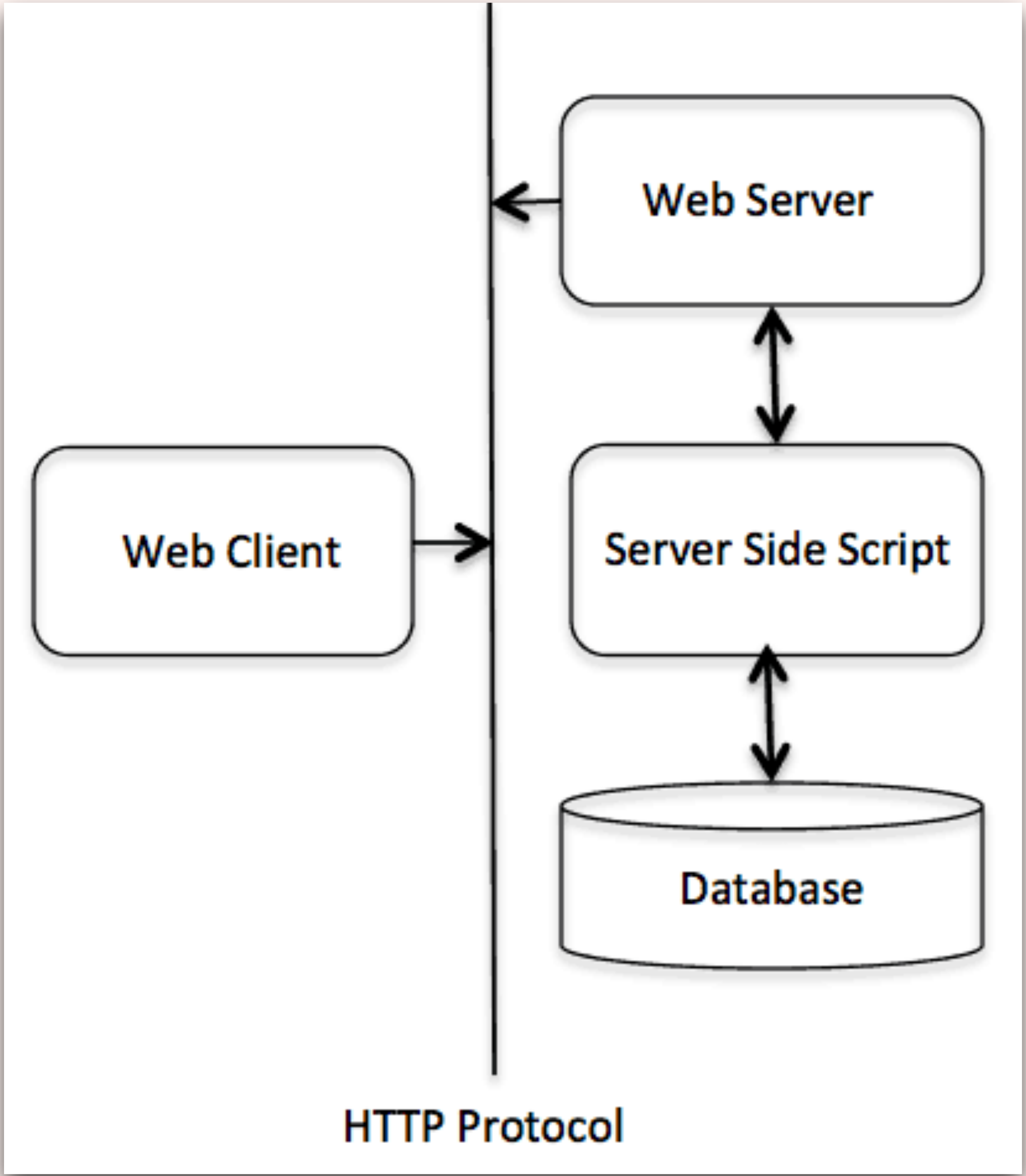
JAVASCRIPT

HTTP. Regular Expressions

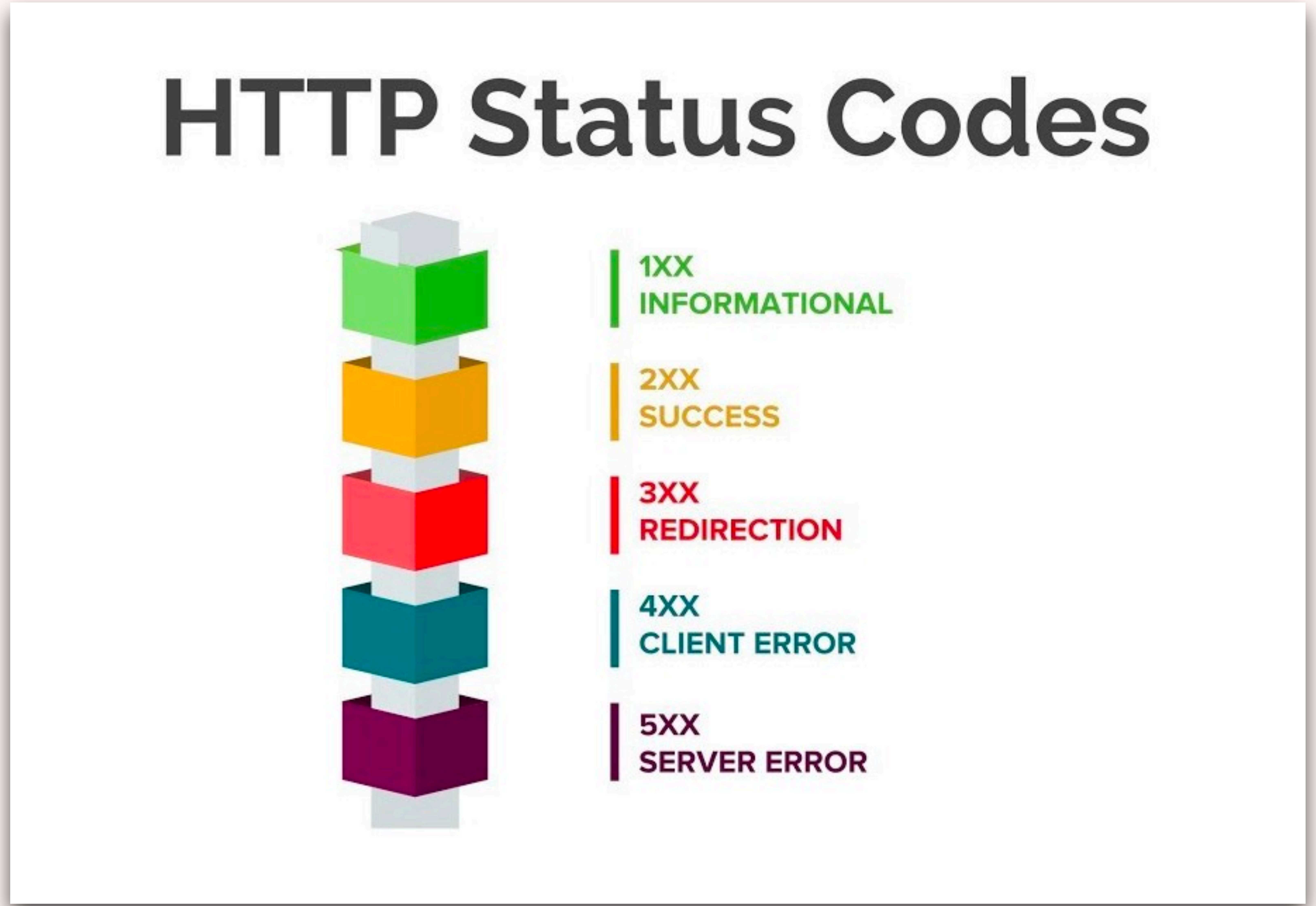
HTTP

- HTTP - Hyper Text Transfer Protocol
- Communication between clients and servers is done by requests and responses:
  - A client (a browser) sends an HTTP request to the web
  - An web server receives the request
  - The server runs an application to process the request
  - The server returns an HTTP response (output) to the browser
  - The client (the browser) receives the response


Web Server
Web Client
Server Side Script
Database
HTTP Protocol

HTTP REQUEST METHODS

- **GET** - requests a representation of the specified resource. Requests using GET should only retrieve data.
- **POST** - sends data to the server. The type of the body of the request is indicated by the Content-Type header.
- **HEAD** - requests the headers that are returned if the specified resource would be requested with an HTTP GET method.
- **PUT** - method creates a new resource or replaces a representation of the target resource with the request payload.
- **PATCH** - method applies partial modifications to a resource.

