**Speaking JavaScript**

## 

Table of contents

1. [Praise for *Speaking JavaScript*](http://speakingjs.com/es5/dedication.html)
2. [Preface](http://speakingjs.com/es5/pr02.html)
   1. [What You Need to Know About This Book](http://speakingjs.com/es5/pr02.html" \l "_what_you_need_to_know_about_this_book)
   2. [Tips for Reading This Book](http://speakingjs.com/es5/pr02.html" \l "how_to_read)
      1. [The Four Parts of This Book](http://speakingjs.com/es5/pr02.html" \l "book_parts)
      2. [JavaScript Command Lines](http://speakingjs.com/es5/pr02.html" \l "_javascript_command_lines)
      3. [Notational Conventions](http://speakingjs.com/es5/pr02.html" \l "_notational_conventions)
         1. [Describing syntax](http://speakingjs.com/es5/pr02.html" \l "_describing_syntax)
         2. [Referring to methods](http://speakingjs.com/es5/pr02.html" \l "_referring_to_methods)
         3. [Command-line interaction](http://speakingjs.com/es5/pr02.html" \l "_command_line_interaction)
         4. [Tips, notes, and warnings](http://speakingjs.com/es5/pr02.html" \l "_tips_notes_and_warnings)
      4. [Quickly Finding Documentation](http://speakingjs.com/es5/pr02.html" \l "_quickly_finding_documentation)
   3. [Safari® Books Online](http://speakingjs.com/es5/pr02.html" \l "_safari_books_online)
   4. [How to Contact Us](http://speakingjs.com/es5/pr02.html" \l "_how_to_contact_us)
   5. [Acknowledgments](http://speakingjs.com/es5/pr02.html" \l "book_acknowledgements)
      1. [Preparing for JavaScript](http://speakingjs.com/es5/pr02.html" \l "_preparing_for_javascript)
      2. [Help with JavaScript](http://speakingjs.com/es5/pr02.html" \l "_help_with_javascript)
      3. [Reviewers](http://speakingjs.com/es5/pr02.html" \l "_reviewers)
3. [I. JavaScript Quick Start](http://speakingjs.com/es5/pt01.html)
   1. [1. Basic JavaScript](http://speakingjs.com/es5/ch01.html)
      1. [Background](http://speakingjs.com/es5/ch01.html" \l "_background)
         1. [JavaScript Versus ECMAScript](http://speakingjs.com/es5/ch01.html" \l "_javascript_versus_ecmascript)
         2. [Influences and Nature of the Language](http://speakingjs.com/es5/ch01.html" \l "_influences_and_nature_of_the_language)
      2. [Syntax](http://speakingjs.com/es5/ch01.html" \l "_syntax)
         1. [An Overview of the Syntax](http://speakingjs.com/es5/ch01.html" \l "_an_overview_of_the_syntax)
         2. [Statements Versus Expressions](http://speakingjs.com/es5/ch01.html" \l "_statements_versus_expressions)
         3. [Semicolons](http://speakingjs.com/es5/ch01.html" \l "_semicolons)
         4. [Comments](http://speakingjs.com/es5/ch01.html" \l "_comments)
      3. [Variables and Assignment](http://speakingjs.com/es5/ch01.html" \l "_variables_and_assignment)
         1. [Assignment](http://speakingjs.com/es5/ch01.html" \l "_assignment)
         2. [Compound Assignment Operators](http://speakingjs.com/es5/ch01.html" \l "_compound_assignment_operators)
         3. [Identifiers and Variable Names](http://speakingjs.com/es5/ch01.html" \l "basic_identifiers_variable_names)
      4. [Values](http://speakingjs.com/es5/ch01.html" \l "_values)
         1. [Primitive Values Versus Objects](http://speakingjs.com/es5/ch01.html" \l "basic_prim_vs_obj)
         2. [Primitive Values](http://speakingjs.com/es5/ch01.html" \l "_primitive_values)
         3. [Objects](http://speakingjs.com/es5/ch01.html" \l "_objects)
         4. [undefined and null](http://speakingjs.com/es5/ch01.html" \l "basic_undefined_null)
         5. [Categorizing Values Using typeof and instanceof](http://speakingjs.com/es5/ch01.html" \l "_categorizing_values_using_typeof_and_instanceof)
      5. [Booleans](http://speakingjs.com/es5/ch01.html" \l "basic_booleans)
         1. [Truthy and Falsy](http://speakingjs.com/es5/ch01.html" \l "basic_truthy_falsy)
         2. [Binary Logical Operators](http://speakingjs.com/es5/ch01.html" \l "_binary_logical_operators)
         3. [Equality Operators](http://speakingjs.com/es5/ch01.html" \l "basic_equality_operators)
      6. [Numbers](http://speakingjs.com/es5/ch01.html" \l "basic_numbers)
      7. [Operators](http://speakingjs.com/es5/ch01.html" \l "_operators)
      8. [Strings](http://speakingjs.com/es5/ch01.html" \l "basic_strings)
         1. [String Operators](http://speakingjs.com/es5/ch01.html" \l "_string_operators)
         2. [String Methods](http://speakingjs.com/es5/ch01.html" \l "_string_methods)
      9. [Statements](http://speakingjs.com/es5/ch01.html" \l "_statements)
