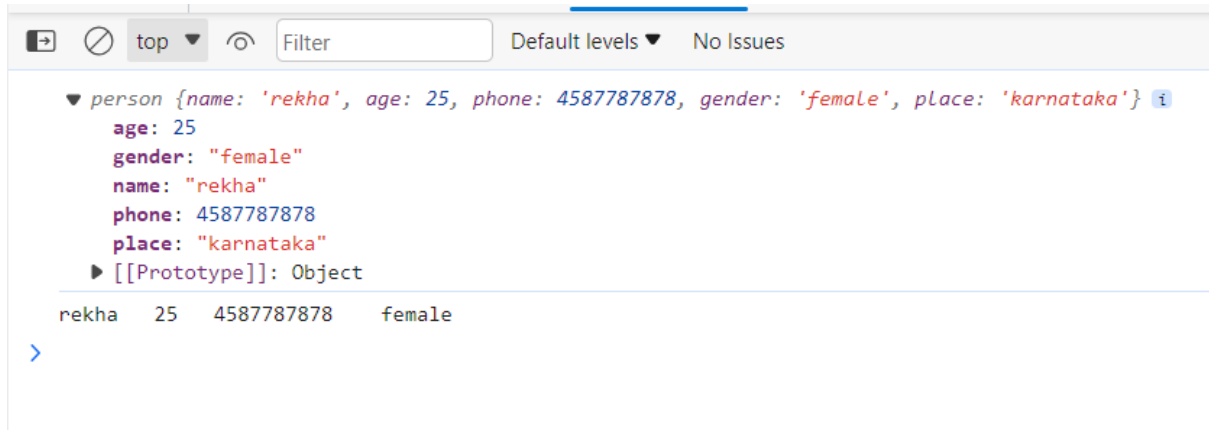
OUTPUT:



3. Write a “person” class to hold all the details.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    class person{
      constructor(name,age,phone,gender,place)
      {
        this.name=name;
        this.age=age;
        this.phone=phone;
        this.gender=gender;
        this.place=place;
      }
    }
    var s1=new person("rekha",25,4587787878,"female","karnataka");
    console.log(s1);
    let{name,age,phone,gender}=s1
    console.log(`${name}   ${age}   ${phone}   ${gender}`)
  </script>
</body>
</html>
```

OUTPUT:






4. write a class to calculate the Uber price.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    class uber{
      constructor(km)
      {
        this.km=km
        this.price=km*3
        console.log(this.km,this.price)
      }
      getPrice()
      {
        console.log(`Ride cost is ${this.price}`)
      }
      applyDiscount()
      {
        console.log("Discount calculation")
        let discount=(25/this.price)*100;
        let priceAfterDiscount=this.price-discount
        console.log(`As a privilege customer, you got discount and
price after discount is RS. ${priceAfterDiscount.toFixed()}`);
      }
    }
    var ride=new uber(30);
    ride.getPrice()
    ride.applyDiscount()
  </script>
</body>
```

```
</html>
```

OUTPUT:

  top ▼  Filter  Default levels ▼ No Issues

30 90

Ride cost is 90

Discount calculation

As a privilege customer, you got discount and price after discount is RS. 62

>