

# Ruby monstas Ruby cheat sheet



## Data types and how to use them

Name	Description	Structure	Examples
Integer literal	A whole number		3 -552
Floating point literal	A decimal number		42.23 -0.133742
Addition		$a + b$	5.2 + 6.34 # => 11.54
Subtraction		$a - b$	2.59 - 4.89 # => -2.3
Multiplication		$a * b$	5 * 3.7 # => 18.5
Division	Mind the difference between integer and float divisions!	$a / b$	6 / 4 # => 1 (integer division) 6 / 4.0 # => 1.5 (float division)
Modulo	Returns the remainder of a division	$a \% b$	13 % 6 # => 1
String literal	A string of characters, text		"this is a string"
String interpolation	Text with Ruby code embedded in it		"another string with an #{interpolation}"

## Arrays

Name	Description	Structure	Examples
Array literal	Creates a new array	<i>[item1, item2, ...]</i>	<code>my_array = [1, 2, 3]</code>
Length	Returns the length of the array (the number of items it contains)	<i>array.length</i>	<code>my_array.length</code> # => 3
Index operator	Lets you access the item at a given position within an array	<i>array[index]</i>	<code>my_array[1]</code> # => 2
delete_at	Deletes the item at a given index and returns it	<i>array.delete_at(index)</i>	<code>my_array.delete_at(1)</code> # => 2
each	Lets you iterate over all elements in an array	<i>array.each do  item </i> <i>end</i>	<code>my_array.each do  item </code> <code>puts item</code> <code>end</code>
first	Returns the very first item of the array	<i>array.first</i>	<code>my_array.first</code> # => 1
last	Returns the very last item of the array	<i>array.last</i>	<code>my_array.last</code> # => 3
include?	Returns a boolean, whether the array contains a certain element or not	<i>array.include?(item)</i>	<code>my_array.include?(4)</code> # => false
pop	Removes the last item of the array and returns it	<i>array.pop</i>	<code>my_array.pop</code> # => 3
push or <<	Adds an item to the end of the array	<i>array.push(item)</i> <i>array &lt;&lt; item</i>	<code>my_array.push(4)</code> <code>my_array &lt;&lt; 4</code>
reverse	Returns a copy of the array with the elements in reverse order	<i>array.reverse</i>	<code>my_array.reverse</code> # [3, 2, 1]
sort	Returns a sorted copy of the array	<i>array.sort</i>	<code>[5, 2, 4].sort</code> # [2, 4, 5]
uniq	Returns a copy of the array with duplicates removed	<i>array.uniq</i>	<code>[1, 1, 2, 2].uniq</code> # [1, 2]

## Hashes

Name	Description	Structure	Examples
Hash literal	Create a hash	<code>{"key" =&gt; "value" }</code>	<code>hash = {}</code> <code>hash = { "key" =&gt; "value", "other_key" =&gt; 42 }</code>
Hash access	Access a value by its key	<code>hash[key]</code>	<code>hash["key"]</code> # => "value"
Key deletion	Delete a key-value pair by its key	<code>hash.delete(key)</code>	<code>hash.delete("key")</code> # => "value"
Empty hash	Remove all pairs from the hash	<code>hash.clear</code>	<code>hash.clear</code>
Iterate over hash	Iterate over all the pairs in the hash	<code>hash.each do  key, value </code> <code>end</code>	<code>hash.each do  key, value </code> <code>puts "#{key} has value: #{value}"</code> <code>end</code>
Iterate over pairs	Iterate over all the pairs in the hash	<code>hash.each_pair do  key, value </code> <code>end</code>	<code>hash.each_pair do  key, value </code> <code>puts "#{key} has value: #{value}"</code> <code>end</code>
Get key	Get a value for a key, with default value if the key does not exist.	<code>hash.fetch(key, default)</code>	<code>hash.fetch("key")</code> # => "value" <code>hash.fetch("xy", "default")</code> <code>=&gt; "default"</code>
Key existence	Ask the hash if it has a certain key	<code>hash.has_key?(key)</code>	<code>hash.has_key?("key")</code> # => true
Value existence	Ask the hash if it has a certain value	<code>hash.has_value?(value)</code>	<code>hash.has_value?("xy")</code> # => false
All keys	Get all the keys stored in the hash	<code>hash.keys</code>	<code>hash.keys</code> # => ["key", "other_key"]
All values	Get all the values stored in the hash	<code>hash.values</code>	<code>hash.values</code> # => ["value", 42]
Merge	Merge two hashes	<code>hash.merge(other_hash)</code>	<code>hash.merge({"a_key" =&gt; 23})</code> <code># =&gt; { "key" =&gt; "value", "a_key" =&gt; 23 }</code>