

Cheat Sheet de Operador Ternario, String, Math y Character

Operador Ternario

Es una forma compacta de la estructura `if-else` para asignar un valor a una variable en una sola línea.

Estructura

Concepto	Sintaxis
Estructura Básica	<code>variable = (condicion) ? valor_si_true : valor_si_false;</code>
Ejemplo (Par/Impar)	<code>String resultado = (num % 2 == 0) ? "Par" : "Impar";</code>
Ejemplo (Mayor/Menor)	<code>int mayor = (a > b) ? a : b;</code>
Anidado (No recomendado)	<code>String res = (n > 0) ? "Positivo" : ((n < 0) ? "Negativo" : "Cero");</code>

String

La clase `String` gestiona cadenas de texto. Es **immutable**, lo que significa que cada modificación crea un nuevo objeto.

Creación

```
// Recomendado (usa el "String Pool")
String s1 = "Hola";

// Crea un nuevo objeto siempre
String s2 = new String("Hola");
```

Métodos Comunes

Método	Descripción	Sintaxis de Ejemplo
<code>length()</code>	Devuelve el número de caracteres.	<code>int len = "Hola".length(); // 4</code>
<code>equals(obj)</code>	Compara si el contenido es idéntico (sensible a mayúsculas).	<code>"a".equals("A"); // false</code>
<code>equalsIgnoreCase(str)</code>	Compara el contenido ignorando mayúsculas/minúsculas.	<code>"a".equalsIgnoreCase("A"); // true</code>

Método	Descripción	Sintaxis de Ejemplo
<code>concat(str)</code>	Concatena una cadena al final.	<code>"Hola".concat(" Mundo"); // "Hola Mundo"</code>
<code>toUpperCase()</code>	Convierte toda la cadena a mayúsculas.	<code>"texto".toUpperCase(); // "TEXT0"</code>
<code>toLowerCase()</code>	Convierte toda la cadena a minúsculas.	<code>"TEXT0".toLowerCase(); // "texto"</code>
<code>substring(begin, end)</code>	Extrae una subcadena. El índice <code>end</code> no se incluye.	<code>"Hola Mundo".substring(0, 4); // "Hola"</code>
<code>indexOf(str)</code>	Devuelve el índice de la primera aparición (-1 si no lo encuentra).	<code>"Hola".indexOf("la"); // 2</code>
<code>contains(str)</code>	Devuelve <code>true</code> si la cadena contiene la subcadena.	<code>"Hola".contains("ol"); // true</code>
<code>replace(old, new)</code>	Reemplaza todas las apariciones de un carácter o cadena.	<code>"banana".replace('a', 'o'); // "bonono"</code>
<code>trim()</code>	Elimina espacios en blanco al principio y al final.	<code>" hola ".trim(); // "hola"</code>
<code>split(regex)</code>	Divide la cadena en un array usando un delimitador.	<code>"a,b,c".split(","); // ["a", "b", "c"]</code>
<code>String.valueOf(dato)</code>	Convierte cualquier tipo de dato a <code>String</code> .	<code>String.valueOf(123); // "123"</code>

StringBuilder

Alternativa **mutable** y eficiente a `String` para cuando se necesitan realizar muchas modificaciones.

Creación

```
StringBuilder sb = new StringBuilder("Texto inicial");
```

Métodos Comunes

Método	Descripción	Sintaxis de Ejemplo
<code>append(valor)</code>	Añade un valor (texto, número, etc.) al final.	<code>sb.append("!");</code>
<code>insert(offset, valor)</code>	Inserta un valor en una posición específica.	<code>sb.insert(0, "Inicio: ");</code>
<code>replace(start, end, str)</code>	Reemplaza los caracteres en un rango por otra cadena.	<code>sb.replace(0, 5, "Adiós");</code>

Método	Descripción	Sintaxis de Ejemplo
<code>delete(start, end)</code>	Elimina los caracteres en un rango específico.	<code>sb.delete(0, 4);</code>
<code>reverse()</code>	Invierte el orden de los caracteres.	<code>sb.reverse();</code>
<code>toString()</code>	Convierte el <code>StringBuilder</code> a un <code>String</code> final.	<code>String resultado = sb.toString();</code>

Math

Proporciona métodos estáticos para operaciones matemáticas comunes. No necesita ser instanciada.

Funciones Principales

Método	Descripción	Sintaxis de Ejemplo
<code>Math.abs(x)</code>	Devuelve el valor absoluto.	<code>Math.abs(-10); // 10</code>
<code>Math.ceil(x)</code>	Redondea hacia arriba al entero más cercano.	<code>Math.ceil(5.1); // 6.0</code>
<code>Math.floor(x)</code>	Redondea hacia abajo al entero más cercano.	<code>Math.floor(5.9); // 5.0</code>
<code>Math.round(x)</code>	Redondea al entero más cercano (redondeo estándar).	<code>Math.round(5.5); // 6</code>
<code>Math.max(a, b)</code>	Devuelve el mayor de dos números.	<code>Math.max(10, 20); // 20</code>
<code>Math.min(a, b)</code>	Devuelve el menor de dos números.	<code>Math.min(10, 20); // 10</code>
<code>Math.pow(base, exp)</code>	Calcula la potencia de un número.	<code>Math.pow(2, 3); // 8.0</code>
<code>Math.sqrt(x)</code>	Calcula la raíz cuadrada.	<code>Math.sqrt(16); // 4.0</code>
<code>Math.random()</code>	Devuelve un <code>double</code> aleatorio entre <code>0.0</code> (incluido) y <code>1.0</code> (excluido).	<code>Math.random(); // 0.123...</code>
<code>Math.sin(rad)</code>	Seno de un ángulo en radianes.	<code>Math.sin(Math.PI / 2); // 1.0</code>
<code>Math.cos(rad)</code>	Coseno de un ángulo en radianes.	<code>Math.cos(Math.PI); // -1.0</code>
<code>Math.log(x)</code>	Logaritmo natural (base <code>e</code>).	<code>Math.log(Math.E); // 1.0</code>

Constantes

- `Math.PI`: El valor de Pi (3.14159...).

- **Math.E**: La base de los logaritmos naturales (2.71828...).

Character

Clase envoltorio (**wrapper**) que proporciona métodos estáticos para manipular el tipo primitivo **char**.

Métodos de Verificación

Método	Descripción	Sintaxis de Ejemplo
isLetter(ch)	true si el carácter es una letra.	<code>Character.isLetter('a');</code> // <code>true</code>
isDigit(ch)	true si el carácter es un dígito.	<code>Character.isDigit('5');</code> // <code>true</code>
isWhitespace(ch)	true si es un espacio en blanco (' ', '\t', '\n').	<code>Character.isWhitespace(' ');</code> // <code>true</code>
isLetterOrDigit(ch)	true si es una letra o un dígito.	<code>Character.isLetterOrDigit('\$');</code> // <code>false</code>

Métodos de Conversión

Método	Descripción	Sintaxis de Ejemplo
toUpperCase(ch)	Convierte el carácter a mayúscula.	<code>Character.toUpperCase('c');</code> // <code>'C'</code>
toLowerCase(ch)	Convierte el carácter a minúscula.	<code>Character.toLowerCase('D');</code> // <code>'d'</code>
getNumericValue(ch)	Devuelve el valor int de un carácter numérico.	<code>Character.getNumericValue('7');</code> // <code>7</code>