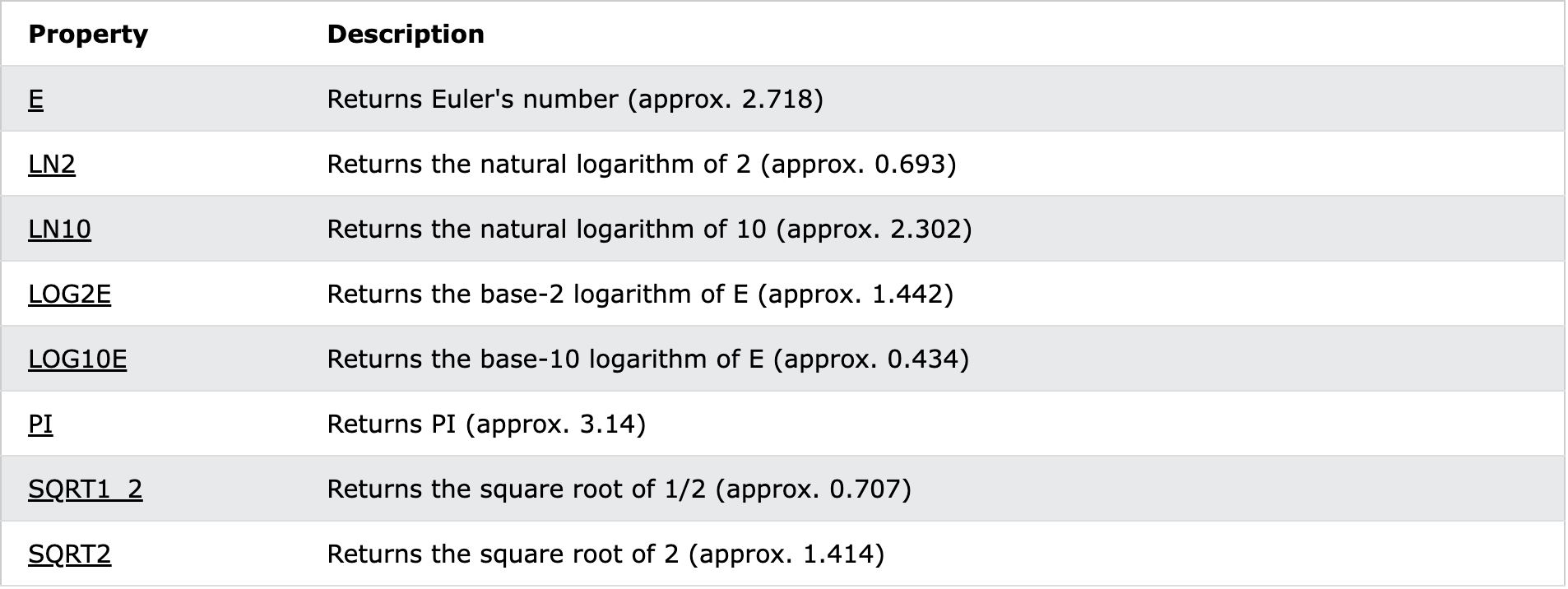
**JAVASCRIPT – Math**

* **Math Properties**



# **Math.E**

Math.E returns Euler's number, base of natural logarithms, approximately 2.718.

**let x = Math.E;**

# **Math.LN2**

Math.LN2 returns the natural logarithm of 2, approximately 0.693.

**let x = Math.LN2;**

* **Math.LN10**

Math.LN10 returns the natural logarithm of 10, approximately 2.302.

**let x = Math.LN10;**

* **Math.LOG2E**

Math.LOG2E returns the base-2 logarithm of E, approximately 1.442.

**let x = Math.LOG2E;**

# **Math.LOG10E**

Math.LOG10E returns the base-10 logarithm of E, approximately 0.434.

