



Denne guide er oprindeligt udgivet på Eksperten.dk

Basal type konvertering i Java

Denne artikel beskriver hvordan man konverterer mellem forskellige typer i Java.

Den forudsætter kun minimalt kendskab til Java og henvender sig primært til dem uden den store Java erfaring.

Skrevet den **03. Feb 2009** af **arne_v** | kategorien **Programmering / Java** | ★★☆☆☆

Historie:

V1.0 - 31/01/2004 - original

byte/short/int/long -> float/double

Brug simpelt assignment.

Eksempel:

```
int iv = 123;
double xv2;
xv2 = iv;
```

byte/short/int/long -> boolean

Brug assignment af resultat af !=0.

Eksempel:

```
int iv = 123;
boolean bv2;
bv2 = (iv != 0);
```

byte/short/int/long -> char

Brug cast og assignment.

Eksempel:

```
int iv = 123;
char cv2;
cv2 = (char)iv;
```

byte/short/int/long -> String

Brug den statiske toString metode på det tilhørende objekt.

Eksempel:

```
int iv = 123;
```

```
String sv2;  
sv2 = Integer.toString(iv);
```

float/double -> byte/short/int/long

Brug cast og assigment. Bemærk at der nedrundes (og decimaler tabes naturligvis).

Eksempel:

```
double xv = 123.456;  
int iv2;  
iv2 = (int)xv;
```

float/double -> String

Brug den statiske toString metode på det tilhørende objekt.

Eksempel:

```
double xv = 123.456;  
String sv2;  
sv2 = Double.toString(xv);
```

boolean -> int

Brug ?: operator.

Eksempel:

```
boolean bv = true;  
int iv2;  
iv2 = (bv ? 1 : 0);
```

boolean -> String

Brug toString metode på det tilhørende objekt.

Eksempel:

```
boolean bv = true;  
String sv2;  
sv2 = (new Boolean(bv)).toString();
```

char -> int

Brug simpelt assigment.

Eksempel:

```
char cv = 'A';  
int iv2;
```

```
iv2 = cv;
```

char -> String

Brug constructor med array initializer.

Eksempel:

```
char cv = 'A';  
String sv2;  
sv2 = new String(new char[] { cv });
```

String -> byte/short/int/long

Brug statiske parse metode i det tilhørende objekt.

Eksempel:

```
String sv = "123";  
int iv2;  
iv2 = Integer.parseInt(sv);
```

String -> float/double

Brug statiske parse metode i det tilhørende objekt.

Eksempel:

```
String sv = "123.456";  
double xv2;  
xv2 = Double.parseDouble(sv);
```

String -> boolean

Brug constructor og booleanValue metode i det tilhørende objekt.

Eksempel:

```
String sv = "true";  
boolean bv2;  
bv2 = Boolean.valueOf(sv).booleanValue();
```

String -> char

Brug charAt metoden.

Eksempel:

```
String sv = "ABC";  
char cv2;  
cv2 = sv.charAt(0);
```

int -> String (som hex)

Brug statiske toHexString metode i tilhørende objekt.

Eksempel:

```
int iv = 123;
String sv2;
sv2 = Integer.toHexString(iv);
```

String (som hex) -> int

Brug statiske parseInt metode i tilhørende objekt med angivelse af radix.

Eksempel:

```
String sv = "7b";
int iv2;
iv2 = Integer.parseInt(sv, 16);
```

byte[] -> String

Brug constructor med angivelse af karakter sæt.

Eksempel:

```
byte[] b = { 65, 66, 67 };
String s2;
s2 = new String(b, "ISO-8859-1");
```

String -> byte[]

Brug getBytes metoden med angivelse af karakter sæt.

Eksempel:

```
String s = "abc";
byte[] b2;
b2 = s.getBytes("ISO-8859-1");
```

byte/short/int/long/float/double/boolean/char -> Byte/Short/Integer/Long/Float/Double/Boolean/Character

Brug constructor.

Eksempel:

```
int iv = 123;
Integer iv2;
iv2 = new Integer(iv);
```

**Byte/Short/Integer/Long/Float/Double/Boolean/Character ->
byte/short/int/long/float/double/boolean/char**

Brug xxxValue metode.

Eksempel:

```
Integer iv = new Integer(123);  
int iv2;  
iv2 = iv.intValue();
```

java.util.Calendar -> java.util.Date

Brug getTime metoden.

Eksempel:

```
java.util.Date d;  
java.util.Calendar cal = new java.util.GregorianCalendar();  
d = cal.getTime();
```

java.util.Date -> java.util.Calendar

Brug setTime metoden.

Eksempel:

```
java.util.Date d = new java.util.Date();  
java.util.Calendar cal = new java.util.GregorianCalendar();  
cal.setTime(d);
```

java.util.Date -> String

Brug SimpleDateFormat og format.

Eksempel:

```
java.util.Date d = new java.util.Date();  
java.text.DateFormat df = new java.text.SimpleDateFormat("dd-MMM-yyyy  
hh:mm");  
String ds;  
ds = df.format(d);
```

String -> java.util.Date

Brug SimpleDateFormat og parse.

Eksempel:

```
String ds = "31-Jan-2004 09:24"  
java.util.Date d;  
java.text.DateFormat df = new java.text.SimpleDateFormat("dd-MMM-yyyy  
hh:mm");
```

```
d = df.parse(ds);
```

java.util.Date -> java.sql.Date/java.sql.Time/java.sql.Timestamp

Brug constructor og getTime metode..

Eksempel:

```
java.util.Date d = new java.util.Date();
java.sql.Timestamp ts;
ts = new java.sql.Timestamp(d.getTime());
```

java.sql.Date/java.sql.Time/java.sql.Timestamp -> java.util.Date

Brug simpelt assignment.

Eksempel:

```
java.sql.Timestamp ts = new java.sql.Timestamp((new Date()).getTime());
java.util.Date d;
d = ts;
```

Kommentar af simonvalter d. 03. Feb 2004 | 1

rigtigt godt opslags værk, noget jeg kunne have haft brug for mange gange.

Kommentar af bgo d. 27. May 2004 | 2

just what the doctor prescribed :)

stod lige og manglede dato-konverteringer, og løber generelt ind i konverteringsproblemer

Kommentar af digitalsoul d. 06. May 2004 | 3

Kommentar af gladmhensk d. 10. Feb 2004 | 4

Den røg lige til printeren :) Altid dejligt når man ikke kan huske hvor ditten og datten skal placeres :) Keep up the good work