         1. [Conditionals](http://speakingjs.com/es5/ch01.html" \l "_conditionals)
         2. [Loops](http://speakingjs.com/es5/ch01.html" \l "_loops)
      10. [Functions](http://speakingjs.com/es5/ch01.html" \l "_functions)
          1. [Function Declarations Are Hoisted](http://speakingjs.com/es5/ch01.html" \l "_function_declarations_are_hoisted)
          2. [The Special Variable arguments](http://speakingjs.com/es5/ch01.html" \l "_the_special_variable_arguments)
          3. [Too Many or Too Few Arguments](http://speakingjs.com/es5/ch01.html" \l "_too_many_or_too_few_arguments)
          4. [Optional Parameters](http://speakingjs.com/es5/ch01.html" \l "_optional_parameters)
          5. [Enforcing an Arity](http://speakingjs.com/es5/ch01.html" \l "_enforcing_an_arity)
          6. [Converting arguments to an Array](http://speakingjs.com/es5/ch01.html" \l "basic_toarray)
      11. [Exception Handling](http://speakingjs.com/es5/ch01.html" \l "_exception_handling)
      12. [Strict Mode](http://speakingjs.com/es5/ch01.html" \l "_strict_mode)
      13. [Variable Scoping and Closures](http://speakingjs.com/es5/ch01.html" \l "basic_var_scope_and_closures)
          1. [Variables Are Function-Scoped](http://speakingjs.com/es5/ch01.html" \l "_variables_are_function_scoped)
          2. [Variables Are Hoisted](http://speakingjs.com/es5/ch01.html" \l "basic_var_hoisting)
          3. [Closures](http://speakingjs.com/es5/ch01.html" \l "_closures)
          4. [The IIFE Pattern: Introducing a New Scope](http://speakingjs.com/es5/ch01.html" \l "_the_iife_pattern_introducing_a_new_scope)
      14. [Objects and Constructors](http://speakingjs.com/es5/ch01.html" \l "_objects_and_constructors)
          1. [Single Objects](http://speakingjs.com/es5/ch01.html" \l "basic_single_objects)
          2. [Arbitrary Property Keys](http://speakingjs.com/es5/ch01.html" \l "_arbitrary_property_keys)
          3. [Extracting Methods](http://speakingjs.com/es5/ch01.html" \l "_extracting_methods)
          4. [Functions Inside a Method](http://speakingjs.com/es5/ch01.html" \l "_functions_inside_a_method)
          5. [Constructors: Factories for Objects](http://speakingjs.com/es5/ch01.html" \l "basic_constructors)
      15. [Arrays](http://speakingjs.com/es5/ch01.html" \l "basic_arrays)
          1. [Array Literals](http://speakingjs.com/es5/ch01.html" \l "_array_literals)
          2. [Array Methods](http://speakingjs.com/es5/ch01.html" \l "_array_methods)
          3. [Iterating over Arrays](http://speakingjs.com/es5/ch01.html" \l "_iterating_over_arrays)
      16. [Regular Expressions](http://speakingjs.com/es5/ch01.html" \l "basic_regexps)
          1. [Method test(): Is There a Match?](http://speakingjs.com/es5/ch01.html" \l "_method_test_is_there_a_match)
          2. [Method exec(): Match and Capture Groups](http://speakingjs.com/es5/ch01.html" \l "_method_exec_match_and_capture_groups)
          3. [Method replace(): Search and Replace](http://speakingjs.com/es5/ch01.html" \l "_method_replace_search_and_replace)
      17. [Math](http://speakingjs.com/es5/ch01.html" \l "basic_math)
      18. [Other Functionality of the Standard Library](http://speakingjs.com/es5/ch01.html" \l "_other_functionality_of_the_standard_library)
4. [II. Background](http://speakingjs.com/es5/pt02.html)
   1. [2. Why JavaScript?](http://speakingjs.com/es5/ch02.html)
      1. [Is JavaScript Freely Available?](http://speakingjs.com/es5/ch02.html" \l "_is_javascript_freely_available)
      2. [Is JavaScript Elegant?](http://speakingjs.com/es5/ch02.html" \l "_is_javascript_elegant)
      3. [Is JavaScript Useful?](http://speakingjs.com/es5/ch02.html" \l "_is_javascript_useful)
         1. [Graphical User Interfaces](http://speakingjs.com/es5/ch02.html" \l "_graphical_user_interfaces)
         2. [Other Technologies Complementing JavaScript](http://speakingjs.com/es5/ch02.html" \l "_other_technologies_complementing_javascript)
      4. [Does JavaScript Have Good Tools?](http://speakingjs.com/es5/ch02.html" \l "_does_javascript_have_good_tools)
      5. [Is JavaScript Fast Enough?](http://speakingjs.com/es5/ch02.html" \l "asm.js)
      6. [Is JavaScript Widely Used?](http://speakingjs.com/es5/ch02.html" \l "_is_javascript_widely_used)
      7. [Does JavaScript Have a Future?](http://speakingjs.com/es5/ch02.html" \l "_does_javascript_have_a_future)
      8. [Conclusion](http://speakingjs.com/es5/ch02.html" \l "_conclusion)
   2. [3. The Nature of JavaScript](http://speakingjs.com/es5/ch03.html)
