

## Typen Char

Characters är alltid 16 bitar.

Möjliga värden: Enskilda tecken.

```
char c1, c2;  
c1 = 'A';  
c2 = 'é';
```

### Escape-sekvenser

```
"\"  
\"  
\"  
\"t (tabulator)  
\"n (ny rad)  
\"O \"33\" \"266\" (Oktalt)  
\"u2663\" \"u03A8\" (Unicode, hexadecimalt) (http://Unicode.org)
```

### Jämförelser

```
if (c1 == '%')  
    System.out.println("Procent");  
if (c1 >= '0' && c1 <= '9')  
    System.out.println("Siffra");
```

## Objekt!

### Exempel TidDemo

```
public class TidDemo {  
    public static void main(String[] arg) {  
        Tidpunkt t1 = new Tidpunkt();  
        Tidpunkt t2 = new Tidpunkt();  
        int a = 17, b = 8, c = 20;  
        t1.sätt(a, b, c);  
        t2.sätt(23, 59, 59);  
        t1.tick();  
        t2.tick();  
        t2.tick();  
        String s1 = t1.toString();  
        String s2 = t2.toString();  
        System.out.println(s1);  
        System.out.println(s2);  
    }  
}
```

*new skapar minnesutrymme*

*Alla klasser har automatiskt metoden toString.*

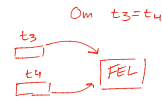
*println anropar automatiskt toString*

### Ex. Klassen "Tidpunkt"

```
public class Tidpunkt {  
    // instansvariabler  
    private int t, m, s;  
    private boolean visaSek = true;  
  
    // konstruktörer = initieringsmetoder  
    public Tidpunkt() {}  
  
    public Tidpunkt(int tim, int min, int sek) {  
        sätt(tim, min, sek);  
    }  
  
    public Tidpunkt(int tim, int min) {  
        this(tim, min, 0);  
        visaSek = false;  
    }  
  
    public Tidpunkt(Tidpunkt tp) {  
        t=tp.t; m=tp.m; s=tp.s; visaSek = tp.visaSek;  
    }  
  
    // metoder  
    public void sätt (int tim, int min, int sek) {  
        if (tim >= 0 && tim < 24 &&  
            min >= 0 && min < 60 &&  
            sek >= 0 && sek < 60) {  
            t = tim; m = min; s = sek;  
        }  
        else  
            System.out.println("Felaktig tidpunkt!");  
    }  
  
    public void sättVisaSek(boolean visa) {  
        visaSek = visa;  
    }  
  
    public int avläsTim () {  
        return t;  
    }  
}
```

*this timme i en konstruktor innebär: anropar en annan konstruktor. i detta fall de konstruktörerna OVAN*

*Kopieringskonstruktor*  
*Tidpunkt t3 = new Tidpunkt(11, 13, 30);*  
*Tidpunkt t4 = new Tidpunkt(t3);*



```

    }

    public int avläsMin () {
        return m;
    }

    public int avläsSek () {
        return s;
    }

    @Override
    public String toString () {
        String tid = t + ":" + m;
        if (visaSek)
            tid = tid + ":" + s;
        return tid;
    }

    public void ticka() {
        s = s+1;
        if (s == 60) {
            s = 0;
            m = m+1;
        }

        if (m == 60) {
            m = 0;
            t = t+1;
        }

        if (t == 24)
            t = 0;
    }
}

```