

Machete oficial - Smalltalk

1. Review

Valores	
Comentarios	"un comentario"
Strings	'uNa CadEna'
Caracteres	\$a
Simbolos	#unSimbolo
Booleanos	true false
Array literal	#(1 2 3) #(1 \$a 'hola')
Operadores lógicos	
Igualdad	= (igual) == (idéntico)
Diferencia	~= (no igual) ~~ (no idéntico)
Comparación	> >= < <=
"O" lógico	c1 c2 c1 or: [c2]
"Y" lógico	c1 & c2 c1 and: [c2]
Negación	<i>unBool not</i>
Operadores matemáticos	
Operaciones básicas	+ - * /
División entera	<i>dividendo // divisor</i>
Resto	<i>dividendo \\ divisor</i>
Valor absoluto	<i>unNro abs</i>
Exponenciación	<i>base raisedTo: exponente</i>
Raíz cuadrada	<i>unNro sqrt</i>
Valor ascii de un caracter	<i>unChar asciiValue</i>
Caracter de un valor ascii	<i>unNro asCharacter</i>
Colecciones y listas	
Longitud	<i>unaCol size</i>
Concatenación (no modifica las colecciones)	<i>unaCol , unaCol</i>
Agregar un elemento	<i>unaCol add: unObjeto</i>
Posición numérica	<i>unaCol at: unNro</i>
Test de pertenencia	<i>unaCol includes: unObj</i>

2. Smalltalk

A pesar de la variedad de mensajes la sintaxis del lenguaje siempre es **objeto mensaje**

Expresiones y mensajes de uso común

<i>unaVariable</i>	Declara a <i>unaVariable</i> como variable local.
.	Indica separación entre sentencias.
self	Referencia al objeto receptor del mensaje que ejecutó el método.
nil	Objeto nulo.
<i>unaVariable</i> := <i>unaExpresión</i>	Asigna <i>unaExpresión</i> a <i>unaVariable</i>
^ <i>unObjeto</i>	Retorna <i>unObjeto</i> y termina la ejecución del método
[<i>sentencias</i>]	Constituye un bloque de código que contiene a las <i>sentencias</i>
<i>UnaClase</i> new	Crea y devuelve una nueva instancia de <i>unaClase</i>
<i>unObjeto</i> isNil	Devuelve true si el receptor es nil
<i>unObjeto</i> notNil	Devuelve true si el receptor no es nil
<i>unObjeto</i> printString	Devuelve un string que representa el contenido de <i>unObjeto</i>
<i>unBool</i> ifTrue: [<i>unasSentencias</i>] ifFalse: [<i>otrasSentencias</i>]	Ejecuta <i>unasSentencias</i> u <i>otrasSentencias</i> dependiendo del valor de verdad de <i>unBool</i> .
[<i>unBool</i>] whileTrue: [<i>sentencias</i>]	Ejecuta iterativamente las <i>sentencias</i> mientras <i>unBool</i> sea verdadero.
<i>unNro</i> timesRepeat: [<i>sentencias</i>]	Ejecuta iterativamente las <i>sentencias</i> exactamente <i>unNro</i> de veces.
<i>unNro</i> to: <i>otroNro</i> do: [: <i>indice</i> <i>sentencias</i>]	Ejecuta iterativamente las <i>sentencias</i> la cantidad de veces comprendida entre <i>unNro</i> y <i>otroNro</i> . <i>Indice</i> varía en cada iteración, desde <i>unNro</i> hasta <i>otroNro</i> .

Clases de colecciones más comunes

- **Bag**: Tamaño variable, sin subíndice.
- **Set**: Tamaño variable, sin subíndice, no permite repetidos.
- **Array**: Tamaño fijo, con subíndice, orden de acuerdo al subíndice.
En este sentido un String se comporta como un Array
- **OrderedCollection**: Tamaño variable, con subíndice, orden de acuerdo al subíndice.
- **SortedCollection**: Tamaño variable, con subíndice, orden de acuerdo a criterio que se especifica.
- **Dictionary**: Tamaño variable, acceso por clave, no permite claves repetidas

Los subíndices empiezan en 1.

Las colecciones con subíndice respetan el orden de los elementos en do: / select: / collect: / etc..

Para Dictionary do: / select: / collect: / etc. **funcionan sobre los valores** incluidos, no se tienen en cuenta las claves.

Métodos de colecciones

Para todas las colecciones	
<i>unaCol</i> size	Devuelve la cantidad de elementos que tiene <i>unaCol</i>
<i>unaCol</i> includes: <i>unObjeto</i>	Devuelve true si <i>unObjeto</i> se encuentra en <i>unaCol</i> .
<i>unaCol</i> occurrencesOf: <i>unObjeto</i>	Devuelve la cantidad de ocurrencias de <i>unObjeto</i> en <i>unaCol</i> .
<i>unaCol</i> asBag <i>unaCol</i> asSet <i>unaCol</i> asOrderedCollection <i>unaCol</i> asArray	Devuelve una nueva colección de la clase indicada con todos los elementos de <i>unaCol</i> .
<i>unaCol</i> asSortedCollection: [: <i>anterior</i> : <i>siguiente</i> <i>unaCondicion</i>]	Devuelve una nueva colección con todos los elementos de <i>unaCol</i> ordenados según <i>unaCondicion</i> . <i>unaCondicion</i> es una expresión de valor booleano en la que intervienen <i>anterior</i> y <i>siguiente</i> . <i>anterior</i> quedará delante de <i>siguiente</i> cuando <i>unaCondicion</i> sea verdadera.