      1. [Quirks and Unorthodox Features](http://speakingjs.com/es5/ch03.html" \l "_quirks_and_unorthodox_features)
      2. [Elegant Parts](http://speakingjs.com/es5/ch03.html" \l "_elegant_parts)
      3. [Influences](http://speakingjs.com/es5/ch03.html" \l "_influences)
   3. [4. How JavaScript Was Created](http://speakingjs.com/es5/ch04.html)
   4. [5. Standardization: ECMAScript](http://speakingjs.com/es5/ch05.html)
   5. [6. Historical JavaScript Milestones](http://speakingjs.com/es5/ch06.html)
5. [III. JavaScript in Depth](http://speakingjs.com/es5/pt03.html)
   1. [7. JavaScript’s Syntax](http://speakingjs.com/es5/ch07.html)
      1. [An Overview of the Syntax](http://speakingjs.com/es5/ch07.html" \l "_an_overview_of_the_syntax_2)
      2. [Comments](http://speakingjs.com/es5/ch07.html" \l "_comments_2)
      3. [Expressions Versus Statements](http://speakingjs.com/es5/ch07.html" \l "expr_vs_stmt)
         1. [Expressions](http://speakingjs.com/es5/ch07.html" \l "_expressions)
         2. [Statements](http://speakingjs.com/es5/ch07.html" \l "expression_statement)
      4. [Control Flow Statements and Blocks](http://speakingjs.com/es5/ch07.html" \l "_control_flow_statements_and_blocks)
      5. [Rules for Using Semicolons](http://speakingjs.com/es5/ch07.html" \l "_rules_for_using_semicolons)
         1. [No Semicolon After a Statement Ending with a Block](http://speakingjs.com/es5/ch07.html" \l "_no_semicolon_after_a_statement_ending_with_a_block)
         2. [The Empty Statement](http://speakingjs.com/es5/ch07.html" \l "_the_empty_statement)
         3. [Automatic Semicolon Insertion](http://speakingjs.com/es5/ch07.html" \l "automatic_semicolon_insertion)
      6. [Legal Identifiers](http://speakingjs.com/es5/ch07.html" \l "identifiers)
      7. [Invoking Methods on Number Literals](http://speakingjs.com/es5/ch07.html" \l "_invoking_methods_on_number_literals)
      8. [Strict Mode](http://speakingjs.com/es5/ch07.html" \l "strict_mode)
         1. [Switching on Strict Mode](http://speakingjs.com/es5/ch07.html" \l "_switching_on_strict_mode)
         2. [Strict Mode: Recommended, with Caveats](http://speakingjs.com/es5/ch07.html" \l "_strict_mode_recommended_with_caveats)
         3. [Variables Must Be Declared in Strict Mode](http://speakingjs.com/es5/ch07.html" \l "_variables_must_be_declared_in_strict_mode)
         4. [Functions in Strict Mode](http://speakingjs.com/es5/ch07.html" \l "_functions_in_strict_mode)
         5. [Setting and Deleting Immutable Properties Fails with an Exception in Strict Mode](http://speakingjs.com/es5/ch07.html" \l "_setting_and_deleting_immutable_properties_fails_with_an_exception_in_strict_mode)
         6. [Unqualified Identifiers Can’t Be Deleted in Strict Mode](http://speakingjs.com/es5/ch07.html" \l "_unqualified_identifiers_can_8217_t_be_deleted_in_strict_mode)
         7. [eval() Is Cleaner in Strict Mode](http://speakingjs.com/es5/ch07.html" \l "_eval_is_cleaner_in_strict_mode)
         8. [Features That Are Forbidden in Strict Mode](http://speakingjs.com/es5/ch07.html" \l "_features_that_are_forbidden_in_strict_mode)
   2. [8. Values](http://speakingjs.com/es5/ch08.html)
      1. [JavaScript’s Type System](http://speakingjs.com/es5/ch08.html" \l "_javascript_s_type_system)
         1. [JavaScript’s Types](http://speakingjs.com/es5/ch08.html" \l "javascript_types)
         2. [Static Versus Dynamic](http://speakingjs.com/es5/ch08.html" \l "_static_versus_dynamic)
         3. [Static Typing Versus Dynamic Typing](http://speakingjs.com/es5/ch08.html" \l "_static_typing_versus_dynamic_typing)
         4. [Static Type Checking Versus Dynamic Type Checking](http://speakingjs.com/es5/ch08.html" \l "_static_type_checking_versus_dynamic_type_checking)
         5. [Coercion](http://speakingjs.com/es5/ch08.html" \l "_coercion)
      2. [Primitive Values Versus Objects](http://speakingjs.com/es5/ch08.html" \l "_primitive_values_versus_objects)
      3. [Primitive Values](http://speakingjs.com/es5/ch08.html" \l "primitive_values)
      4. [Objects](http://speakingjs.com/es5/ch08.html" \l "object_values)
      5. [undefined and null](http://speakingjs.com/es5/ch08.html" \l "undefined_null)
         1. [Occurrences of undefined and null](http://speakingjs.com/es5/ch08.html" \l "_occurrences_of_undefined_and_null)
         2. [Checking for undefined or null](http://speakingjs.com/es5/ch08.html" \l "_checking_for_undefined_or_null_2)
         3. [The History of undefined and null](http://speakingjs.com/es5/ch08.html" \l "_the_history_of_undefined_and_null)