**let x = Math.LOG10E;**

# **Math.PI**

Math.PI returns PI (the ratio of a circle's area to the square of its radius, approximately 3.14)

**let x = Math.PI;**

# **Math.SQRT1\_2**

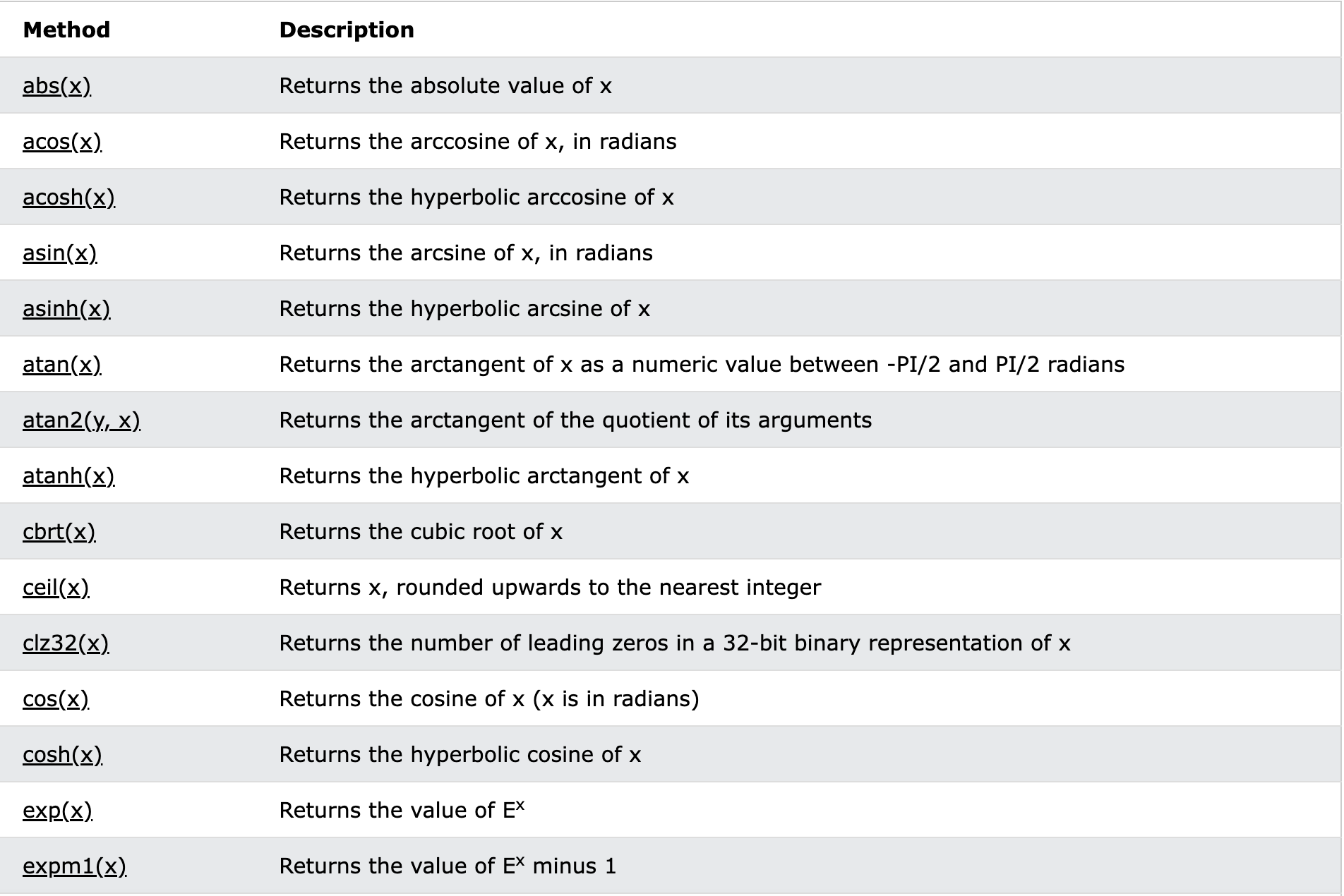
Math.SQRT1\_2 returns the square root of 1/2, approximately 0.707.

**let x = Math.SQRT1\_2;**

# **Math.SQRT2**

Math.SQRT2 returns the square root of 2, approximately 1.414.

* **Math Methods**



* **abs ()**

The Math.abs() method returns the absolute value of a number.

**let x = Math.abs(-7.25);**

* **acos ()**

The Math.acos() method returns the arccosine of a number as a value between 0 and PI radians.

Note: If the parameter x is outside the range -1 to 1, the method will return NaN.

Tip: -1 will return the value of PI.

**let x = Math.acos(0.5);**

* **asin ()**

The Math.asin() method returns the arcsine of a number as a value between -PI/2 and PI/2 radians.

Note: If the parameter x is outside the range -1 to 1, the browser will return NaN.

Tip: 1 will return the value of PI/2. -1 will return the value of -PI/2.

