

:	<code>a -&gt; [a] -&gt; [a]</code>	Add a single element to the front of a list. <code>3: [2,3] ~ [3,2,3]</code>
++	<code>[a] -&gt; [a] -&gt; [a]</code>	Join two lists together. <code>"Ron"++"aldo" ~ "Ronaldo"</code>
!!	<code>[a] -&gt; Int -&gt; a</code>	<code>xs!!n</code> returns the <i>n</i> th element of <i>xs</i> , starting at the beginning and counting from 0. <code>[14,7,3]!!1 ~ 7</code>
concat	<code>[[a]] -&gt; [a]</code>	Concatenate a list of lists into a single list. <code>concat [[2,3],[],[4]] ~ [2,3,4]</code>
length	<code>[a] -&gt; Int</code>	The length of the list. <code>length "word" ~ 4</code>
head,last	<code>[a] -&gt; a</code>	The first/last element of the list. <code>head "word" ~ 'w'</code> <code>last "word" ~ 'd'</code>
tail,init	<code>[a] -&gt; [a]</code>	All but the first/last element of the list. <code>tail "word" ~ "ord"</code> <code>init "word" ~ "wor"</code>
replicate	<code>Int -&gt; a -&gt; [a]</code>	Make a list of <i>n</i> copies of the item. <code>replicate 3 'c' ~ "ccc"</code>
take	<code>Int -&gt; [a] -&gt; [a]</code>	Take <i>n</i> elements from the front of a list. <code>take 3 "Peccary" ~ "Pec"</code>
drop	<code>Int -&gt; [a] -&gt; [a]</code>	Drop <i>n</i> elements from the front of a list. <code>drop 3 "Peccary" ~ "cary"</code>
splitAt	<code>Int-&gt;[a]-&gt;([a],[a])</code>	Split a list at a given position. <code>splitAt 3 "Peccary" ~ ("Pec","cary")</code>
reverse	<code>[a] -&gt; [a]</code>	Reverse the order of the elements. <code>reverse [2,1,3] ~ [3,1,2]</code>
zip	<code>[a]-&gt;[b]-&gt;[(a,b)]</code>	Take a pair of lists into a list of pairs. <code>zip [1,2] [3,4,5] ~ [(1,3),(2,4)]</code>
unzip	<code>[(a,b)] -&gt; ([a],[b])</code>	Take a list of pairs into a pair of lists. <code>unzip [(1,5),(3,6)] ~ ([1,3],[5,6])</code>

Figure 6.1 Some polymorphic list operations from `Prelude.hs`