         4. [Changing undefined](http://speakingjs.com/es5/ch08.html" \l "changing_undefined)
      6. [Wrapper Objects for Primitives](http://speakingjs.com/es5/ch08.html" \l "wrapper_objects)
         1. [Wrapper Objects Are Different from Primitives](http://speakingjs.com/es5/ch08.html" \l "_wrapper_objects_are_different_from_primitives)
         2. [Wrapping and Unwrapping Primitives](http://speakingjs.com/es5/ch08.html" \l "_wrapping_and_unwrapping_primitives)
         3. [Primitives Borrow Their Methods from Wrappers](http://speakingjs.com/es5/ch08.html" \l "primitive_methods_via_wrappers)
      7. [Type Coercion](http://speakingjs.com/es5/ch08.html" \l "type_coercion)
         1. [Type Coercion Can Hide Bugs](http://speakingjs.com/es5/ch08.html" \l "_type_coercion_can_hide_bugs)
         2. [Functions for Converting to Boolean, Number, String, and Object](http://speakingjs.com/es5/ch08.html" \l "convert_to_primitive)
         3. [Algorithm: ToPrimitive()—Converting a Value to a Primitive](http://speakingjs.com/es5/ch08.html" \l "toprimitive)
   3. [9. Operators](http://speakingjs.com/es5/ch09.html)
      1. [Operators and Objects](http://speakingjs.com/es5/ch09.html" \l "_operators_and_objects)
      2. [Assignment Operators](http://speakingjs.com/es5/ch09.html" \l "_assignment_operators)
         1. [Compound Assignment Operators](http://speakingjs.com/es5/ch09.html" \l "_compound_assignment_operators_2)
      3. [Equality Operators: === Versus ==](http://speakingjs.com/es5/ch09.html" \l "equality_operators)
         1. [Strict Equality (===, !==)](http://speakingjs.com/es5/ch09.html" \l "_strict_equality)
         2. [Normal (Lenient) Equality (==, !=)](http://speakingjs.com/es5/ch09.html" \l "normal_equality)
         3. [There Are No Valid Use Cases for ==](http://speakingjs.com/es5/ch09.html" \l "_there_are_no_valid_use_cases_for)
      4. [Ordering Operators](http://speakingjs.com/es5/ch09.html" \l "ordering_operators)
         1. [The Algorithm](http://speakingjs.com/es5/ch09.html" \l "_the_algorithm)
      5. [The Plus Operator (+)](http://speakingjs.com/es5/ch09.html" \l "plus_operator)
         1. [The Algorithm](http://speakingjs.com/es5/ch09.html" \l "_the_algorithm_2)
      6. [Operators for Booleans and Numbers](http://speakingjs.com/es5/ch09.html" \l "_operators_for_booleans_and_numbers)
      7. [Special Operators](http://speakingjs.com/es5/ch09.html" \l "_special_operators)
         1. [The Conditional Operator ( ? : )](http://speakingjs.com/es5/ch09.html" \l "conditional_operator)
         2. [The Comma Operator](http://speakingjs.com/es5/ch09.html" \l "comma_operator)
         3. [The void Operator](http://speakingjs.com/es5/ch09.html" \l "void_operator)
      8. [Categorizing Values via typeof and instanceof](http://speakingjs.com/es5/ch09.html" \l "_categorizing_values_via_typeof_and_instanceof)
         1. [typeof: Categorizing Primitives](http://speakingjs.com/es5/ch09.html" \l "typeof)
         2. [instanceof: Checking Whether an Object Is an Instance of a Given Constructor](http://speakingjs.com/es5/ch09.html" \l "instanceof)
      9. [Object Operators](http://speakingjs.com/es5/ch09.html" \l "_object_operators)
   4. [10. Booleans](http://speakingjs.com/es5/ch10.html)
      1. [Converting to Boolean](http://speakingjs.com/es5/ch10.html" \l "toboolean)
         1. [Manually Converting to Boolean](http://speakingjs.com/es5/ch10.html" \l "_manually_converting_to_boolean)
         2. [Truthy and Falsy Values](http://speakingjs.com/es5/ch10.html" \l "truthy_falsy)
      2. [Logical Operators](http://speakingjs.com/es5/ch10.html" \l "_logical_operators)
         1. [Binary Logical Operators: And (&&) and Or (||)](http://speakingjs.com/es5/ch10.html" \l "binary_logical_operators)
         2. [Logical And (&&)](http://speakingjs.com/es5/ch10.html" \l "_logical_and_amp_amp)
         3. [Logical Or (||)](http://speakingjs.com/es5/ch10.html" \l "logical_or)
         4. [Logical Not (!)](http://speakingjs.com/es5/ch10.html" \l "logical_not)
      3. [Equality Operators, Ordering Operators](http://speakingjs.com/es5/ch10.html" \l "_equality_operators_ordering_operators)
      4. [The Function Boolean](http://speakingjs.com/es5/ch10.html" \l "function_boolean)
   5. [11. Numbers](http://speakingjs.com/es5/ch11.html)
      1. [Number Literals](http://speakingjs.com/es5/ch11.html" \l "_number_literals)
         1. [Exponent](http://speakingjs.com/es5/ch11.html" \l "_exponent)
         2. [Invoking Methods on Literals](http://speakingjs.com/es5/ch11.html" \l "_invoking_methods_on_literals)