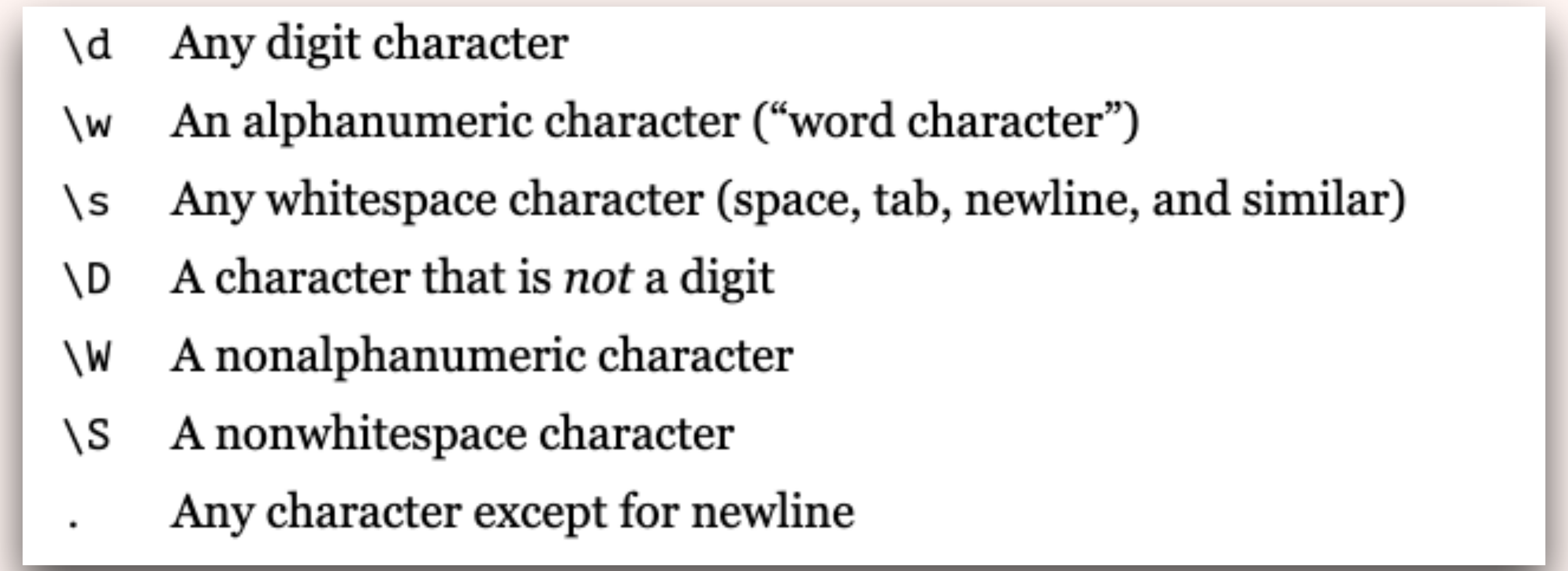

HTTP Status Codes
1XX
INFORMATIONAL
2XX
SUCCESS
3XX
REDIRECTION
4XX
CLIENT ERROR
5XX
SERVER ERROR

REGULAR EXPRESSIONS

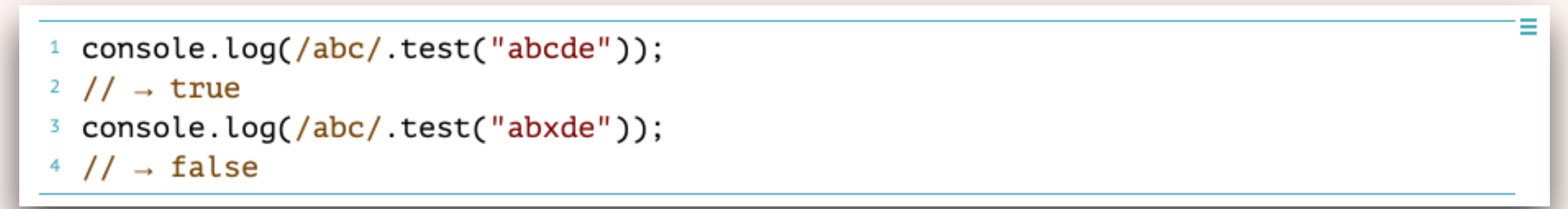
- Regular expression is an object that describes a pattern of characters.
- To perform powerful pattern-matching and search-and-replace functions on text.
- Two ways: **new RegExp(pattern, attributes)** or **/pattern/attributes**
  - **pattern** – A string that specifies the pattern of the regular expression or another regular expression.
  - **attributes** – An optional string containing any of the "g", "i", and "m" attributes that specify global, case-insensitive, and multi-line matches, respectively.

