ISO/IEC 16262:2002-06 (E)

Information technology - ECMAScript language specification

Contents

Page

1 Scope 1

2 Conformance 1

3 Normative References 1

4 Overview 2
4.1 Web Scripting 2
4.2 Language Overview 2
4.2.1 Objects 3
4.3 Terms and Definitions 4
4.3.1 Type 4
4.3.2 Primitive Value 4
4.3.3 Object 4
4.3.4 Constructor 4
4.3.5 Prototype 5
4.3.6 Native Object 5
4.3.7 Built-in Object 5
4.3.8 Host Object 5
4.3.9 Undefined Value 5
4.3.10 Undefined Type 5
4.3.11 Null Value 5
4.3.12 Null Type 5
4.3.13 Boolean Value 5
4.3.14 Boolean Type 5
4.3.15 Boolean Object 5
4.3.16 String Value 6
4.3.17 String Type 6
4.3.18 String Object 6
4.3.19 Number Value 6
4.3.20 Number Type 6
4.3.21 Number Object 6
4.3.22 Infinity 6
4.3.23 NaN 6

5 Notational Conventions 6
5.1 Syntactic and Lexical Grammars 6
5.1.1 Context-Free Grammars 6
5.1.2 The Lexical and RegExp Grammars 7
5.1.3 The Numeric String Grammar 7
5.1.4 The Syntactic Grammar 7
5.1.5 Grammar Notation 8
5.2 Algorithm Conventions 10

6 Source Text 11

7 Lexical Conventions 11
7.1 Unicode Format-Control Characters 12
7.2 White Space 12
7.3 Line Terminators 12
7.4 Comments 13
7.5 Tokens 14

7.5.1 Reserved Words 14
7.5.2 Keywords 14
7.5.3 Future Reserved Words 14
7.6 Identifiers 15
7.7 Punctuators 16
7.8 Literals 16
7.8.1 Null Literals 16
7.8.2 Boolean Literals 17
7.8.3 Numeric Literals 17
7.8.4 String Literals 19
7.8.5 Regular Expression Literals 21
7.9 Automatic Semicolon Insertion 22
7.9.1 Rules of Automatic Semicolon Insertion 22
7.9.2 Examples of Automatic Semicolon Insertion 23

8 Types 24
8.1 The Undefined Type 24
8.2 The Null Type 24
8.3 The Boolean Type 24
8.4 The String Type 24
8.5 The Number Type 25
8.6 The Object Type 26
8.6.1 Property Attributes 26
8.6.2 Internal Properties and Methods 26
8.7 The Reference Type 29
8.7.1 GetValue (V) 30
8.7.2 PutValue (V, W) 30
8.8 The List Type 30
8.9 The Completion Type 30

9 Type Conversion 30
9.1 ToPrimitive 30
9.2 ToBoolean 31
9.3 ToNumber 31
9.3.1 ToNumber Applied to the String Type 31
9.4 ToInteger 34
9.5 ToInt32: (Signed 32 Bit Integer) 34
9.6 ToUint32: (Unsigned 32 Bit Integer) 35
9.7 ToUint16: (Unsigned 16 Bit Integer) 35
9.8 ToString 35
9.8.1 ToString Applied to the Number Type 36
9.9 ToObject 37

10 Execution Contexts 37
10.1 Definitions 37
10.1.1 Function Objects 37
10.1.2 Types of Executable Code 37
10.1.3 Variable Instantiation 38
10.1.4 Scope Chain and Identifier Resolution 38
10.1.5 Global Object 38
10.1.6 Activation Object 39
10.1.7 This 39
10.1.8 Arguments Object 39
10.2 Entering An Execution Context 39
10.2.1 Global Code 39
10.2.2 Eval Code 40
10.2.3 Function Code 40

11 Expressions 40
11.1 Primary Expressions 40
11.1.1 The this Keyword 40
11.1.2 Identifier Reference 40

11.1.3 Literal Reference 40
11.1.4 Array Initialiser 40
11.1.5 Object Initialiser 42
11.1.6 The Grouping Operator 43
11.2 Left-Hand-Side Expressions 43
11.2.1 Property Accessors 44
11.2.2 The new Operator 44
11.2.3 Function Calls 45
11.2.4 Argument Lists 45
11.2.5 Function Expressions 45
11.3 Postfix Expressions 46
11.3.1 Postfix Increment Operator 46
11.3.2 Postfix Decrement Operator 46
11.4 Unary Operators 46
11.4.1 The delete Operator 46
11.4.2 The void Operator 47
11.4.3 The typeof Operator 47
11.4.4 Prefix Increment Operator 47
11.4.5 Prefix Decrement Operator 47
11.4.6 Unary + Operator 47
11.4.7 Unary - Operator 48
11.4.8 Bitwise NOT Operator (~) 48
11.4.9 Logical NOT Operator (!) 48
11.5 Multiplicative Operators 48
11.5.1 Applying the * Operator 49
11.5.2 Applying the / Operator 49
11.5.3 Applying the % Operator 49
11.6 Additive Operators 50
11.6.1 The Addition operator (+) 50
11.6.2 The Subtraction Operator (-) 51
11.6.3 Applying the Additive Operators (+,-) to Numbers 51
11.7 Bitwise Shift Operators 51
11.7.1 The Left Shift Operator (<<) 51
11.7.2 The Signed Right Shift Operator (>>) 52
11.7.3 The Unsigned Right Shift Operator (>>>) 52
11.8 Relational Operators 52
11.8.1 The Less-than Operator (<) 53
11.8.2 The Greater-than Operator (>) 53
11.8.3 The Less-than-or-equal Operator (<=) 53
11.8.4 The Greater-than-or-equal Operator (>=) 53
11.8.5 The Abstract Relational Comparison Algorithm 54
11.8.6 The instanceof operator 54
11.8.7 The in operator 54
11.9 Equality Operators 55
11.9.1 The Equals Operator (==) 55
11.9.2 The Does-not-equals Operator (!=) 55
11.9.3 The Abstract Equality Comparison Algorithm 56
11.9.4 The Strict Equals Operator (===) 57
11.9.5 The Strict Does-not-equal Operator (!==) 57
11.9.6 The Strict Equality Comparison Algorithm 57
11.10 Binary Bitwise Operators 57
11.11 Binary Logical Operators 58
11.12 Conditional Operator (?:) 59
11.13 Assignment Operators 60
11.13.1 Simple Assignment (=) 60
11.13.2 Compound Assignment (op=) 60
11.14 Comma Operator (,) 60