      2. [Converting to Number](http://speakingjs.com/es5/ch11.html" \l "tonumber)
         1. [Manually Converting to Number](http://speakingjs.com/es5/ch11.html" \l "_manually_converting_to_number)
         2. [parseFloat()](http://speakingjs.com/es5/ch11.html" \l "parseFloat)
      3. [Special Number Values](http://speakingjs.com/es5/ch11.html" \l "_special_number_values)
         1. [NaN](http://speakingjs.com/es5/ch11.html" \l "nan)
         2. [Infinity](http://speakingjs.com/es5/ch11.html" \l "infinity)
         3. [Two Zeros](http://speakingjs.com/es5/ch11.html" \l "two_zeros)
      4. [The Internal Representation of Numbers](http://speakingjs.com/es5/ch11.html" \l "number_representation)
         1. [Special Exponents](http://speakingjs.com/es5/ch11.html" \l "_special_exponents)
      5. [Handling Rounding Errors](http://speakingjs.com/es5/ch11.html" \l "rounding_errors)
      6. [Integers in JavaScript](http://speakingjs.com/es5/ch11.html" \l "integers)
         1. [Ranges of Integers](http://speakingjs.com/es5/ch11.html" \l "_ranges_of_integers)
         2. [Representing Integers as Floating-Point Numbers](http://speakingjs.com/es5/ch11.html" \l "_representing_integers_as_floating_point_numbers)
         3. [Safe Integers](http://speakingjs.com/es5/ch11.html" \l "safe_integers)
      7. [Converting to Integer](http://speakingjs.com/es5/ch11.html" \l "converting_to_integer)
         1. [Integers via Math.floor(), Math.ceil(), and Math.round()](http://speakingjs.com/es5/ch11.html" \l "integers_via_math)
         2. [Integers via the Custom Function ToInteger()](http://speakingjs.com/es5/ch11.html" \l "ToInteger)
         3. [32-bit Integers via Bitwise Operators](http://speakingjs.com/es5/ch11.html" \l "integers_via_bitwise_operators)
         4. [Integers via parseInt()](http://speakingjs.com/es5/ch11.html" \l "parseInt)
      8. [Arithmetic Operators](http://speakingjs.com/es5/ch11.html" \l "arithmetic_operators)
      9. [Bitwise Operators](http://speakingjs.com/es5/ch11.html" \l "bitwise_operators)
         1. [Background Knowledge](http://speakingjs.com/es5/ch11.html" \l "_background_knowledge)
         2. [Bitwise Not Operator](http://speakingjs.com/es5/ch11.html" \l "_bitwise_not_operator)
         3. [Binary Bitwise Operators](http://speakingjs.com/es5/ch11.html" \l "binary_bitwise_operators)
         4. [Bitwise Shift Operators](http://speakingjs.com/es5/ch11.html" \l "_bitwise_shift_operators)
      10. [The Function Number](http://speakingjs.com/es5/ch11.html" \l "function_number)
      11. [Number Constructor Properties](http://speakingjs.com/es5/ch11.html" \l "_number_constructor_properties)
      12. [Number Prototype Methods](http://speakingjs.com/es5/ch11.html" \l "_number_prototype_methods)
          1. [Number.prototype.toFixed(fractionDigits?)](http://speakingjs.com/es5/ch11.html" \l "_number_prototype_tofixed_fractiondigits)
          2. [Number.prototype.toPrecision(precision?)](http://speakingjs.com/es5/ch11.html" \l "_number_prototype_toprecision_precision)
          3. [Number.prototype.toString(radix?)](http://speakingjs.com/es5/ch11.html" \l "Number.prototype.toString)
          4. [Number.prototype.toExponential(fractionDigits?)](http://speakingjs.com/es5/ch11.html" \l "_number_prototype_toexponential_fractiondigits)
      13. [Functions for Numbers](http://speakingjs.com/es5/ch11.html" \l "global_number_functions)
      14. [Sources for This Chapter](http://speakingjs.com/es5/ch11.html" \l "_sources_for_this_chapter)
   6. [12. Strings](http://speakingjs.com/es5/ch12.html)
      1. [String Literals](http://speakingjs.com/es5/ch12.html" \l "quoting_strings)
      2. [Escaping in String Literals](http://speakingjs.com/es5/ch12.html" \l "_escaping_in_string_literals)
      3. [Character Access](http://speakingjs.com/es5/ch12.html" \l "_character_access)
      4. [Converting to String](http://speakingjs.com/es5/ch12.html" \l "tostring)
         1. [Manually Converting to String](http://speakingjs.com/es5/ch12.html" \l "manual_tostring)
      5. [Comparing Strings](http://speakingjs.com/es5/ch12.html" \l "comparing_strings)
      6. [Concatenating Strings](http://speakingjs.com/es5/ch12.html" \l "_concatenating_strings)
         1. [Concatenation: The Plus (+) Operator](http://speakingjs.com/es5/ch12.html" \l "string_plus)
         2. [Concatenation: Joining an Array of String Fragments](http://speakingjs.com/es5/ch12.html" \l "_concatenation_joining_an_array_of_string_fragments)