<i>unaCol do</i> : [: <i>unElem</i> <i>sentencias</i>]	Ejecuta iterativamente las sentencias para cada <i>unElem</i> de <i>unaCol</i> . Retorna <i>unaCol</i> , en otras palabras no devuelve nada interesante
<i>unaCol select</i> : [: <i>unElem</i> <i>unaExpr</i>]	Devuelve una nueva colección con los elementos de <i>unaCol</i> que hacen verdadero a <i>unaExpr</i> .*
<i>unaCol reject</i> : [: <i>unElem</i> <i>unaExpr</i>]	Devuelve una nueva colección con todos los elementos de <i>unaCol</i> excepto los que hacen verdadero a <i>unaExpr</i> .*
<i>unaCol detect</i> : [: <i>unElem</i> <i>unaExpr</i>] <i>ifNone</i> : <i>unBloque</i>	Devuelve el primer elemento de <i>unaCol</i> que hace verdadero a <i>unaExpr</i> . Si ninguno lo hiciera, se retornara la ejecución de <i>unBloque</i> .*
<i>unaCol collect</i> : [: <i>unElem</i> <i>sentencias</i>]	Devuelve una nueva colección con el resultado de evaluar iterativamente las <i>sentencias</i> para cada <i>unElem</i> de <i>unaCol</i> .*
<i>unaCol inject</i> : <i>valorInicial</i> <i>into</i> : [: <i>acumulador</i> : <i>unElem</i> <i>unaOperación</i>]	El <i>acumulador</i> empieza siendo el <i>valorInicial</i> . Luego se evalúa <i>unaOperación</i> para cada elemento en <i>unaCol</i> , y el resultado es puesto en el <i>acumulador</i> . Retorna el valor final del <i>acumulador</i> P.ej. para obtener la suma de los elementos de una colección <i>unaCol inject: 0 into: [:resul :elem resul + elem]</i>
<i>unaCol allSatisfy</i> : [: <i>unElem</i> <i>unaExpr</i>]	Devuelve true si todos los elementos de la colección hacen verdadera <i>unaExpr</i> .*
<i>unaCol anySatisfy</i> : [: <i>unElem</i> <i>unaExpr</i>]	Devuelve true si algún elemento de la colección hace verdadera <i>unaExpr</i> .*
Sólo para colecciones de tamaño variable	
<i>unaCol add</i> : <i>unObjeto</i>	Agrega <i>unObjeto</i> a <i>unaCol</i> . Devuelve unObjeto . Para las colecciones con subíndice se agrega al final.
<i>unaCol addAll</i> : <i>otraCol</i>	Agrega todos los elementos de <i>otraCol</i> a <i>unaCol</i> . Para las colecciones con subíndice se agregan al final. Devuelve otraCol
<i>unaCol remove</i> : <i>unObjeto</i>	Elimina <i>unObjeto</i> de <i>unaCol</i> . Devuelve unObjeto .
<i>unaCol removeAll</i>	Elimina todos los elementos de <i>unaCol</i> . Devuelve unaCol .
Sólo para colecciones con subíndice	OrderedCollection, SortedCollection, Array, String
<i>unaCol</i> , <i>otraCol</i>	Devuelve una nueva colección con la concatenación de <i>unaCol</i> y <i>otraCol</i> . Respeto el orden.
<i>unaCol at</i> : <i>unNro</i>	Devuelve <i>el elemento</i> en la posición <i>unNro</i> .
<i>unaCol at</i> : <i>unNro</i> put : <i>unObjeto</i>	Coloca <i>unObjeto</i> en la posición <i>unNro</i> de <i>unaCol</i> . Inválido para SortedCollection.
<i>unaCol first</i>	Devuelve el primer elemento de <i>unaCol</i> También hay second y otros.
<i>unaCol last</i>	Devuelve el último elemento de <i>unaCol</i>
<i>unaCol indexOf</i> : <i>unObjeto</i>	Devuelve la posición en la que aparece <i>unObjeto</i> dentro de <i>unaCol</i> (la primera si estuviera repetido); 0 si <i>unObjeto</i> no está en <i>unaCol</i> .
<i>unaCol beginsWith</i> : <i>otraCol</i>	<i>true</i> si ... <i>unaCol</i> empieza con <i>otraCol</i> . También hay endsWith :
<i>unaCol copyFrom</i> : <i>unNro</i> to : <i>otroNro</i>	Devuelve una nueva colección con los elementos de <i>unaCol</i> comprendidos entre las posiciones <i>unNro</i> y <i>otroNro</i> .
<i>unaCol allButLast</i>	Devuelve una nueva colección con todos los elementos de <i>unaCol</i> excepto el último. También hay allButFirst .
<i>UnaClase new</i> : <i>unNro</i>	Devuelve una nueva colección de <i>UnaClase</i> de tamaño <i>unNro</i>
Sólo para Dictionary	
<i>unaCol at</i> : <i>unaClave</i>	Devuelve el valor asociado a <i>unaClave</i> , <i>nil</i> si <i>unaClave</i> no tiene asociado ningún valor.

* *unElem* referencia iterativamente a cada uno de los elementos de *unaCol*. *unaExpr* es una expresión de valor booleano en la que interviene *unElem*.

<i>unaCol</i> at: <i>unaClave</i> put: <i>unObjeto</i>	Coloca <i>unObjeto</i> como valor asociado a <i>unaClave</i> .
----------------------------------------------------------------------	----------------------------------------------------------------