

Machete de sintaxis Wollok¹

Elementos Comunes

Sintaxis básica

Objeto con un atributo y dos métodos	<pre>object pepita { var energia = 0 method volar(kilometros) { energia += kilometros } method puedeVolar() { return energia > 0 } }</pre>
Comentario	<pre>// un comentario /* un comentario multilínea */</pre>
Strings	<pre>"uNa CadEna" 'uNa CadEna'</pre>
Booleanos	<pre>true false</pre>
Conjunto	<pre>#{} #{1, 2}</pre>
Lista	<pre>[] [1, 1, 2]</pre>
Bloques sin parámetros	<pre>{algo}</pre>
Bloques / Exp. lambda (De un parámetro)	<pre>{x => algo con x}</pre>
Bloques / Exp. lambda (Más de un parámetro)	<pre>{x, y => algo con x e y}</pre>

Operadores lógicos y matemáticos

Equivalencia	<pre>==</pre>
Identidad	<pre>===</pre>
~ Equivalencia	<pre>!=</pre>
Comparación de orden	<pre>> >= < <=</pre>
Disyunción (O lógico)	<pre> or</pre>

¹ Este apunte está basado en la *Guía de lenguajes 3.1.1*, elaborada por docentes de la cátedra Paradigmas de Programación de la Universidad Tecnológica Nacional, Facultad Regional Buenos Aires.

Conjunción (Y lógico)	<code>&&</code> <code>and</code>
Negación	<code>! unBool</code> <code>unBool.negate()</code> <code>not unBool</code>
Operadores aritméticos	<code>+</code> <code>-</code> <code>*</code> <code>/</code>
División entera	<code>dividendo.div(divisor)</code>
Resto	<code>dividendo % divisor</code>
Valor absoluto	<code>unNro.abs()</code>
Exponenciación	<code>base ** exponente</code>
Raíz cuadrada	<code>unNro.squareRoot()</code>
Máximo entre dos números	<code>unNro.max(otroNro)</code>
Mínimo entre dos números	<code>unNro.min(otroNro)</code>
Verificar si un número está entre otros dos	<code>unNro.between(un, otro)</code>
Par	<code>unNro.even()</code>
Impar	<code>unNro.odd()</code>

Operaciones simples sin efecto sobre colecciones / listas

Longitud	<code>coleccion.size()</code>
Si está vacía	<code>coleccion.isEmpty()</code>
Concatenación	<code>coleccion + otraColeccion</code>
Unión	<code>set.union(coleccion)</code>
Intersección	<code>set.intersection(coleccion)</code>
Acceso por índice	<code>lista.get(indice)</code> (<i>base 0</i>)
Pertenencia	<code>coleccion.contains(elem)</code>
Máximo	<code>coleccionOrdenable.max()</code>
Mínimo	<code>coleccionOrdenable.min()</code>
Sumatoria	<code>coleccionNumerica.sum()</code>
Aplanar	<code>coleccionDeColecciones.flatten()</code>

Primeros n elementos	<code>lista.take(n)</code>
Primer elemento	<code>lista.head()</code> <code>lista.first()</code>
Último elemento	<code>lista.last()</code>
Sin repetidos	<code>coleccion.asSet()</code>

Operaciones avanzadas (de orden superior) sin efecto sobre colecciones/listas

Sumatoria según transformación	<code>coleccion.sum(bloqueNumericoDe1)</code>
Filtrar	<code>coleccion.filter(bloqueBoolDe1)</code>
Transformar	<code>coleccion.map(bloqueDe1)</code>
Todos cumplen (true para lista vacía)	<code>coleccion.all(bloqueBoolDe1)</code>
Alguno cumple (false para lista vacía)	<code>coleccion.any(bloqueBoolDe1)</code>
Transformar y aplanar	<code>coleccion.flatMap(bloqueDe1)</code>
Reducir/plegar a izquierda	<code>coleccion.fold(valorInicial, bloqueDe2)</code>
Reducir/plegar a derecha	NA
Apareo con transformación	NA
Primer elemento que cumple condición	<code>coleccion.find(bloqueBoolDe1)</code> <code>coleccion.findOrElse(bloqueBoolDe1, bloqueSinParametros)</code>
Cantidad de elementos que cumplen condición	<code>coleccion.count(bloqueBoolDe1)</code>
Obtener colección ordenada.	<code>coleccion.sortedBy(bloqueBoolDe2)</code>
Máximo según criterio.	<code>coleccion.max(bloqueOrdenableDe1)</code>
Mínimo según criterio.	<code>coleccion.min(bloqueOrdenableDe1)</code>

Mensajes de colecciones con efecto

Agregar un elemento.	<code>coleccion.add(objeto)</code>
Agregar todos los elementos de la otra colección	<code>coleccion.addAll(otraColeccion)</code>
Evaluar el bloque para cada elemento.	<code>coleccion.forEach(bloqueConEfectoDe1)</code>

Eliminar un objeto.	<code>coleccion.remove(objeto)</code>
Eliminar todos los elementos.	<code>coleccion.clear()</code>
Deja ordenada la lista según un criterio.	<code>lista.sortBy(bloqueBoolDe2)</code>