String

////////// Convertie //

"toto".**crypt**("terre")	=> "tegNlzW5VjNcM"	"ab".**intern**	=> :ab
"ab".**hash**	=> -565794218	"ab toto titi".**split**	=> ["ab", "toto", "titi"]
"ff".**hex**	=> 255	"5ab12ab23".**split**("ab")	=> ["5", "12", "23"]
"0xFFFF".**hex**	=> 65535	"5u12v23".**split**(/u|v/)	=> ["5", "12", "23"]
"-f".**hex**	=> -15	"1a2a3a4a5".**split**("a",3)	=> ["1", "2", "3a4a5"]
"a".**ord**	=> 97	"2+3i".**to_c**	=> (2+3i)
"bonjour".**partition**("jo")	=> ["bon", "jo", "ur"]	"12.5e3".**to_f**	=> 12500.0
"bonjour".**partition**(/o.*o/)	=> ["b", "onjo", "ur"]	"123ab".**to_i**	=> 123
"hello".**rpartition**("l")	=> ["hel", "l", "o"]	"11111111".**to_i**(2)	=> 255
"hello".**rpartition**(/.l/)	=> ["he", "ll", "o"]	"FFFF".**to_i**(16)	=> 65535
"ab uvw".**scan**(/\w+/)	=> ["ab", "uvw"]	"ab".**to_s**	=> "ab"
"abctoto aaa".**scan**(/a../)	=> ["abc", "aaa"]	"ab".**to_str**	=> "ab"
"a25fra78f".**scan**(/a(.)(.)f/)	=> [["2", "5"], ["7", "8"]]	"ab".**to_sym**	=> :ab

////////// Formate //

"%05d" **%** 22	=> "00022"	"aaa\naaa".**dump**	=> "\"aaa\\naaa\""
"%-5d" **%** 22	=> "22 "	"ab".**encode**("utf-8")	=> "ab"
"%b" **%** 5	=> "101"	"ab".**encode**("cp850","utf-8")	=> "ab"
"%x" **%** 255	=> "ff"	s.**encode!**(dst,src)	
"%.2f" **%** 3.1415	=> "3.14"	"ab".**encoding**	=> #<Encoding:UTF-8>
"bonjour".**capitalize**	=> "Bonjour"	"ab".**force_encoding**("utf-8")	=> "ab"
"BONJOUR".**capitalize**	=> "Bonjour"	"aaa\naaa".**inspect**	=> "\"aaa\\naaa\""
"123bonjour".**capitalize**	=> "123bonjour"	"toto".**ljust**(10)	=> "hello "
s="abc"; s.**capitalize!**; s	=> "ABC"	"toto".**ljust**(10,"123")	=> "hello12312"
"toto".**center**(15)	=> " toto "	" abc".**lstrip**	=> "abc"
"toto".**center**(15,"=/")	=> "=/=/=toto=/=/=/"	s=" abc"; s.**lstrip!**; s	=> "abc"
"toto\r\n".**chomp**	=> "toto"	"toto".**rjust**(10)	=> " hello"
"toto\r\n".**chomp**("to")	=> "to"	"toto".**rjust**(10,"123")	=> "12312hello"
s="ab"; s.**chomp!**; s	=> "ab"	"abc ".**rstrip**	=> "abc"
s.**chomp!**("b")	=> "a"	s="abc "; s.**rstrip!**; s	=> "abc"
"abc".**chop**	=> "ab"	" abc ".**strip**	=> "abc"
s="abc"; s.**chop!**; s	=> "ab"	s=" abc "; s.**strip!**; s	=> "abc"

////////// Énumère //

"abc".**bytes**	=> #<Enumerator: "abc":bytes>	97, 98, 99
"abc".**chars**	=> #<Enumerator: "abc":chars>	"a", "b", "c"
"abc".**codepoints**	=> #<Enumerator: "abc":codepoints>	97, 98, 99
"abc".**each_byte**	=> #<Enumerator: "abc":each_byte>	97, 98, 99
"abc".**each_char**	=> #<Enumerator: "abc":each_char>	"a", "b", "c"
"abc".**each_codepoint**	=> #<Enumerator: "abc":each_codepoint>	97, 98, 99
"ab\ncd\n".**each_line**	=> #<Enumerator: "ab\ncd\n":each_line>	"ab\n", "cd\n"
"abaribarobara".**each_line**("ba")	=> #<Enumerator: "abaribarobara":each_line("ba")>	"aba", "riba", "roba", "ra"
"ababa".**gsub**(/a/, "GG")	=> "GGbGGbGG"	
"ababa".**gsub**(/a/)	=> #<Enumerator: "abcabc":gsub(/a/)>	"a", "a", "a"
"hello".**gsub**(/e|o/){|x| x+x}	=> "heelloo"	
s="hello"; s.**gsub!**(/e|o/){|x| x+x}; s	=> "heelloo"	
s="hello"; s.**gsub!**(/l/, "GG"); s	=> "heGGGGo"	
"12io34io56".**lines**("io")	=> #<Enumerator: "ababa":lines("ba")>	"12io", "34io", "56"
"a5auvb".**match**("a(.)a(.*)b")	=> #<MatchData "a5auvb" 1:"5" 2:"uv">	
"a5auvb".**match**("a(.)a(.*)b")[1]	=> "5"	
"a8".**upto**("b2")	=> #<Enumerator: "a9":upto("aaa")>	"a8", "a9", "b0", "b1", "b2"