      7. [The Function String](http://speakingjs.com/es5/ch12.html" \l "_the_function_string)
      8. [String Constructor Method](http://speakingjs.com/es5/ch12.html" \l "String.fromCharCode)
      9. [String Instance Property length](http://speakingjs.com/es5/ch12.html" \l "_string_instance_property_length)
      10. [String Prototype Methods](http://speakingjs.com/es5/ch12.html" \l "string_prototype_methods)
          1. [Extract Substrings](http://speakingjs.com/es5/ch12.html" \l "String.prototype.charCodeAt)
          2. [Transform](http://speakingjs.com/es5/ch12.html" \l "_transform)
          3. [Search and Compare](http://speakingjs.com/es5/ch12.html" \l "String.prototype.localeCompare)
          4. [Test, Match, and Replace with Regular Expressions](http://speakingjs.com/es5/ch12.html" \l "_test_match_and_replace_with_regular_expressions)
   7. [13. Statements](http://speakingjs.com/es5/ch13.html)
      1. [Declaring and Assigning Variables](http://speakingjs.com/es5/ch13.html" \l "_declaring_and_assigning_variables)
      2. [The Bodies of Loops and Conditionals](http://speakingjs.com/es5/ch13.html" \l "loops_conditionals_bodies)
      3. [Loops](http://speakingjs.com/es5/ch13.html" \l "_loops_2)
         1. [Mechanisms to Be Used with Loops](http://speakingjs.com/es5/ch13.html" \l "_mechanisms_to_be_used_with_loops)
         2. [while](http://speakingjs.com/es5/ch13.html" \l "_while)
         3. [do-while](http://speakingjs.com/es5/ch13.html" \l "_do_while)
         4. [for](http://speakingjs.com/es5/ch13.html" \l "for_loop)
         5. [for-in](http://speakingjs.com/es5/ch13.html" \l "for-in)
         6. [for each-in](http://speakingjs.com/es5/ch13.html" \l "_for_each_in)
      4. [Conditionals](http://speakingjs.com/es5/ch13.html" \l "_conditionals_2)
         1. [if-then-else](http://speakingjs.com/es5/ch13.html" \l "_if_then_else)
         2. [switch](http://speakingjs.com/es5/ch13.html" \l "_switch)
      5. [The with Statement](http://speakingjs.com/es5/ch13.html" \l "with_statement)
         1. [Syntax and Semantics](http://speakingjs.com/es5/ch13.html" \l "_syntax_and_semantics)
         2. [The with Statement Is Deprecated](http://speakingjs.com/es5/ch13.html" \l "_the_with_statement_is_deprecated)
         3. [The Rationale for the Deprecation](http://speakingjs.com/es5/ch13.html" \l "_the_rationale_for_the_deprecation)
      6. [The debugger Statement](http://speakingjs.com/es5/ch13.html" \l "_the_debugger_statement)
   8. [14. Exception Handling](http://speakingjs.com/es5/ch14.html)
      1. [What Is Exception Handling?](http://speakingjs.com/es5/ch14.html" \l "_what_is_exception_handling)
      2. [Exception Handling in JavaScript](http://speakingjs.com/es5/ch14.html" \l "_exception_handling_in_javascript)
         1. [throw](http://speakingjs.com/es5/ch14.html" \l "_throw)
         2. [try-catch-finally](http://speakingjs.com/es5/ch14.html" \l "_try_catch_finally)
         3. [Examples](http://speakingjs.com/es5/ch14.html" \l "_examples)
      3. [Error Constructors](http://speakingjs.com/es5/ch14.html" \l "error_constructors)
      4. [Stack Traces](http://speakingjs.com/es5/ch14.html" \l "_stack_traces)
      5. [Implementing Your Own Error Constructor](http://speakingjs.com/es5/ch14.html" \l "_implementing_your_own_error_constructor)
   9. [15. Functions](http://speakingjs.com/es5/ch15.html)
      1. [The Three Roles of Functions in JavaScript](http://speakingjs.com/es5/ch15.html" \l "_the_three_roles_of_functions_in_javascript)
      2. [Terminology: “Parameter” Versus “Argument”](http://speakingjs.com/es5/ch15.html" \l "_terminology_parameter_versus_argument)
      3. [Defining Functions](http://speakingjs.com/es5/ch15.html" \l "defining_functions)
         1. [Function Expressions](http://speakingjs.com/es5/ch15.html" \l "function_expressions)
         2. [Function Declarations](http://speakingjs.com/es5/ch15.html" \l "_function_declarations)
         3. [The Function Constructor](http://speakingjs.com/es5/ch15.html" \l "_the_function_constructor)
      4. [Hoisting](http://speakingjs.com/es5/ch15.html" \l "function_hoisting)
      5. [The Name of a Function](http://speakingjs.com/es5/ch15.html" \l "function_names)
      6. [Which Is Better: A Function Declaration or a Function Expression?](http://speakingjs.com/es5/ch15.html" \l "_which_is_better_a_function_declaration_or_a_function_expression)
      7. [More Control over Function Calls: call(), apply(), and bind()](http://speakingjs.com/es5/ch15.html" \l "_more_control_over_function_calls_call_apply_and_bind)
