ES6

DESTRUCTING

Dr. Jose Annunziato

Array Matching

- Flexible Array destructuring into variables during assignment
- ES5:

```
var list = [ 1, 2, 3 ];
var a = list[0], b = list[2];
var tmp = a; a = b; b = tmp;
```

- ES6:

```
const list = [ 1, 2, 3 ]
let [ a, , b ] = list       // a = list[0]; b = list[2]
[ b, a ] = [ a, b ]         // swapping a and b
```

Object Matching, Shorthand Notation

- Flexible Object destructuring into variables during assignment
- ES5

```
const tmp = {a: "123", b: "234", c: "345", d: "456"}
var a  = tmp.a;
var c = tmp.c;
```

- ES6

```
let { a, c } = tmp
```

Parameter Matching

- Consider the following ES5

```
function h (arg) {
 var name = arg.name
 var val  = arg.val
 console.log(arg.name, arg.val)
}
h({ name: "bar", val: 42 })
```

Parameter Matching

- In ES6 use shorthand parameter matching

```
const h = ({ name, val }) => console.log(name, val)
h({ name: "bar", val: 42 })
```

Order is irrelevant here

OR

```
const h = ({ val, name }) => console.log(name, val)
h({ name: "bar", val: 42 })
```

Renaming matched parameters

```
const h = ({ name : n, val : v }) => {
    console.log(n, v)
}
h({ name: "bar", val: 42 })
```

Also works with arrays

```
function f ([ name, val ]) {
    console.log(name, val)
}
f([ "bar", 42 ])
```

Property Shorthand

- If property names are same as values:

```
var x = 0, y = 0;
obj = { x: x, y: y };
```

- Use shorthand instead

```
obj = { x, y };
```