Ruby master - Misc #15514

Add documentation for implicit array decomposition

01/07/2019 10:26 AM - sos4nt (Stefan Schüßler)

Status:	Open	
Priority:	Normal	
Assignee:		

Description

The documentation for Array Decomposition says: "[...] you can decompose an Array during assignment using parenthesis [sic]" and gives an example:

```
(a, b) = [1, 2]

p a: a, b: b # prints {:a=>1, :b=>2}
```

But – as we all know – it's also possible without parentheses, i.e.

```
a, b = [1, 2]

p a: a , b: b #=> {:a=>1, :b=>2}
```

This also applies to block arguments when yielding multiple values vs. yielding a single array:

```
def foo
  yield 1, 2
end

def bar
  yield [1, 2]
end

foo { |a, b| p a: a, b: b }
#=> {:a=>1, :b=>2}

bar { |a, b| p a: a, b: b }
#=> {:a=>1, :b=>2}
```

In both cases, parentheses are optional.

This implicit array decomposition could be quite surprising for newcomers. The documentation should cover it.

History

#1 - 01/07/2019 11:10 AM - shevegen (Robert A. Heiler)

Agreed.

#2 - 01/10/2019 04:43 PM - lugray (Lisa Ugray)

If that's covered (and I agree it should be) it's also worth showing a case where they are not optional:

```
def baz
  yield [1, 2], 3
end

baz { |a, b, c| p a: a, b: b, c: c }
#=> {:a=>[1, 2], :b=>3, :c=>nil}

baz { |(a, b), c| p a: a, b: b, c: c }
#=> {:a=>1, :b=>2, :c=>3}
```