Week 2 Notes

Ross Emile Aparece

Class 3
02/04/2025

- Standard syntax for an object is key value pairs

```
let myfrac = {num:1, den:4};
```

- Keys can be any type of string value but there are restrictions
- Do not need quotes for a property
- Values can be any data type

```
let myfrac = {
    num:1,
    den:4,
    toDecimal: function(){return this.num / this.den}
    };
```

- Functions are called using ()

```
myfrac.toDecimal;
returns f(){return this,num / this.den}

myfrac.toDecimal();
returns 0.25;
```

- Deconstructing assignment

```
let downloaded = {a:5, b:7, c:8, z:89};

console.log(b);
console.log(z);
```

- Spread operator converst from container type to parameter list

```
let arr1 = [1, 2, 3, 4 , 5];
let arr2 = [8, 9, 10]

[..arr1, 6, 7, ...arr2, 10, 11, 12,13]
returns (13) [1, 2, 3, 4, 5, 6, 7, 8, 9, 10 , 11, 12, 13]
```

- Informally removes the outermost braces and turns it into a comma seperated list
- Works with any container type, common use case:

```
let downloaded = {a:5, b:7, c:8, z:89}
let augmenteded = {...downloaded, d:100}
```

- Spread operator cannot be used anywhere

```
...arr1
let arr1 = [1, 2, 3, 4 , 5];
let arr2 = [8, 9, 10]

//but this works:

[...arr1]
let arr1 = [1, 2, 3, 4 , 5];
let arr2 = [8, 9, 10]
```

  - Assume downloaded is a massive array

```
Math.max(downloaded);
//Returns NaN because it wants a parameter list

Math.max(...downloaded);
//Converts the array to a parameter list allowing the
    function to work properly
```

- Rest operator ... (Same symbol as spread operator)
  - Context lets us determine the difference between the two
- Converts from parameter list to array
- Scenario: assume download is a massive array

```
let download = [5, 6, 7, 8, 9, 10, 11];
let [first,second, ...r] = downloaded;
```

```
console.log(first);
console.log(second);

console.log(r);
//returns the array containing the rest of the numbers
```

- Second use case:

```
//Use a rest operator for an infinite amount of numberes
function myMax(...n){
    let max_candidate = n[0];
    for(let i = 1; i < n.length; i++){
        if(n[i]< nax_candidate){
            max_candidate = n[i];
        }
    }
    return max_candidate;
}

myMax(1, 2, 3, 4, 5, 6, 7, 8, 9);
//Should return 9;
```

  - Allows a function to accept any amount of inputs

- Topic 2 start

```
let n1;
n1 = 31;
let n2;
n2 = n1;
n1 = 32;
console.log(n2);
```

Class 4
02/06/2025