12 Statements 61
12.1 Block 61
12.2 Variable statement 62
12.3 Empty Statement 63

12.4 Expression Statement 63
12.5 The if Statement 64
12.6 Iteration Statements 64
12.6.1 The do-while Statement 65
12.6.2 The while statement 65
12.6.3 The for Statement 65
12.6.4 The for-in Statement 66
12.7 The continue Statement 67
12.8 The break Statement 67
12.9 The return Statement 68
12.10 The with Statement 68
12.11 The switch Statement 68
12.12 Labelled Statements 70
12.13 The throw statement 70
12.14 The try statement 71

13 Function Definition 72
13.1 Definitions 73
13.1.1 Equated Grammar Productions 73
13.1.2 Joined Objects 73
13.2 Creating Function Objects 73
13.2.1 [[Call]] 75
13.2.2 [[Construct]] 75

14 Program 75

15 Native ECMAScript Objects 76
15.1 The Global Object 77
15.1.1 Value Properties of the Global Object 77
15.1.2 Function Properties of the Global Object 77
15.1.3 URI Handling Function Properties 79
15.1.4 Constructor Properties of the Global Object 83
15.1.5 Other Properties of the Global Object 84
15.2 Object Objects 84
15.2.1 The Object Constructor Called as a Function 84
15.2.2 The Object Constructor 84
15.2.3 Properties of the Object Constructor 84
15.2.4 Properties of the Object Prototype Object 85
15.2.5 Properties of Object Instances 86
15.3 Function Objects 86
15.3.1 The Function Constructor Called as a Function 86
15.3.2 The Function Constructor 86
15.3.3 Properties of the Function Constructor 87
15.3.4 Properties of the Function Prototype Object 87
15.3.5 Properties of Function Instances 88
15.4 Array Objects 88
15.4.1 The Array Constructor Called as a Function 89
15.4.2 The Array Constructor 89
15.4.3 Properties of the Array Constructor 89
15.4.4 Properties of the Array Prototype Object 90
15.4.5 Properties of Array Instances 97
15.5 String Objects 98
15.5.1 The String Constructor Called as a Function 98
15.5.2 The String Constructor 98
15.5.3 Properties of the String Constructor 99
15.5.4 Properties of the String Prototype Object 99
15.5.5 Properties of String Instances 106
15.6 Boolean Objects 107
15.6.1 The Boolean Constructor Called as a Function 107
15.6.2 The Boolean Constructor 107
15.6.3 Properties of the Boolean Constructor 107
15.6.4 Properties of the Boolean Prototype Object 107

15.6.5 Properties of Boolean Instances 108
15.7 Number Objects 108
15.7.1 The Number Constructor Called as a Function 108
15.7.2 The Number Constructor 108
15.7.3 Properties of the Number Constructor 108
15.7.4 Properties of the Number Prototype Object 109
15.7.5 Properties of Number Instances 112
15.8 The Math Object 112
15.8.1 Value Properties of the Math Object 112
15.8.2 Function Properties of the Math Object 113
15.9 Date Objects 118
15.9.1 Overview of Date Objects and Definitions of Internal Operators 118
15.9.2 The Date Constructor Called as a Function 122
15.9.3 The Date Constructor 122
15.9.4 Properties of the Date Constructor 123
15.9.5 Properties of the Date Prototype Object 124
15.9.6 Properties of Date Instances 131
15.10 RegExp (Regular Expression) Objects 131
15.10.1 Patterns 131
15.10.2 Pattern Semantics 133
15.10.3 The RegExp Constructor Called as a Function 145
15.10.4 The RegExp Constructor 145
15.10.5 Properties of the RegExp Constructor 146
15.10.6 Properties of the RegExp Prototype Object 146
15.10.7 Properties of RegExp Instances 147
15.11 Error Objects 148
15.11.1 The Error Constructor Called as a Function 148
15.11.2 The Error Constructor 148
15.11.3 Properties of the Error Constructor 148
15.11.4 Properties of the Error Prototype Object 148
15.11.5 Properties of Error Instances 149
15.11.6 Native Error Types Used in This Standard 149
15.11.7 NativeError Object Structure 149

16 Errors 151

Annex A (informative) Grammar Summary 152

A.1 Lexical Grammar 152
A.2 Number Conversions 156
A.3 Expressions 157
A.4 Statements 161
A.5 Functions and Programs 163
A.6 Universal Resource Identifier Character Classes 163
A.7 Regular Expressions 164

Annex B (informative) Compatibility 166

B.1 Additional Syntax 166
B.1.1 Numeric Literals 166
B.1.2 String Literals 166
B.2 Additional Properties 167
B.2.1 escape (string) 167
B.2.2 unescape (string) 168
B.2.3 String.prototype.substr (start, length) 168
B.2.4 Date.prototype.getYear ( ) 169
B.2.5 Date.prototype.setYear (year) 169
B.2.6 Date.prototype.toGMTString ( ) 169