

Codigo	Visual Basic	Java	Python	Observaciones
Tipos:	Integer Decimal Char	int float char		
Solicitud de Datos a Usuarios	intinput floatinput charinput	intinput floatinput charinput	intinput floatinput charinput	
Expresiones Aritmeticas	'+' Suma - Resta o sustracción * Multiplicación / División \ División entera o parte entera del cociente de una división Mod Residuo de la división entera ^ Potencia o exponenciación	'+' Suma - Resta o sustracción * Multiplicación / División % Residuo de la división entera	'+' Suma - Resta o sustracción * Multiplicación / División //Cociente % Residuo de la división entera **Potencia	
Mensajes a Pantalla	Console.WriteLine(<Mensaje que va a tener>) Console.Write(<Mensaje que va a tener>)	System.out.println("..."); System.out.println("...");	print(<id> string [, <id> string])	
Ciclos	For VAR = Vi To Vf ... Next VAR while [condición] ... end while Do While [condición] Loop do ... Loop While[condición]	for(<tipo> <var> = <inicial>; <var> <Comparacion> valor final; <Incremento>) { ... } while([condición]){...} do{...}while([condición]);	for [id] in range([int] [, int]?): TAB ... [else:]? for [id] in [arreglo]: TAB ... [else:]? while [condicion]: Tab...	
Break	Continue <loop> Exit <loop>	break;	break	
Manejo de Condiciones	If [condicion] [Then]?... ElseIf [condicion] [Then]?... Else.. End If Select Case [id] Case [valor] Case [valor] To [valor] Case [valor], [valor], ... Case Else End Select	if([condicion]){...} {(else{...})?} {else if([condicion]){...})?}* switch([expression]) { case [val]: ... [break]? default:... [break]? }	if [expression] : ... {(elif [expression] : ...)* [else] : ...}	The obligatorio si los statment son de una linea en VB
Declaración de Variables	Dim <id> As <TipoDato> = <valor> Dim <id>= <valor> Dim <id> [<Listado de Id separado por .>] As <TipoDato>	public <Tipo> <id> = <valor> <id>; public <Tipo> <id>; public <Tipo> [<Listado de Id separado por .>] = <valor> <id> ?>;	[id] = <valor> [id]	
Asignacion de Variables	<id> = <valor> <id> <id> incrementos <valor> <id>	<id> = <valor> <id>; <id> incrementos <valor> <id>;	Declaracion de Variables <id> incrementos <valor> <id>	los incrementos pueden ser : += /= -=
Funciones	Function <nombre> [(parameterlist)] [As returntype] ... [Exit Function] [Return Expression] End Function	public [tipo] <id>{([<tipo><id> [,<tipo><id>]*]?) { [return [valor];]* }	def <id>{([<id> , <id>]*]?)?: tab return [expresion]	
Procedimientos	Sub <id> [(parameterlist)] ... [Exit Sub] [Return] End Sub	public [tipo] <id>{([<tipo><id> [,<tipo><id>]*]?) { [return [valor];]* }	def <id>{([<id> , <id>]*]?)?: tab ...	para vb los parametros deben ir =<id> as <tipo> separados por coma
Clases Modules		public class <id>{ public <id>{parametros}{ this.<id>=<id>; } }		