

# Tabeller

En tabell er en mengde variabler av samme type som blir referert med et felles navn.

- En tabell kan bli deklartert som andre variabler med `[]` etter datatypen eller navnet. **Eksempel:** `int[] arr` eller `int arr[]`
- Alle variablene i en tabell har en indeks som tilsvarer plassen i tabellen variabelen er på. Indeksene starter fra og med 0.
- En tabell i Java kan brukes som en klassevariabel, lokalvariabel og som parameter i en metode (**Eks.** `void(int[] arr)`).
- Størrelsen på en tabell må spesifiseres med en *int*-verdi og ikke *long* eller *short*.

For å deklarere en tabell kan vi følge den generelle formen som følger:

```
T navn[];  
ELLER  
T[] navn;  
// Hvor T er datatypen, f.eks. int
```

Eks.

```
int intArr[];  
int[] intArr;
```

Selv om vi har deklartert `intArr` som en variabel, finnes det faktisk ikke en tabell enda. For å linke `intArr` med en faktisk, fysisk tabell av tall, må vi instansiere en tabell og tildele den til `intArr` som følger:

```
int intArr[];  
intArr = new int[størrelse];  
ELLER  
int[] intArr = new int[størrelse];
```

- Elementene i en tabell opprettet med `new` vil bli initialisert til 0 (for numeriske typer), false (for boolean) eller *null* (for referansetyper/objekter).
- Tabeller kan også bli deklartert med verdier hvis disse er kjent når tabellen skal opprettes. **Eksempel:**  
`int[] intArr = new int[]{1, 2, 3, 4, 5};`

Hvor lengden av tabellen blir bestemt av antall elementer vi oppretter det med. I tilfellet over vil lengden være 5.

## Hente elementer fra tabellen

Hvert element i en tabell kan bli hentet via indeksen det har i tabellen. **Eksempel:**

```
// Henter ut elementet med indeks 4
int element = intArr[4];
```

Indeksen begynner på 0 og slutter på (lengde-1). Dette er fordi indeksen begynner på 0, men når tabellen telles starter telleren på 1. Vi kan hente ut alle elementene fra en tabell med *for-løkker*. Vi kan gjøre dette både med en vanlig *for-løkke* og med en *enhanced for-løkke*.

### For-løkke

```
int[] intArr = new int[]{1, 2, 3, 4, 5};

for(int i = 0; i < intArr.length; i++) {
    System.out.print(intArr[i]);
}
// 12345
```

### Enhanced for-løkke

```
int[] intArr = new int[]{1, 2, 3, 4, 5};

for(int element : intArr) {
    System.out.print(element);
}
//12345
```

### Tabeller med objekter

Vi kan ha tabeller som inneholder referanser til objekter. Vi bruker disse slik som vi bruker tabeller av hvilken som helst annen type. **Eksempel:**

```
class Student {
    public int id;
    public String name;

    Student(int id, String name) {
        this.id = id;
        this.name = name;
    }
}
```

```
class Main {  
    public static void main() {  
        Student[] studArr = new Student[5];  
        arr[0] = new Student(1, "Leonardo");  
        arr[1] = new Student(2, "Vincent");  
        arr[2] = new Student(3, "Mona");  
        arr[3] = new Student(4, "Lisa");  
        arr[4] = new Student(5, "Georg");  
  
        for(int i = 0; i < studArr.length; i++) {  
            System.out.println(studArr[i].id + studArr[i].name);  
        }  
    }  
}
```

Siden `studArr[i]` er en referanse til et objekt, kan vi bruke det som vi bruker et hvilket som helst objekt, og dermed også hente ut instansvariablene.