

# Cheat Sheet: Conversión de Tipos (Casting) en Java

## 1. Conversiones Numéricas Primitivas

### Conversión Automática, implícita (Widening Casting)

Ocurre al pasar de un tipo de dato pequeño a uno más grande. Es segura y no pierde información. **Jerarquía:**  
`byte -> short -> int -> long -> float -> double`

```
int numeroEntero = 100;
double numeroDecimal = numeroEntero; // Se convierte automáticamente a 100.0
```

### Conversión Manual, explícita (Narrowing Casting)

Se debe forzar al pasar de un tipo grande a uno más pequeño. **Puede haber pérdida de datos. Sintaxis:**  
`tipo_destino = (tipo_destino) valor_origen;`

```
double numeroDecimal = 9.78;
int numeroEntero = (int) numeroDecimal; // Resultado: 9 (se pierde la parte decimal)

long numeroLargo = 150L;
byte numeroByte = (byte) numeroLargo; // Riesgo de desbordamiento si el long es muy grande
```

## 2. Conversiones con char e int

Se basan en el valor numérico (Unicode/ASCII) de los caracteres.

Conversión	Descripción	Ejemplo de Sintaxis
char a int	(Automática) Obtiene el código numérico del carácter.	char letra = 'A'; int codigo = letra; // 65
int a char	(Manual) Obtiene el carácter correspondiente a un código.	int codigo = 65; char letra = (char) codigo; // 'A'

## 3. Conversión DE String a Otros Tipos

**¡Importante!** No se puede usar casting directo como `(int) "123"`. Se deben usar los métodos `parse` de las clases envoltorio (`Integer`, `Double`, etc.).

Tipo Destino	Método	Ejemplo de Sintaxis
<code>int</code>	<code>Integer.parseInt(str)</code>	<code>int num = Integer.parseInt("123");</code>
<code>double</code>	<code>Double.parseDouble(str)</code>	<code>double dec = Double.parseDouble("45.67");</code>
<code>long</code>	<code>Long.parseLong(str)</code>	<code>long l = Long.parseLong("1234567890");</code>
<code>float</code>	<code>Float.parseFloat(str)</code>	<code>float f = Float.parseFloat("12.34f");</code>

**Advertencia:** Si el `String` no contiene un número válido, se producirá una `NumberFormatException`.

## 4. Conversión A `String` desde Otros Tipos

Método 1: `String.valueOf()` (Recomendado)

Convierte explícitamente cualquier tipo de dato a su representación en `String`.

```
int numero = 500;
String texto = String.valueOf(numero); // Resultado: "500"

double decimal = 42.5;
String textoDecimal = String.valueOf(decimal); // Resultado: "42.5"
```

Método 2: Concatenación (Atajo)

El truco más rápido. Unir cualquier dato con una cadena vacía lo convierte a `String`.

```
int numero = 100;
String texto = "" + numero; // Resultado: "100"
```

## 5. Extracción de `String` a `char`

No es una conversión directa, sino la **extracción** de un carácter de la cadena en una posición específica usando el método `.charAt(índice)`.

```
String palabra = "Java";
char primerCaracter = palabra.charAt(0); // Resultado: 'J'
char tercerCaracter = palabra.charAt(2); // Resultado: 'v'
```