REGEXP METHODS

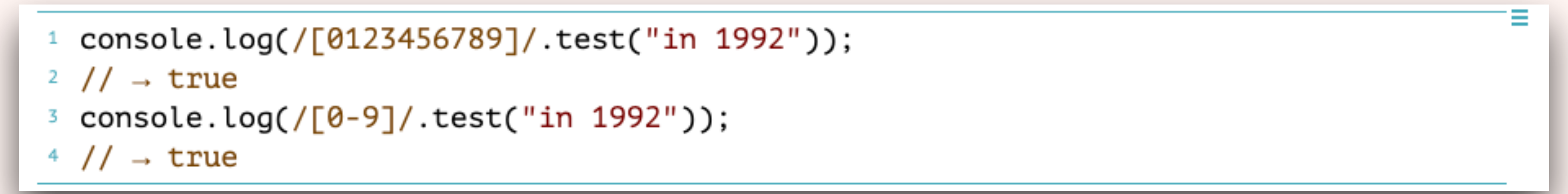
- **.test()** - searches a string for a pattern, and returns true or false, depending on the result.

- 

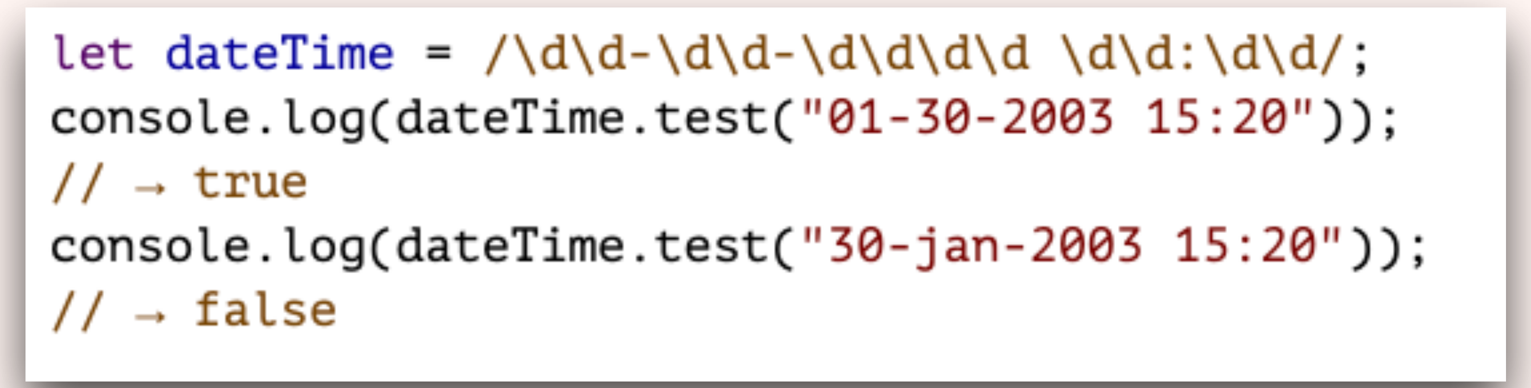
`\d`	Any digit character
`\w`	An alphanumeric character ("word character")
`\s`	Any whitespace character (space, tab, newline, and similar)
`\D`	A character that is *not* a digit
`\W`	A nonalphanumeric character
`\S`	A nonwhitespace character
`.`	Any character except for newline

- 

```
console.log(/abc/.test("abcde"));
// → true
console.log(/abc/.test("abxde"));
// → false
```

