

Class MatchData < Object

All pattern matches set the special variable `$~` to a MatchData containing information about the match. The methods `Regexp#match` and `Regexp.last_match` also return a MatchData object. The object encapsulates all the results of a pattern match, results normally accessed through the special variables `$&`, `$'`, `$``, `$1`, `$2`, and so on (see the list on page 340).

Instance methods

[] *match[i] → string*
match[name] → string
match[start, length] → array
match[range] → array

1.9 / Match Reference—MatchData acts as an array and/or hash and may be accessed using the normal indexing techniques. Numeric indices return the captures at the corresponding position in the regular expression (starting at 1). Symbol indices return the corresponding named capture. `match[0]` is equivalent to the special variable `$&` and returns the entire matched string. See also `MatchData#select` and `MatchData#values_at`.

```
m = /(.)(.)\d+\d/.match("THX1138.")
m[0]           # => "HX1138"
m[1, 2]        # => ["H", "X"]
m[1..3]        # => ["H", "X", "113"]
m[-3, 2]       # => ["X", "113"]
m = /..(?<digit_prefix>\d+)\d/.match("THX1138.")
m[:digit_prefix] # => "113"
```

begin *match.begin(n) → int*
match.begin(name) → int

1.9 / Returns the offset in the original string of the start of the *n*th capture or the named capture.

```
m = /(.)(.)\d+\d/.match("THX1138.")
m.begin(0)      # => 1
m.begin(2)      # => 2
m = /..(?<digit_prefix>\d+)\d/.match("THX1138.")
m.begin(:digit_prefix) # => 3
```

captures *match.captures → array*

Returns the array of all the matching groups. Compare to `MatchData#to_a`, which returns both the complete matched string and all the matching groups.

```
m = /(.)(.)\d+\d/.match("THX1138.")
m.captures # => ["H", "X", "113", "8"]
```

`captures` is useful when extracting parts of a match in an assignment.

```
f1, f2, f3 = /(.)(.)\d+\d/.match("THX1138.").captures
f1 # => "H"
f2 # => "X"
f3 # => "113"
```

end *match.end(n) → int*
match.end(name) → int

1.9 / Returns the offset in the original string of the end of the *n*th capture or the named capture.

```
m = /(.)().(\d+)(\d)/.match("THX1138.")
m.end(0)           # => 7
m.end(2)           # => 3
m = /..(?<digit_prefix>\d+)\d/.match("THX1138.")
m.end(:digit_prefix) # => 6
```

length *match.length → int*

Returns the number of elements in the match array.

```
m = /(.)().(\d+)(\d)/.match("THX1138.")
m.length # => 5
m.size   # => 5
```

names *match.names → array*

1.9 / Returns the list of named captures in the regular expression that created *match*.

```
m = /(?<prefix>[A-Z]+)(?<hyphen>-?)(?<digits>\d+)/.match("THX1138.")
m.names      # => ["prefix", "hyphen", "digits"]
m.captures   # => ["THX", "", "1138"]
m[:prefix]   # => "THX"
```

offset *match.offset(n) → array*
match.offset(name) → array

1.9 / Returns a two-element array containing the beginning and ending offsets of the *n*th or named capture.

```
m = /(.)().(\d+)(\d)/.match("THX1138.")
m.offset(0)           # => [1, 7]
m.offset(4)           # => [6, 7]
m = /..(?<digit_prefix>\d+)\d/.match("THX1138.")
m.offset(:digit_prefix) # => [3, 6]
```

post_match *match.post_match → string*

Returns the portion of the original string after the current match. Equivalent to the special variable \$'.

```
m = /(.)().(\d+)(\d)/.match("THX1138: The Movie")
m.post_match # => ": The Movie"
```

pre_match *match.pre_match → string*

Returns the portion of the original string before the current match. Equivalent to the special variable \$'.

```
m = /(.)().(\d+)(\d)/.match("THX1138.")
m.pre_match # => "T"
```

regexp *match.regexp* → *a_regexp*

1.9 Returns the regexp object for the regular expression that created *match*.

```
m = /(.)+(\d+)/.match("THX1138: The Movie")
m.regexp # => /(.)+(\d+)/
```

size *match.size* → *int*

A synonym for `MatchData#length`.

string *match.string* → *string*

Returns a frozen copy of the string passed in to `match`.

```
m = /(.)+(\d+)/.match("THX1138.")
m.string # => "THX1138."
```

to_a *match.to_a* → *array*

Returns the array of matches. Unlike `MatchData#captures`, returns the full string matched.

```
m = /(.)+(\d+)/.match("THX1138.")
m.to_a # => ["HX1138", "H", "X", "113", "8"]
```

to_s *match.to_s* → *string*

Returns the entire matched string.

```
m = /(.)+(\d+)/.match("THX1138.")
m.to_s # => "HX1138"
```

values_at *match.values_at*(< *index*)*) → *array*

Synonym for `MatchData#select`.