**let x = Math.asin(0.5);**

# **atan()**

The Math.atan() method returns the arctangent of a number as a value between -PI/2 and PI/2 radians.

**let x = Math.atan(2);**

# **atan2()**

The Math.atan2() method returns the arctangent of the quotient of its arguments, as a numeric value between PI and -PI radians.

The number returned represents the counterclockwise angle in radians (not degrees) between the positive X axis and the point (x, y).

Note: With atan2(), the y coordinate is passed as the first argument and the x coordinate is passed as the second argument.

**Math.atan2(8, 4);**

# **ceil()**

The Math.ceil() method rounds a number UPWARDS to the nearest integer, and returns the result.

**Math.ceil(1.4) // Rounds 1.4 upward to its nearest integer**

# **cos()**

The Math.cos() method returns the cosine of a number.

Note: Math.cos() returns a numeric value between -1 and 1, which represents the cosine of the angle.

**Math.cos(3)**

# **exp()**

The Math.exp() method returns the value of Ex, where [E is Euler's number](https://www.w3schools.com/jsref/jsref_e.asp) (approximately 2.7183) and x is the number passed to it.

**let x = Math.exp(3);**

# **floor()**

The Math.floor() method rounds a number DOWNWARDS to the nearest integer, and returns the result.

**let x = Math.floor(1.6);**

# **log()**

# The Math.log() method returns the natural logarithm (base E) of a number.

**let x = Math.log(2);**

# **max()**

The Math.max() method returns the number with the highest value.

**let x = Math.max(5, 10);**

# **min()**

The Math.max() method returns the number with the lowest value.

**let x = Math.min(5, 10);**

# **pow()**

The Math.pow() method returns the value of x to the power of y (xy).

**let x = Math.pow(4, 3);**

# **random()**

The Math.random() method returns a random number from 0 (inclusive) up to but not including 1 (exclusive).

**let x = Math.random();**

# **round()**

The Math.round() method rounds a number to the nearest integer.

Note: 2.49 will be rounded down (2), and 2.5 will be rounded up (3).

**let x = Math.round(2.5);**

# **sin()**

The Math.sin() method returns the sine of a number.

Note: Math.sin(x) returns a value between -1 and 1, which represents the sine of the parameter *x*.

**let x = Math.sin(3);**

# **sqrt()**

The Math.sqrt() method returns the square root of a number.

**let x = Math.sqrt(9);**

# **tan()**

The Math.tan() method returns the tangent of a number.

**let x = Math.tan(1);**

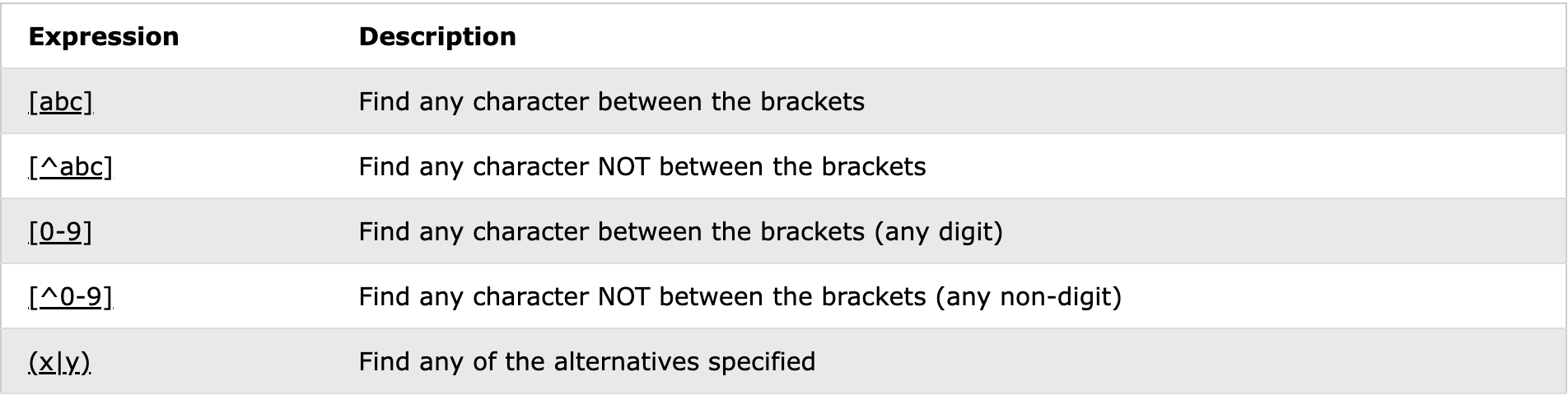
* **toSource()**

This method returns the string "Math". But this method does not work with IE.