////////// Extrait //

"abcd"[0]	=> "a"	"abc".**chr**	=> "a"
"abcd"[3]	=> "d"	s.**clear**	=> ""
"abcd"[4]	=> nil	"abc".**getbyte**(1)	=> 98
"abcd"[-1]	=> "d"	"abc".**insert**(0,"X")	=> "Xabc"
"abcd"[-3]	=> "a"	"abc".**insert**(3,"X")	=> "abcX"
"abcdef"[2..3]	=> "cd"	"abc".**insert**(-1,"X")	=> "abcX"
"abcdef"[2,3]	=> "cde"	s.**replace**("abc")	=> "abc"
"abcdef"[/cd/]	=> "cd"	s="pp"; s.**setbyte**(1,97); s	=> "pa"
"abcdef"[/cd/,1]	=> nil	"abc".**slice**(1)	=> "b"
s="abcde".[2..4]="A";s	=> "abA"	"abc".**slice**(1..2)	=> "bc"
"abcab".**delete**("ab")	=> "c"	s="pqr"; s.**slice!**(1); s	=> "pr"
"abcab".**delete**("ab","bc","ba")	=> "aca"	"hello".**sub**(/[aeiou]/,"-")	=> "h-llo"
"abcab".**delete**("abc","^ab")	=> "abab"	"hello".**sub**(/[aeiou]/){|x| x.capitalize}	=> "hEllo"
s="ababa"; s.**delete!**("b"); s	=> "aaa"	s="hello"; s.**sub!**(/h../, "-")	=> "-lo"

///////// Compose //

"ga" * 3	=> "gagaga"	"abbbolli".**squeeze**	=> "aboli"
"ga" + "bu"	=> "gabu"	"aaaabbbbeeee".**squeeze**("a-d")	=> "abeeee"
"ga" << "bu"	=> "gabu"	"aaaabbbbeeee".**squeeze**("ab")	=> "abeeee"
"ab".**concat**("fg")	=> "abfg"	s="abba"; s.**squeeze!**; s	=> "aba"
"HELLO".**downcase**	=> "hello"	"aa".**succ**	=> "ab"
s="AB"; s.**downcase!**; s	=> "ab"	s="ab"; s.**succ!**; s	=> "ac"
"a".**next**	=> "b"	"aBcD".**swapcase**	=> "AbCd"
"1".**next**	=> "2"	s="aBcD"; s.**swapcase!**; s	=> "AbCd"
"zz".**next**	=> "aaa"	"abcaaa".**tr**("abc","uvw")	=> "uvwuuu"
"a9".**next**	=> "b0"	"abcdef".**tr**("abcd","+-")	=> "+---ef"
s="ab"; s.**next!**; s	=> "ac"	s="abcaaa"; s.**tr!**("abc","123"); s	=> "123111"
"123".**reverse**	=> "321"	"ab".**upcase**	=> "AB"
s="123"; s.**reverse!**; s	=> "321"	s="ab"; s.**upcase!**; s	=> "AB"

///////// Compare //

"a" <=> "b"	=> -1	"ABC".**casecmp**("abc")	=> 0
"b" <=> "a"	=> 1	"a".**casecmp**("b")	=> -1
"aa" <=> "a"	=> 1	"b".**casecmp**("a")	=> 1
"cd" <=> "cd"	=> 0	"aa".**casecmp**("a")	=> 1
"cd" == "cd"	=> true	"".**empty?**	=> true
2 == "2"	=> false	"abcd".**end_with?**("cd","fg")	=> true
"ab".**eql?**("ab")	=> true	"hello".**include**?("lo")	=> true
"ab".**ascii_only?**	=> true	"abc".**start_with?**("ab","cd")	=> true

///////// Recherche //

"abcdef" =~ /de/	=> 3	"hello".**index**("e")	=> 1
"ab\u0321".**bytesize**	=> 4	"hello".**index**("lo")	=> 3
"abcab".**count**("ab")	=> 4	"hello".**index**(/e|o/, 2)	=> 4
"abcab".**count**("ab","bc","ba")	=> 2	"abababab".**rindex**("ab")	=> 4
"abcab".**count**("abc","^ab")	=> 1	"abab".**rindex**(/a/,1)	=> 0
"ab.**length**.	=> 2	"abc".**size**	=> 3