

# ES6 & ES6 CLASSES

EXPLORING ES6

ES6 CHEAT SHEET

# VAD VI REDAN BERÖRT

- Arrow function =>
- const & let

Tar nu upp några som jag tycker är användbara

# TEMPLATE LITERALS

"Backticks"

"Är det jQuery?"

jQuery har inte patent på \$\$\$\$

```
var name = "Ronny";  
console.log(`Hi, my name is ${name}`);
```

# TEMPLATE LITERALS

Shift + ' : Grav accent

A small topographic map showing a river, roads, and elevation contours with labels like 1:17, 1:120, and 1:122.

```
var a = 5;
```

```
var b = 10;
```

```
console.log(`${a} + ${b} equals ${ a + b }`);
```

```
//5 + 10 equals 15
```

```
var name = "Gnurufrr";
```

```
console.log(`I AM ${name.toUpperCase()}!!!!`);
```



# TEMPLATE LITERALS

Vi slipper oroa oss för att behöva escapea

```
var text = " \"double quotes\"";
```

```
var text = `"double quotes"`;
```

```
var text = 'I'm a noob';
```

```
var text = `I'm a noob`;
```

# TEMPLATE LITERALS

Behåller newline `\n`, ny rad

```
var html = `  
  <ul>  
    <li>Wowo this is great</li>  
    <li>Awesome</li>  
  </ul>`;
```

Soft att skriva html-block.

# DESTRUCTURING

Ref: ExploringJS - Destructing



```
var obj = {  
    x: 100,  
    y: 100  
};
```

```
var x = obj.x;
```

```
var y = obj.y;
```

# ES6 DESTRUCTURING

```
var obj = {  
  x: 100,  
  y: 100  
}
```

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

# DESTRUCTURING

När du vill plocka ut egenskaper eller värden från ett objekt eller array

Tänk på att det är viktigt med positionen i arrayen:

```
//Jumping over values  
var [x,,,y] = array;
```

**REST & SPREAD**

*The spread syntax allows an expression to be expanded in places where multiple arguments/elements or variables are expected.*

# REST OPERATOR

Ser exakt ut som spread operator

Skillnaden är bara hur `...` används

"Gör detta med resten av argumenten"

```
function restFunction(a, b, ...rest){  
  console.log(a,b, rest);  
}
```

# SPREAD/REST

*The spread operator (...) looks exactly like the rest operator, but is used inside function calls and Array literals (not inside destructuring patterns).*

Ref: ExploringJS - Spread

I JavaScript kan vi skicka med fler eller färre argument än parametrar.  
De överiga argumenten kommer bara att struntas i.

```
function add(a,b,c){  
  add a + b + c;  
}  
add(5,5); //Will work
```



# DEFAULT PARAMETERS

ES6 gör det även möjligt att skicka med default värden

```
function add(x = 0, y = 2) {  
  return x + y;  
}
```

```
add() // 2
```

```
add(2) // 4
```

```
add(2, 8) // 10
```

# ES6 CLASSES

Ref: ExploringJS - Classes

```
//class keyword
class Person{
    //explicit constructor method
    constructor(name, age){
        this.name = name;
        this.age = age;
    }
}
```

# CLASS

Kan liknas vid Constructor Pattern

Måste alltid skrivas med **class**

Måste alltid skapas med **new**

```
var newPerson = new Person("Ronny", 52);
```

# CLASS METHODS

```
class Person{  
    constructor(name, age){  
        this.name = name;  
        this.age = age;  
    }  
    getName(){  
        return this.name;  
    }  
    setName(newName){  
        this.name = newName;  
    }  
}
```

# GET & SET

```
class Person{  
    constructor(name, age){  
        this.name = name;  
        this.age = age;  
    }  
    get name(){  
        return this.name;  
    }  
    set name(newName){  
        this.name = newName;  
    }  
}
```

```
class Person{  
    set name(newName) {  
        if(newName !== ' '){  
            this.name = newName;  
        }  
        return 'You must supply a name!';  
    }  
}
```

# ES6 CLASSES

Tillför inget nytt egentligen

"Syntactical sugar"

Ärver med **extends**

Den klassen man ärver ifrån kan kallas på med **super**



# ES6 OBJECT METHOD SYNTAX

```
var obj = {  
  printStuff(){  
    console.log( 'Hey! ' )  
  }  
  printName(){  
    console.log( 'Hey again! ' );  
  }  
}
```

Objektets funktioner kan skrivas i stil med en klass metoder.

# ES6 OBJECT SHORTHAND

```
var name = "Gnurfur";  
var race = "Troll";  
var char = {  
  name: name,  
  race: race  
};
```

# ES6 OBJECT SHORTHAND

```
var name = "Gnurfur";  
var race = "Troll";  
var char = {  
  name,  
  race  
};
```