

**Static**

# Klass och instans

Vi har sett hur en **klass** är en mall som beskriver objekt (**instanser**).

Hittills har det alltid varit instanserna som har metoder eller egenskaper.

Men en klass, alltså själva mallen, kan också ha egenskaper och metoder. Dessa tillhör inte instanserna utan klassen. De kallas statiska egenskaper och statiska metoder (static).

```
class Player {  
  static allPlayers = [];  
  static getById (id) { // body of method }  
}
```

```
console.log(Player.allPlayers); // []
```

# Static

## Statiska egenskaper och metoder tillhör klassen, inte instanserna

```
class Player {  
  static allPlayers = [];  
  static getById (id) { // body of method }  
  
  constructor () {  
    // body of constructor  
  }  
  myTeams () {  
    // body of method  
  }  
}
```

```
let player1 = new Player({...});  
player1.myTeams(); // OK, metoden tillhör instanserna
```

```
let a = player1.allPlayers; // NOT OK!, statiska egenskaper och metoder tillhör klassen  
                             // a kommer att tilldelas undefined.
```

```
let b = Player.allPlayers; // OK! b kommer att tilldelas [] (i detta exempel)
```

```
let c = player1.myTeams(); // OK!, metoden tillhör instanserna
```

```
let d = Player.myTeams(); // NOT OK!, metoden tillhör instanserna, inte klassen  
                          // Det kommer att ge Error (Player.myTeams is not a function)
```

# Exempel

Det kan vara nödvändigt att hålla reda på alla instanser som skapas av en klass. Exempelvis av alla spelare.

Vi så tidigare att det inte finns någon inbyggd metod för att göra det, utan det måste kodas. Det är vanligt att man kodar det såhär:

```
class Player {  
  static allPlayers = [];  
  constructor (data) {  
    this.id = data.id;  
    etc. // name, ...  
  
    Player.allPlayers.push(this);  
  }  
}
```

Statisk egenskap där vi sparar alla instanser

Rad i konstruktören som ser till att varje ny instans sparas i Player.allPlayers