- 

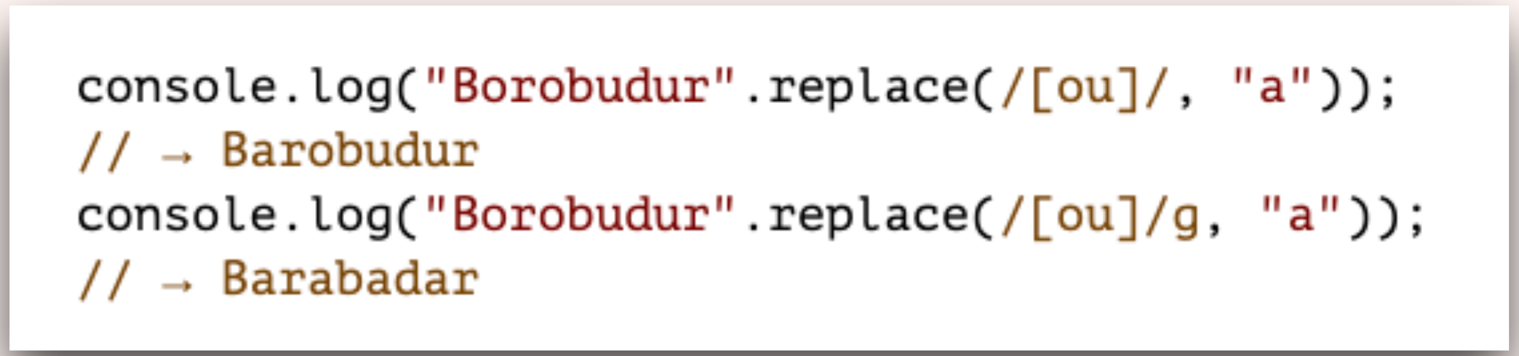
```
console.log(/[0123456789]/.test("in 1992"));
// → true
console.log(/[0-9]/.test("in 1992"));
// → true
```

- 

```
let dateTime = /\d\d-\d\d-\d\d\d\d \d\d:\d\d/;
console.log(dateTime.test("01-30-2003 15:20"));
// → true
console.log(dateTime.test("30-jan-2003 15:20"));
// → false
```

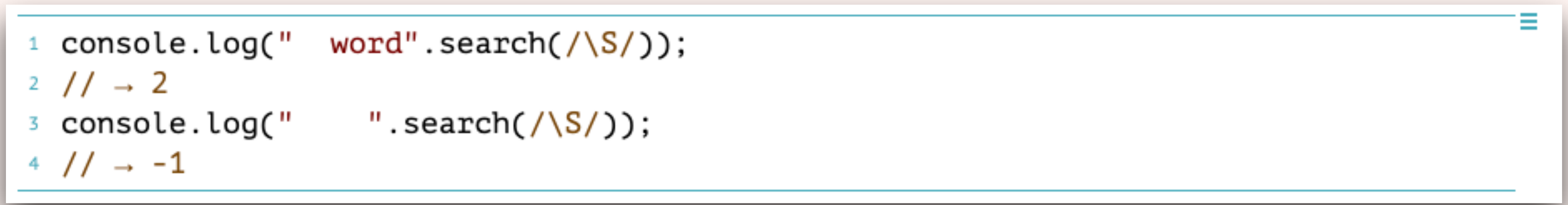
STRING METHODS

- **.replace()** - replaces a specified value with another value in a string.



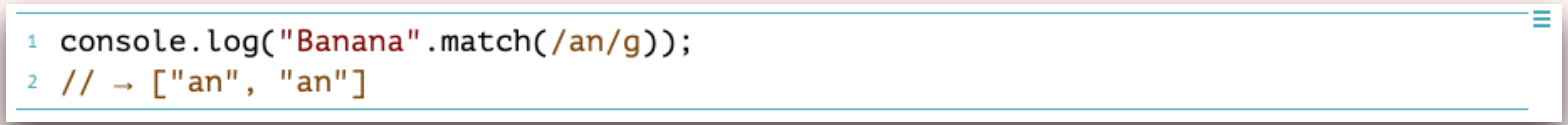
```
console.log("Borobudur".replace(/[ou]/, "a"));
// → Barobudur
console.log("Borobudur".replace(/[ou]/g, "a"));
// → Barabadar
```

- **.search()** - to search for a match, and returns the position of the match. Returns the first index on which the expression was found, or -1 when it wasn't found.



