

Impresión en consola		
System.out. <b>print</b> ("");	System.out. <b>println</b> ("");	Imprime sin salto de línea y con salto de línea
System.out. <b>printf</b> ("Cadena formato %d" , argument);		Imprime con formato
System.out. <b>printf</b> ("%1.2f", 12500045465.445645665);		12500045465,45

Enteros	byte	short	int	long (L)
Decimales	float (F)	double		
Otros	char	boolean		

Comentarios	/*... diversas líneas ...*/	/**... documentación ...*/	//... una sola línea
-------------	-----------------------------	----------------------------	----------------------

Variables constantes	final double PI = 3.1415926535897932384626433832795;
----------------------	--

*, /, %, +, -	Operadores aritméticos	a * b , a / b , a % b , a + b , a - b
=, +=, -=, *=, /=, %=	Operadores de asignación	a = b , a += b , a -= b , a *= b , a / b , a % b (a=a % b)
==, !=, <, >, <=, >=	Operadores de comparación	a == b , a != b , a < b , a > b , a <= b , a >= b
++, --, (tipo) expr	Operadores especiales	a++, a-- (post); ++a, --a (pre) , a = (int) b
&&,   , !		True = si las 2 expresiones son true , True = si una expresión es true , Lo contrario que

Clase Scanner - (import java.util.Scanner) - (métodos no estáticos)		
nextInt	int i = scr. <b>nextInt</b> ();	int var1 = <b>Scanner</b> scr = <b>new Scanner</b> (System.in);
nextDouble	double d = scr. <b>nextDouble</b> ();	scr. <b>nextInt</b> ();
nextLine	String str = scr. <b>nextLine</b> ();	// Vacía buffer entrada teclado antes de usar nextLine scr. <b>nextLine</b> ();

Clase GregorianCalendar - (import java.util.GregorianCalendar)		
DATE	25	<b>GregorianCalendar</b> fecha = <b>new GregorianCalendar</b> ();
MONTH	1	int dia = fecha. <b>get</b> ( <b>GregorianCalendar.DATE</b> );
YEAR	2016	System.out.println(dia + "/" + (mes + 1) + "/" + año);

Clase Date - (import java.util.Date) - representa un instante específico en el tiempo	
<b>Date</b> fecha = <b>new Date</b> ();	//calendar es un objeto GregorianCalendar
<b>Date</b> fechaContrato = <b>calendar.getTime</b> ();	

Convertir un valor String a numérico	Convertir un valor numérico a String
int valorEntero = <b>Integer.parseInt</b> (cadena);	<b>String</b> numero = <b>String.valueOf</b> (1234);
double valorReal = <b>Double.parseDouble</b> (cadena);	<b>String</b> fecha = <b>String.valueOf</b> (new <b>Date</b> ());
	<b>String</b> cadena = <b>Integer.toString</b> (valorEntero);

Clase Random - (import java.util.Random;) - Genera números aleatorios	
Random rnd = new Random();	//Creamos un objeto Random
int numero = rnd.nextInt(1001);	//devuelve entre 0-1000
Int ranDate = rnd.seSeed(new Date().getTime());	//devuelve números aleatorios a partir de la hora

Clase Math			
abs	Valor absoluto	int x = <b>Math.abs</b> (-2);	x = 2;
sqrt	Raíz cuadrada	double x = <b>Math.sqrt</b> (9);	x = 3.0;
pow	Potencia	double x = <b>Math.pow</b> (2, 3);	x = 8.0;
PI	Número PI	double x= <b>Math.PI</b>	x = 3.14159265358979;
round	Redondea decimal a entero	int x = (int) <b>Math.round</b> (4.6);	x = 5;
max	Devuelve el mayor	int x = <b>Math.max</b> (5, 10);	x = 10;
min	Devuelve el menor	int x = <b>Math.min</b> (5, 10);	x = 5;
random	Nº real aleatorio [0-1]	double x = <b>Math.random</b> ();	x = 0.20614522323378;
	Nº entero aleatorio 1-10	int x = (int)( <b>Math.random</b> () * (Mx-Min+1) + Min);	//devuelve entre 1-6

