2)

```
1   try{
1)          let response = await fetch('data/iris.json');
2)          let data = await response.json();
3)          let possibleColor = ["#5d3fd3", "#a73fd3", "#d33fb5",
    "#d35d3f", "#d3a73f"];
4)
5)          let irisesWithColor = data.map(addColorToIris);
6)
7)          console.log(irisesWithColor);
8)          console.log(irisesWithColor[0]);
9)
10)         function addColorToIris(flower){
11)             let randomColor = possibleColor
    [Math.floor(Math.random()*possibleColor.length)];
12)             return {
13)                 ...flower,
14)                 color: randomColor
15)
16)             };
17)
18)
19)         }
20)
21)     } catch(err){
22)         console.log(err)
23)     }
```

3)

```
//3. Use a filter on irisesWithColor


  const filteredIrises = irisesWithColor.filter(function(flower){
     return flower.sepalWidth <4;
  });


  console.log("FilteredIris ", filteredIrises)
  //results:
```

4)

```
//4. calculate the average petalLenght using reduce


const totalPetalLength = irisesWithColor.reduce(function(sum, flower) {
    return sum + flower.petalLength;

}, 0);
const averagePetalLength = totalPetalLength / irisesWithColor.length;

console.log("Average Petal Lenght = ", averagePetalLength);
//results: 3.7580000000000027
```

5)

```
//5. Use Find to access an object whose petalWidth is under 1.0

const findIris = irisesWithColor.find(function(flower){
    return flower.petalWidth >1.0;

});
console.log("iris with petalWith > 1 = ", findIris);
//results: It finds one
```

6)

```
//6. Find an object that has a petal lenght over 10 using some

const petalOver10 = irisesWithColor.some(function(flower){
    return flower.petalLength >10;

});
console.log("flower with petalLength over 10", petalOver10);
//results: No, it doesn't exist
```

7)

```
//7. Find if there is an Iris with petal Lenght = 4.2
    const petalEqual42 = irisesWithColor.some(function(flower){
        return flower.petalLength === 4.2;

    });
    console.log("iris with petal equal to 4.2", petalEqual42);
    //results: Yes, it does exist
```

8)

```
//8. Does all objects have a petalWidths over 3
    const allPetalWidthsUnder3 = irisesWithColor.every(function(flower){
        return flower.petalWidth <3;

    });

    console.log("All iris have width under 3? ", allPetalWidthsUnder3);
    //results: Yes, it's true
```

9)

```
 //9. Find if all Iris have sepal width over 1.2

    const allSpetalWidthGreaterThen12 = irisesWithColor.every(function(flower){
        return flower.sepalWidth>1.2;
    });
    console.log("irises have sepal Width over 1.2", allSpetalWidthGreaterThen12);
    //results: Yes, it's true
```

10)

```
        //10. output an array sorted on the petalWidth field
```

```
    const irisesWithColorsSorted = irisesWithColor.toSorted(function(a, b){
        return a.petalWidth - b.petalWidth;



});

console.log("sorted by petalWidth(smallest to largest)",irisesWithColorsSorted);
//results:
```

11)

Visualisation: I want to visualize all the flowers with an increasing petalWidth from left to right of the webpage. And then, when you click on them, the color of the flower changes, which adds an interactive aspect to the whole visualisation.