         1. [func.apply(thisValue, argArray)](http://speakingjs.com/es5/ch15.html" \l "functional_apply)
         2. [func.bind(thisValue, arg1, ..., argN)](http://speakingjs.com/es5/ch15.html" \l "_func_bind_thisvalue_arg1_argn)
      8. [Handling Missing or Extra Parameters](http://speakingjs.com/es5/ch15.html" \l "_handling_missing_or_extra_parameters)
         1. [All Parameters by Index: The Special Variable arguments](http://speakingjs.com/es5/ch15.html" \l "arguments_variable)
         2. [Mandatory Parameters, Enforcing a Minimum Arity](http://speakingjs.com/es5/ch15.html" \l "_mandatory_parameters_enforcing_a_minimum_arity)
         3. [Optional Parameters](http://speakingjs.com/es5/ch15.html" \l "optional_parameters)
         4. [Simulating Pass-by-Reference Parameters](http://speakingjs.com/es5/ch15.html" \l "_simulating_pass_by_reference_parameters)
         5. [Pitfall: Unexpected Optional Parameters](http://speakingjs.com/es5/ch15.html" \l "_pitfall_unexpected_optional_parameters)
      9. [Named Parameters](http://speakingjs.com/es5/ch15.html" \l "named_parameters)
         1. [Named Parameters as Descriptions](http://speakingjs.com/es5/ch15.html" \l "_named_parameters_as_descriptions)
         2. [Optional Named Parameters](http://speakingjs.com/es5/ch15.html" \l "_optional_named_parameters)
         3. [Simulating Named Parameters in JavaScript](http://speakingjs.com/es5/ch15.html" \l "_simulating_named_parameters_in_javascript)
   10. [16. Variables: Scopes, Environments, and Closures](http://speakingjs.com/es5/ch16.html)
       1. [Declaring a Variable](http://speakingjs.com/es5/ch16.html" \l "_declaring_a_variable)
       2. [Background: Static Versus Dynamic](http://speakingjs.com/es5/ch16.html" \l "_background_static_versus_dynamic)
       3. [Background: The Scope of a Variable](http://speakingjs.com/es5/ch16.html" \l "variable_scope)
       4. [Variables Are Function-Scoped](http://speakingjs.com/es5/ch16.html" \l "_variables_are_function_scoped_2)
       5. [Variable Declarations Are Hoisted](http://speakingjs.com/es5/ch16.html" \l "_variable_declarations_are_hoisted)
       6. [Introducing a New Scope via an IIFE](http://speakingjs.com/es5/ch16.html" \l "iife)
          1. [IIFE Variation: Prefix Operators](http://speakingjs.com/es5/ch16.html" \l "iife_prefix)
          2. [IIFE Variation: Already Inside Expression Context](http://speakingjs.com/es5/ch16.html" \l "_iife_variation_already_inside_expression_context)
          3. [IIFE Variation: An IIFE with Parameters](http://speakingjs.com/es5/ch16.html" \l "_iife_variation_an_iife_with_parameters)
          4. [IIFE Applications](http://speakingjs.com/es5/ch16.html" \l "_iife_applications)
       7. [Global Variables](http://speakingjs.com/es5/ch16.html" \l "_global_variables)
          1. [Best Practice: Avoid Creating Global Variables](http://speakingjs.com/es5/ch16.html" \l "avoid_global_variables)
          2. [Module Systems Lead to Fewer Globals](http://speakingjs.com/es5/ch16.html" \l "_module_systems_lead_to_fewer_globals)
       8. [The Global Object](http://speakingjs.com/es5/ch16.html" \l "global_object)
          1. [Cross-Platform Considerations](http://speakingjs.com/es5/ch16.html" \l "_cross_platform_considerations)
          2. [Use Cases for window](http://speakingjs.com/es5/ch16.html" \l "_use_cases_for_window)
       9. [Environments: Managing Variables](http://speakingjs.com/es5/ch16.html" \l "environments)
       10. [Closures: Functions Stay Connected to Their Birth Scopes](http://speakingjs.com/es5/ch16.html" \l "closures)
           1. [Handling Closures via Environments](http://speakingjs.com/es5/ch16.html" \l "_handling_closures_via_environments)
           2. [Pitfall: Inadvertently Sharing an Environment](http://speakingjs.com/es5/ch16.html" \l "inadvertently_sharing_environments)
   11. [17. Objects and Inheritance](http://speakingjs.com/es5/ch17.html)
       1. [Layer 1: Single Objects](http://speakingjs.com/es5/ch17.html" \l "single_objects)
          1. [Kinds of Properties](http://speakingjs.com/es5/ch17.html" \l "kinds_of_properties)
          2. [Object Literals](http://speakingjs.com/es5/ch17.html" \l "object_literals)
          3. [Dot Operator (.): Accessing Properties via Fixed Keys](http://speakingjs.com/es5/ch17.html" \l "dot_operator)
          4. [Unusual Property Keys](http://speakingjs.com/es5/ch17.html" \l "_unusual_property_keys)