Clase String - Texto entre comillas			
<b>equals</b>	Compara 2 cad.	boolean var = s1. <b>equals</b> (s2);	true o false;
<b>equalsIgCa</b>	Compara 2 cad.	boolean var = s1. <b>equalsIgnoreCase</b> (s2);	true o false;
<b>length</b>	Longitud de cadena	s1 = "Prueba"; int x = s1. <b>length</b> ();	x = 6;
<b>charAt</b>	Guarda char de pos pedida	s1 = "Prueba"; <b>char</b> x = s1. <b>charAt</b> (2) ;	x ='u';
<b>substring</b>	Genera subcadena	s1 = "Ey tu"; String x = s1. <b>substring</b> (0,2);	x ="Ey";
<b>toCharArra</b>	String a array	<b>char []</b> x = s1. <b>toCharArray</b> ();	x[0]= 'H'; x[1]= 'o';...
<b>indexOf</b>	1era pos donde aparece una cadena de texto	String s1 = "Quería decirte que"; int x = s1. <b>indexOf</b> ("t");	x=12; Devuelve -1 si no la encuentra
<b>lastIndexOf</b>	Última pos donde aparece una cadena de texto	String s1 = "Quería decirte que"; int x = s1. <b>lastIndexOf</b> ("ue");	x=16; Devuelve -1 si no la encuentra
<b>endsWith</b>	True si cadena termina con texto determinado texto	String s1 = "Quiero que te vayas"; boolean x = s1. <b>endsWith</b> ("vayas");	x = true;
<b>startsWith</b>	True si cadena empieza con texto determinado	String s1 = "Quiero que te vayas"; boolean x = s1. <b>startsWith</b> ("Qui");	x = true;
<b>replace</b>	Sustituye en cadena las apariciones de un carácter	String s1 = "Mariposa"; String x = s1. <b>replace</b> ('a', 'e');	x= "Meripose"
<b>replaceAll</b>	Sustituye en cadena las apariciones de una cadena	String s1="Armadillos"; String x = s1. <b>replaceAll</b> ("Ar", "Er");	x="Ermadillos";
<b>toUpperCa</b>	Cadena a mayús.	String s1="Hola"; String x = s1. <b>toUpperCase</b> ();	x= "HOLA"
<b>toLowerCa</b>	Cadena a minús.	String s1="Hola"; String x = s1. <b>toLowerCase</b> ();	x= "hola"
<b>contains</b>	True si encuentra cadena	Boolean x = s1. <b>contains</b> ("a");	s1 = "hola"; x = true;
<b>isEmpty</b>	True si es igual a 0	Boolean x = s1. <b>isEmpty</b> ();	x = false;
<b>Character.toUpperCase</b> (caracter);		Convierte el carácter indicado en un carácter en mayúsculas	

Clase StringBuilder - (StringBuilder sb = new StringBuilder()); - Texto entre comillas		
<b>lenght</b>	Devuelve longitud cadena	sb. <b>length</b> ();
<b>capacity</b>	Devuelve capacidad cadena	sb. <b>capacity</b> ();
<b>insert</b>	Añade caracteres principio	sb. <b>insert</b> (pos,String);
<b>append</b>	Añade caracteres al final	sb. <b>append</b> (String);
<b>append</b>	Invierte dirección de cadena	sb. <b>reverse</b> ();
<b>Delete</b>	Modifican la cadena	sb. <b>delete</b> (posInicial,posFinal);
<b>replace</b>		sb. <b>replace</b> (posInicial,posFinal,String);

Enumerados	<b>public enum</b> DiaSemana {v1, v2, v3, v4, v5, v6, v7} DiaSemana primerDia = DiaSemana.v1;	generar una lista de valores
------------	--	------------------------------

Array unidimensional (vectores) - Almacena tipos primitivos y objetos	
int [] miArray = <b>new</b> int [10];	//declarar el array
<b>int</b> [] miArray = {15, 25, ..., n};	//declarar e inicializar el array
miArray. <b>length</b> ;	//devuelve longitud del array
miArray = cadena. <b>getBytes</b> ();	//transforma cadena en array de bytes

Array multidimensional (matrices) – Almacena tipos primitivos y objetos	
int [] [] miArray = <b>new</b> int [5] [5];	//declarar el array
int [][] miArray = {{15, 25},{50, 60},{Vn, Vn}};	//declarar e inicializar el array

Estructuras de decisión: If...else, If...else if - Se usan para ejecutar 2 trozos de código mutuamente excluyentes.			
if (condicion) { sentencia 1n; } else { sentencia 2n; }	if (condicion) { sentencia 11; } else if { (condición1) sentencia 11; } else {...}	F. clásica if (condicion) variable = v1; else variable = v2;	F. compacta variable = condicion ? v1 : v2 ;

[ switch...case ] Se usa cuando se conoce la lista valores (int, boolean y char).	while (condición) { instrucciones si es true; }	While: Se ejecuta 0 o más veces.
switch (expresión) { case 1: case 2: instrucciones; [ break; ] default: instrucciones; }	do { instrucciones; } while (condición);  for (inicBucle; condición; contBucle) { instrucciones; }	Do While: Se ejecuta al menos una vez.  For: Se usa cuando conocemos el número de veces que se repite la ejecución de un código.
for (int dato : miArray) { System.out.println(dato); }		For - Each: se usa pare recorrer arrays de forma sencilla.

Crear clase	public class NomClase { atributos; métodos; }	//una sola clase  public en cada fichero  Modificadores: public, private , [de paquete] Pueden ser: [e objeto] o static
	private int valor1;	//declarar atributos: objeto, de clase o static
	private static int valor2;	//compartida por todos los objetos de la clase
	private final String valor3;	//final prohíbe cambiar el valor de la variable
	public NomClase (int valor) { this.valor1 = valor; }	//constructor //mismo nombre que la clase
	public int getNomMetodo() { return this.valor1; }	//método GET, si devuelve: se le llama función //return para devolver dato
	public void setNomMetodo(int valor) { this.valor1 = valor;	//método SET, //no devuelve valor: se le llama procedimiento
	public static tipo nomMetodo (int valor) { ...} ;	//método estático: no opera sobre objeto. //llamar a método estático: clase.método
	NomClase obj1 = new NomClase(); obj1.nomAtributo; obj1.nomMetodo(); NomClase.nomMetodo();	//instancia, llamada al constructor //accede a un atributo si es public //accede a un método del objeto (dinámico) //accede a un método de la clase (estático)

Operaciones con Arrays y Colecciones - (import java.util.Arrays)	
Arrays.sort(miArray);	//ordena Array de int, String o date
Arrays.fill(miArray,elemento);	//rellena todo el array con el elemento
Arrays.binarySearch(miArray,elemento);	//búsqueda rápida en Arrays ordenados

--