case of true || X, no matter what X is—even if it’s an expression that does something terrible—the result will be true, and X is never evaluated. The same goes for false && X, which is false and will ignore X. This is called short-circuit evaluation.

A single var statement may define multiple variables. The definitions must be separated by commas.

var one = 1, two = 2;

console.log(one + two);

// → 3

**KEYWORDS**

break case catch class const continue debugger

default delete do else enum export extends false

finally for function if implements import in

instanceof interface let new null package private

protected public return static super switch this

throw true try typeof var void while with yield

---------------------

do {

var name = prompt("Who are you?");

} while (!name);

console.log(name);

This program will force you to enter a name. It will ask again and again until it gets something that is not an empty string. Applying the ! operator will convert a value to Boolean type before negating it, and all strings except "" convert to true.

-------------------------

FOR LOOP

for (var current = 20; ; current++) {

2

if (current % 7 == 0)

3

break;

4

}

5

console.log(current);

6

// → 21

The for construct in the example does not have a part that checks for the end of the loop. This means that the loop will never stop unless the break statement inside is executed.

For counter += 1 and counter -= 1, there are even shorter equivalents: counter++ and counter--.

switch (prompt("What is the weather like?")) {

case "rainy":

console.log("Remember to bring an umbrella.");

break;

case "sunny":

console.log("Dress lightly.");

case "cloudy":

console.log("Go outside.");

break;

default:

console.log("Unknown weather type!");

break;

}

Chapter3

Power function

var power = function(base, exponent) {

var result = 1;

for (var count = 0; count < exponent; count++)

result \*= base;

return result;

};

console.log(power(2, 10));

// → 1024