Tipos de dato		
int	Números enteros	int num = 1;
double	Números reales (con decimales)	double dob = 2.45;
boolean	Verdadero o falso	boolean cond = true; boolean cond2 = false;
char	Caracteres	char c = 'A';
String	Cadenas de caracteres	String cadena = "Hola"; String cadena = new String("Hola");

Operadores Aritméticos		
+	Suma cuando se usa entre datos de tipo numérico. Concatenación cuando se usa entre String.	1 + 5 = 6 "Hola" + "Mundo" => HolaMundo
-	Resta	5 – 2 = 3
*	Multiplicación	2*3 = 6
/	División ¡Atención al dividir por 0!	10/2 = 5 10/0 = error.
%	Resto de una división	10 % 5 = 0 Explicación 10 / 5 = 2 con **resto 0**

Operadores lógicos (para comparar)		
==	**Igual que.** Compara si dos variables SON IGUALES. True si son iguales. False si son diferentes	2 == 2 -> true 2 == 3 -> false true == true -> true false == false -> true true == false -> false
!=	**Distinto que.** Compara si dos variables SON DIFERENTES. True si son diferentes. False si son iguales	2 != 2 -> false 2 != 3 -> true true != true -> false false != false -> false true != false -> true
>	**Mayor que.** Compara si la variable a la izquierda es mayor que la variable a la derecha True si es mayor. False si no lo es.	2 > 1 -> true 2 > 3 -> false 2 > 2 -> false
<	**Menor que.** Compara si la variable a la izquierda es menor que la variable a la derecha True si es menor. False si no lo es.	2 < 1 -> false 2 < 3 -> true 2 < 2 -> false
>=	**Mayor o igual que.** Compara si la variable a la izquierda es mayor o igual que la variable a la derecha True si es mayor o igual False si no lo es.	2 >= 1 -> true 2 >= 3 -> false 2 >= 2 -> true
<=	**Menor o igual que.**	2 <= 1 -> false

	Compara si la variable a la izquierda es menor o igual que la variable a la derecha True si es menor o igual. False si no lo es.	2 <= 3 -> true 2 <= 2 -> true

Operadores lógicos complejos		
&	AND estricto. Devuelve true si ambas condiciones son true. **Evalúa ambas condiciones siempre.**	2 == 2 & 4 > 2 -> true 2 == 2 & 4 > 4 -> false true == true & 4>= 4 -> true (evalúa ambas condiciones siempre)
&&	AND con cortocircuito Devuelve true si ambas condiciones son true. **Si encuentra una condición a false, no evalúa el resto de condiciones**	**2 == 2 && 4 > 2** -> true **2 == 2 && 4 > 4** -> false **2 != 2** && 4>= 4 -> false (evalúa sólo lo que está en negrita)
|	OR estricto Devuelve true si una u otra condición es true. Devuelve false si todas las condiciones son false **Evalúa ambas condiciones siempre.**	2 == 2 | 4 > 2 -> true 2 == 2 | 4 > 4 -> true 2 != 2 | 4>= 4 -> true 2 != 2 | 4 > 4 -> false (evalúa ambas condiciones siempre)
||	OR con cortocircuito Devuelve true si una u otra condición es true. Devuelve false si todas las condiciones son false **Cuando encuentra una condición a true, no evalúa el resto de condiciones.**	**2 == 2** || 4 > 2 -> true **2 == 2** || 4 > 4 -> true **2 != 2 || 4>= 4** -> true **2 != 2 || 4 > 4** -> false (evalúa sólo lo que está en negrita)
!	Not El operador de negación evalúa un solo operando con valor lógico y devuelve como resultado el valor de este invertido. Es decir, si el operando tiene valor true este operador devolverá false y viceversa.	!(2 == 2) -> false !true -> false !(2!=2) -> true !false -> true

Operadores de asignación		

=	Para asignar un valor a una variable	int a;
		a = 1;
		int b;
		b = 70;
		String cadena;
		cadena = “Hola”;

+=	Para sumar o concatenar un valor a una variable, y guardarlo en esa variable.	int a = 1;
		int b = 2;
		a += b; //a = a + b; -> a sería 3
		b += 1; // b = b + 1; -> b sería 3
		
		String cad = “Hola”;
		String cad2 = “Mundo”;
		cad += cad2; // cad sería “HolaMundo”

-=	Para restar un valor a una variable y asignarlo a esa variable	int a = 2;
		int b = 1;
		a -= b; //a = a - b; -> a sería 1
		b -= 1; // b = b - 1; -> b sería 0

*=	Para multiplicar un valor a una variable y asignarlo a esa variable	int a = 2;
		int b = 3;
		a *= b; //a = a * b; -> a sería 6
		b *= 4; // b = b * 4; -> b sería 12

/=	Para dividir un valor a una variable y asignarlo a esa variable	int a = 4;
		int b = 2;
		a /= b; //a = a / b; -> a sería 2
		b /= 2; // b = b / 2; -> b sería 1

%=	Para dividir un valor a una variable y asignar el valor de su resto a esa variable	int a = 4;
		int b = 2;
		a %= b; //a = a % b; -> a sería 0
		b %= 2; // b = b % 2; -> b sería 0

Operadores incrementales		

++	Incrementa en 1 el valor de esa variable	int i = 0;
		i++;
		//i valdría 1

--	Decrementa en 1 el valor de esa variable	int j = 2;
		j--;
		//j valdría 1

Impresión por pantalla	
System.out.println();	Imprime un mensaje e inserta una nueva línea.
System.out.print();	Imprime un mensaje pero NO inserta una nueva línea

Pedir datos por teclado al usuario	
Scanner scan = new Scanner(System.in); String cadena = scan.nextLine();	Lectura de una cadena de caracteres por teclado
Scanner scan = new Scanner(System.in); String cadena = scan.nextInt();	Lectura de un dato numérico por teclado

Estructura condicionales		

if	if([condición]) {
	[cuerpo]
	}
		Si la [condición] se evalúa a true, entonces se ejecuta el [cuerpo].

if else	if([condición]) {
	[cuerpoIf]
	} else {
	[cuerpoElse]
	}
		Si la [condición] se evalúa a true, entonces se ejecuta el [cuerpoIf], si no, se ejecuta el [cuerpoElse]

if else if	if([condición1]) {
	[cuerpoIf1]
	} else if([condición2]){
	[cuerpoIf2]
	}
		Si la [condición1] se evalúa a true, entonces se ejecuta el [cuerpoIf1], si no, si la [condición2] se evalúa a true, se ejecuta el [cuerpoIf2]

switch	switch ([variable]) {
	case [opc1]:
	[cuerpo]
	break;
	case [opc2]:
	[cuerpo]
	break;
	default:
	[cuerpo]
	break;
	}
		Se evalúa la [variable] indicada en el switch con las diferentes opciones [opc] de los case.
		Se ha de añadir un break dentro de cada case.
		
		Default. Si no se da ninguna de las opciones, se va a la opción por defecto default.

Bucles		

while	while ([condicion]) {
	[cuerpo]
	}
		MIENTRAS que la condición se cumpla, entonces se ejecuta el [cuerpo] una y otra vez.

for	for (int i=0; i<10; i++) {
	[cuerpo]
	}
	
	for ([variable auxiliar]; [condición]; [incremento de la variable]) {
	[cuerpo]
	}
		Mientras que la [condición] se cumpla, se ejecuta el cuerpo.
		El for es menos susceptible que el while a los bucles infinitos, pero hay que tener cuidado.