Ruby - Feature #20768

Add Hash#delete_at

09/28/2024 09:08 PM - seanpdoyle (Sean Doyle)

Status:	Open	
Priority:	Normal	
Assignee:		
Target version:		

Description

Add Hash#delete_at to delete values from a list keyed by its arguments.

The interface and return values draw inspiration from Hash#values_at and Array#delete_at. It can pair nicely with multiple assignment to deconstruct a Hash instance:

```
hash = { a: true, b: false }
a, c = hash.delete_at(:a, :c) # => [ true, nil ]
hash # => { b: false }
```

Similar outcomes can be achieved with rightward assignment:

```
hash = { a: true, b: false }
hash => { a:, **rest }
a # => true
rest # => { b: false }
```

However, rightward assignment will raise an error when a key is missing, whereas Hash#delete_at could return nil or the Hash#default_value:

```
hash => { c: } # => raises NoMatchingPatternKeyError
```

Similarly, rightward assignment does not mutate the original Hash instance. In some circumstances, this is beneficial. However, others that require mutation would involve re-assigning the variable:

```
hash => { a:, **hash }
a # => true
hash # => { b:false }
```

Finally, a block can be passed in the same style as Hash#delete to provide a default value:

```
attributes = { disabled: true, aria: { label: "label text" } }
aria, data = attributes.delete_at(:aria, :data) { Hash.new }
attributes # => { disabled: true }
aria # => { label: "label text" }
data # => {}
```

History

#1 - 10/01/2024 06:27 PM - Eregon (Benoit Daloze)

```
a, c = [:a, :c].map { hash.delete(it) }
```

seems enough and does not require adding a new method for what looks like (to me) a narrow use case.