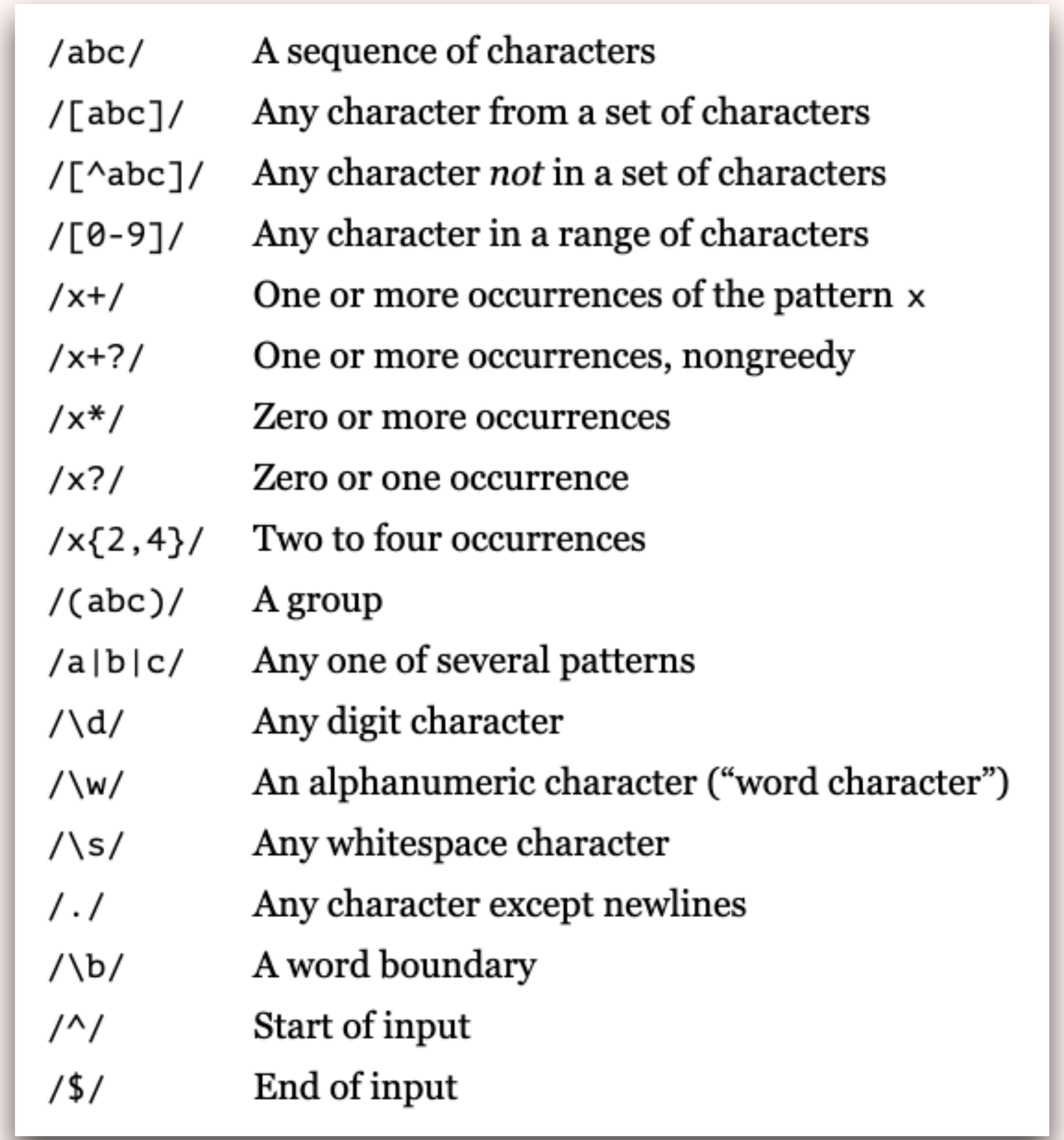
```
console.log("  word".search(/\S/));
// → 2
console.log("    ".search(/\S/));
// → -1
```

- **.match() -** find *all* matches of the pattern in the string and return an array containing the matched strings.



```
console.log("Banana".match(/an/g));
// → ["an", "an"]
```

PATTERNS



`/abc/`	A sequence of characters
`/[abc]/`	Any character from a set of characters
`/[^abc]/`	Any character *not* in a set of characters
`/[0-9]/`	Any character in a range of characters
`/x+/`	One or more occurrences of the pattern `x`
`/x+?/`	One or more occurrences, nongreedy
`/x*/`	Zero or more occurrences
`/x?/`	Zero or one occurrence
`/x{2,4}/`	Two to four occurrences
`/(abc)/`	A group
`/a|b|c/`	Any one of several patterns
`/\d/`	Any digit character
`/\w/`	An alphanumeric character ("word character")
`/\s/`	Any whitespace character
`/./`	Any character except newlines
`/\b/`	A word boundary
`/^/`	Start of input
`/$/`	End of input