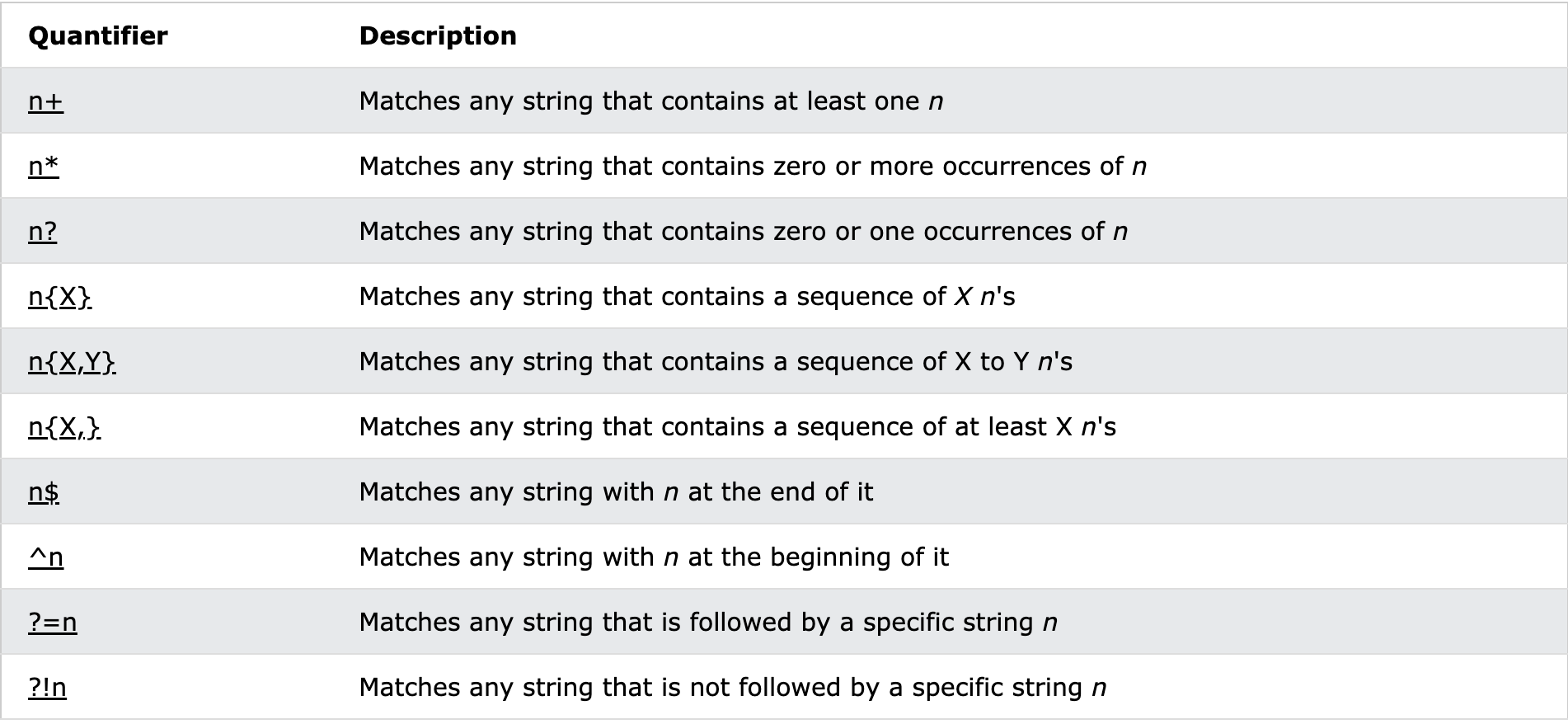
**Math.toSource();**

**JAVASCRIPT – RegExp**

* **Brackets**

Brackets are used to find a range of characters:

