|  |  |
| --- | --- |
| Globston | |
| program{  ……  } | cuerpo del programa |
| Mover(Norte)  Mover(Este)  Mover(Sur)  Mover(Oeste) | mueve el cabezal en la dirección respectivamente.  Nota: Cuidado con el borde del tablero |
| Poner(Rojo)  Poner(Azul)  Poner(Negro)  Poner(Verde) | pone una bolita del color del paramentro en el lugar que se encuentra el cabezal. |
| Sacar(Rojo)  Sacar(Azul)  Sacar(Negro)  Sacar(Verde) | Saca una bolita del color del paramentro en el lugar que se encuentra el cabezal.  Nota: no se puede sacar bolitas que no existen. |
| nroBolitas(Azul)  …. | devuelve la cantidad de bolitas del color indicado hay en el lugar que se encuentra el cabezal |
| hayBolitas(Rojo)  ….. | true o false |
| IrAlBorde(Norte)  …. | Si no concemos la ubicación del cabezal entonces usarmos para hubicar en el borde. |
| puedeMover(Norte)  …. | True o false si se pude mover a la dirección indcada. |
| procedure NombreDelProcedure(){  ….  }  program{  NombreDelProcedure()  } | se declara un proceso. (una función viod)(puede recibir parametros)  Cuerpo del proceso  invocamos el procedure |
| repeat(10){  …  } | función repetir (10 veces)  cuerpo el cliclo |
| if (condicion){  ….  } | >,<,=, not, ||, && |
| function nombreDeLaFuncion(){  …..  return x  } | funcion con return  cuerpo de la función  Nota: las fuciones no realizan cambios. |
| JAVA SCRIPT | |
| function nombreDeLaFuncion(var){  …;  return x;  } |  |
| Math.round(4.5);  Math.max(4,5,n);  Math.min(4,5,n); | redondea  devuelve el máximo  devuelve el mínimo |
| operadores de comparacion | ¡==  >=  <=  >  <  ||  &&  “!” |
| concatenacion | “A”+”b” |
| longitud(String) | devuelve la cantidad de carecateres del str |
| imprimir(“un texto” + variable) | imprimir por pantalla |
| let x=2; |  |
| listas |  |
| let lista1=[]  longitud(lista1);  agregar(lista1,”otra”);  remover(lista1,”aleatoria”);  posición(lista1, elmento);  máximo(lista);  minimo(lista1); | devuelve la cantidad de elementos de la lista  agrega al final de la lista  borra el elemento de la lista  devuelve la ubicación del elemento en la lista (index 0-5);  nota: si el elemento no esta en la lista (-1)  devuelve el máximo de una lista  devuelve el minimo de una lista |
| for (let aux of lista1){  …  } | toma una por una los elementos de la lista, comenzando desde el cero. |
| **regitros o diccionarios** |  |
| let monumentoNacionalALaBandera={ nombre:"Monumento Nacional A La Bandera", locacion:"Rosario, Argentina", anioDeConstruccion: 1957}; | acceso  diccionario1.nombre  diccionario1.edad  …. |
| anio("14/09/1989"); | devuelve el año |
| Ruby (Todo es un objeto) | |
| Pepita.enegia  Pepita.cantar!  Pepita.cantar!(2,”las mañanitas”)  Pepita.nueva=(ave)  Pepita.saba\_cantar? | llamada a obejetos y sus metodos |
| característica de Ruby | 2.+3 es lo “mismo” que 2+3  -Se puden guardar objetos dentro de una variable. (@var=obera)y usar esos objetos mediante la varible |
| module Pepita  @energia = 100  def self.volar\_en\_circulos!  @energia = @energia - 10  end  end | creación de un modulo  atributos  método y tipo de método (¡)  acción  fin método  fin modulo |
| (@variable).abs  (variable).upcase  variable.size  varible.equal? variable2  variable1 == variable2 | nos devuelve el valor absoluto  mayuscula  len  true o false (===) si ambas variables se refieren exactamente al mismo objeto (parecidos es falso)  a diferencia de equal este admite parecidos |
| 3.timer{} | ciclo |
| if self.debil?  self.comer\_alpiste!(10)  elsif self.feliz?  5.times {self.volar\_en\_circulos!}  else  3.times {self.volar\_en\_circulos!}  end | condicionales |
| @juegos=[CarlosDuty,TimbaElLeon, Metroide] | lista |
| Lista.push 1  Lista.delete 7  Lista.include? 1  Lista.size  lita.max  no van en los sets:  Lista.first  Lista.last  Lista.index 2 | pone el elemento al final de la lista  borra el elementos de la lista  true o false si incluyo o no la lista el element  len  (en caso de ser lista de números) retorna el maximo valor de la lista  devuelve el primer elemento  devuelve el segundo elemento  devuelve la posición del elemento dentro dela lista |
| Lista.to\_set | convierte la lista en un conjunto, set o “diccionario”. A diferencia es que no admite valores repetidos |
| funciones lambda |  |
| var=proc{|argumento1, argumento2| vartotal=arg1+arg2}  var.call(arg1,arg2)  var.call | permite guardar una función simple en una variable  pude no llevar argumentos |
| @lista.select{|nameAux| nameAux>3} | lambda para listas  recorre la lista y evalua en su condición si es true o flase  crea una nueva lista(no modifica la lista original) |
| @lista.find{|aux| aux>3} | devuelve solo el primer elemento que cumpla la condición  nueva variable |
| estudiantes.all? { |un\_estudiante| un\_estudiante.aprobo? }  estudiantes.any? { |un\_estudiante| un\_estudiante.aprobo? } | devuelven true o false según si criterio  all=todos  any=alguno |
| lista\_aves.map { |una\_golondrina| una\_golondrina.energia } | A diferencia de otro este llama a métodos,  crea una nueva lista con los retornos de los metodos |
| estudiantes.count { |un\_estudiante| un\_estudiante.aprobo? }  estudiantes.sum { **|**un\_estudiante**|** un\_estudiante.nota\_en\_examen | devuelve la cantidad que cumplen true en la condición  devuelve la suma de los retornos de la condicion |
| lista.each{|aux|aux.motodo!(6)} | ESTA NOS PERMITE HACER CORRER LOS METODOS (SIMILAR A MAP pero nos permite modificar)  no recibe retornos ni crea listas |
| Clases (igual se los modules pero sin self y las atributos van dentro de def initialize)  Constructores:  Initialize puede recibir parámetros para dar un valor inicial al crear una instancia  Def initialize (x,y,z)  @letra1=x  @letra2=y  End  -…. | nuevoObjeto=Class.new a,b,c |
| herenzcia  class clase<Superclase | para |
| **class** Saludo  **def** saludar  "Buen día"  **end**  **end**  **class** SaludoDocente **<** Saludo  **def** saludar  **super** **+** " estudiantes"  **end**  **end** | super |
| nota: para evitar heredar automáticamente un método  creamos un método en el hijo con el mismo nombre  Al llamar al método, pisa al del padre |  |