* **Quantifiers**



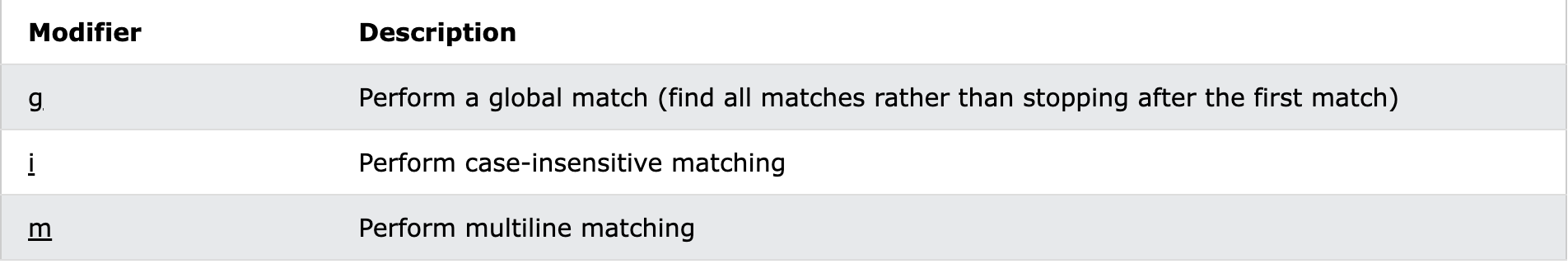
* **Literal Characters**

The most basic regular expression consists of a single literal character, such as a. **It matches the first occurrence of that character in the string**. If the string is Jack is a boy, it matches the a after the J.

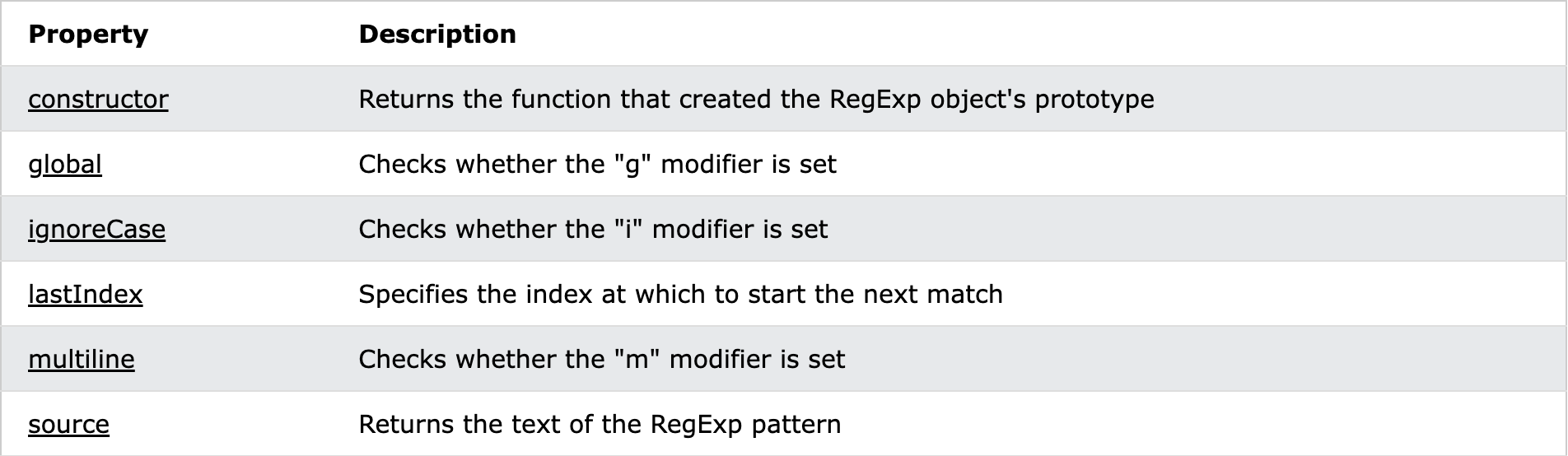
## **Metacharacters**

## **Modifiers**

Modifiers are used to perform case-insensitive and global searches:

****

* **RegExp Properties**



# **constructor**

The constructor property returns the function that created the RegExp prototype.

For a regular expression the constructor property returns:

function RegExp() { [native code] }

# **global**

The global property specifies whether or not the ["g" modifier](https://www.w3schools.com/jsref/jsref_regexp_g.asp) is set.

This property returns true if the "g" modifier is set, otherwise it returns false.

**let pattern = /W3S/g;**

**let result = pattern.global;**

# **ignoreCase**

The ignoreCase property specifies whether or not the ["i" modifier](https://www.w3schools.com/jsref/jsref_regexp_i.asp) is set.

This property returns true if the "i" modifier is set, otherwise it returns false.

**let pattern = /W3S/i;**

**let result = pattern.ignoreCase;**

# **lastIndex**

The lastIndex property specifies the index at which to start the next match.

Note: This property only works if the "g" modifier is set.

This property returns an integer that specifies the character position immediately after the last match found by exec( ) or test( ) methods.

Note: exec( ) and test( ) reset lastIndex to 0 if they do not get a match.

**let text = "The rain in Spain stays mainly in the plain";**

**let pattern = /ain/g;**

**let result = "";**

**while (pattern.test(text)==true) {**

**result += "Found at pos " + pattern.lastIndex + "<br>";**

**}**

# **multiline**

The multiline property specifies whether or not the m modifier is set.

This property returns true if the "m" modifier is set, otherwise it returns false.

**let text = "Visit W3Schools!";**

**let pattern = /W3S/gi; // "g" and "i" is set, "m" is not.**

**let result = pattern.multiline;**

# **source**

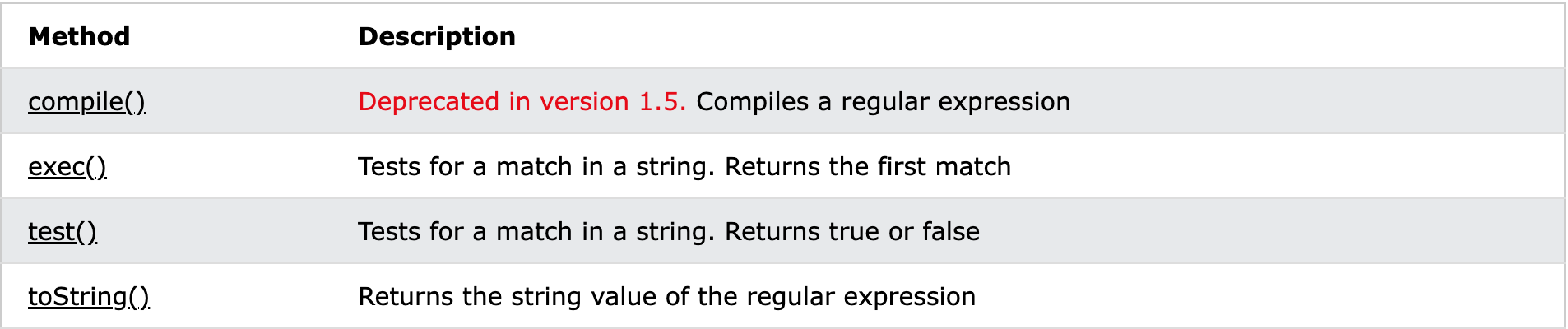
The source property returns the text of the RegExp pattern.

**let text = "Visit W3Schools";**

**let pattern = /W3S/g;**

**let result = pattern.source;**

* **RegExp Methods**

****

# **exec()**

The exec() method tests for a match in a string.

If it finds a match, it returns a result array, otherwise it returns null.

**let text = "The best things in life are free";**

**let result = /e/.exec(text);**

# **test()**

The test() method tests for a match in a string.

If it finds a match, it returns true, otherwise it returns false.

**let text = "The best things in life are free";**

**let pattern = /e/;**

**let result = pattern.test(text);**

* **toSource ( )**

The toSource method string represents the source code of the object. This method does not work with all the browsers.

**RegExpObject.toSource();**

# **toString()**

The toString() method returns the string value of the regular expression.

**let pattern = new RegExp("Hello World", "g");**

**let text = pattern.toString();**

**JAVASCRIPT – DOM**

* **The Legacy DOM**

This is the model which was introduced in early versions of JavaScript language. It is well supported by all browsers, but allows access only to certain key portions of documents, such as forms, form elements, and images.

This model provides several read-only properties, such as title, URL, and lastModified provides information about the document as a whole. Apart from that, there are various methods provided by this model which can be used to set and get document property values.

* **The W3C DOM**

The DOM is a W3C (World Wide Web Consortium) standard.

The DOM defines a standard for accessing documents:

*"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."*

The W3C DOM standard is separated into 3 different parts:

* Core DOM - standard model for all document types
* XML DOM - standard model for XML documents
* HTML DOM - standard model for HTML documents
* **The IE 4 DOM**

This document object model was introduced in Version 4 of Microsoft's Internet Explorer browser. IE 5 and later versions include support for most basic W3C DOM features.

* **DOM Compatibility**