          5. [Bracket Operator ([]): Accessing Properties via Computed Keys](http://speakingjs.com/es5/ch17.html" \l "bracket_operator)
       2. [Converting Any Value to an Object](http://speakingjs.com/es5/ch17.html" \l "toobject)
       3. [this as an Implicit Parameter of Functions and Methods](http://speakingjs.com/es5/ch17.html" \l "_this_as_an_implicit_parameter_of_functions_and_methods)
          1. [Calling Functions While Setting this: call(), apply(), and bind()](http://speakingjs.com/es5/ch17.html" \l "oop_call_apply_bind)
          2. [apply() for Constructors](http://speakingjs.com/es5/ch17.html" \l "apply_constructors)
          3. [Pitfall: Losing this When Extracting a Method](http://speakingjs.com/es5/ch17.html" \l "_pitfall_losing_this_when_extracting_a_method)
          4. [Pitfall: Functions Inside Methods Shadow this](http://speakingjs.com/es5/ch17.html" \l "_pitfall_functions_inside_methods_shadow_this)
       4. [Layer 2: The Prototype Relationship Between Objects](http://speakingjs.com/es5/ch17.html" \l "prototype_relationship)
          1. [Inheritance](http://speakingjs.com/es5/ch17.html" \l "_inheritance)
          2. [Overriding](http://speakingjs.com/es5/ch17.html" \l "_overriding)
          3. [Sharing Data Between Objects via a Prototype](http://speakingjs.com/es5/ch17.html" \l "sharing_via_prototype)
          4. [Getting and Setting the Prototype](http://speakingjs.com/es5/ch17.html" \l "get_set_prototype)
          5. [The Special Property \_\_proto\_\_](http://speakingjs.com/es5/ch17.html" \l "proto)
          6. [Setting and Deleting Affects Only Own Properties](http://speakingjs.com/es5/ch17.html" \l "setting_properties_proto_chain)
       5. [Iteration and Detection of Properties](http://speakingjs.com/es5/ch17.html" \l "iterate_and_detect_properties)
          1. [Listing Own Property Keys](http://speakingjs.com/es5/ch17.html" \l "Object.keys)
          2. [Listing All Property Keys](http://speakingjs.com/es5/ch17.html" \l "getAllPropertyNames)
          3. [Checking Whether a Property Exists](http://speakingjs.com/es5/ch17.html" \l "_checking_whether_a_property_exists)
          4. [Examples](http://speakingjs.com/es5/ch17.html" \l "_examples_2)
       6. [Best Practices: Iterating over Own Properties](http://speakingjs.com/es5/ch17.html" \l "object_iteration)
       7. [Accessors (Getters and Setters)](http://speakingjs.com/es5/ch17.html" \l "getters_setters)
          1. [Defining Accessors via an Object Literal](http://speakingjs.com/es5/ch17.html" \l "_defining_accessors_via_an_object_literal)
          2. [Defining Accessors via Property Descriptors](http://speakingjs.com/es5/ch17.html" \l "_defining_accessors_via_property_descriptors)
          3. [Accessors and Inheritance](http://speakingjs.com/es5/ch17.html" \l "_accessors_and_inheritance)
       8. [Property Attributes and Property Descriptors](http://speakingjs.com/es5/ch17.html" \l "property_attributes)
          1. [Property Attributes](http://speakingjs.com/es5/ch17.html" \l "_property_attributes)
          2. [Property Descriptors](http://speakingjs.com/es5/ch17.html" \l "property_descriptors)
          3. [Getting and Defining Properties via Descriptors](http://speakingjs.com/es5/ch17.html" \l "functions_for_property_descriptors)
          4. [Copying an Object](http://speakingjs.com/es5/ch17.html" \l "code_copyOwnPropertiesFrom)
          5. [Properties: Definition Versus Assignment](http://speakingjs.com/es5/ch17.html" \l "definition_vs_assignment)
          6. [Inherited Read-Only Properties Can’t Be Assigned To](http://speakingjs.com/es5/ch17.html" \l "_inherited_read_only_properties_can_t_be_assigned_to)
          7. [Enumerability: Best Practices](http://speakingjs.com/es5/ch17.html" \l "enumerability_best_practices)
       9. [Protecting Objects](http://speakingjs.com/es5/ch17.html" \l "protecting_objects)
          1. [Preventing Extensions](http://speakingjs.com/es5/ch17.html" \l "_preventing_extensions)
          2. [Sealing](http://speakingjs.com/es5/ch17.html" \l "_sealing)
          3. [Freezing](http://speakingjs.com/es5/ch17.html" \l "freezing_objects)
          4. [Pitfall: Protection Is Shallow](http://speakingjs.com/es5/ch17.html" \l "_pitfall_protection_is_shallow)
       10. [Layer 3: Constructors—Factories for Instances](http://speakingjs.com/es5/ch17.html" \l "constructors)
           1. [The new Operator Implemented in JavaScript](http://speakingjs.com/es5/ch17.html" \l "_the_new_operator_implemented_in_javascript)
           2. [Terminology: The Two Prototypes](http://speakingjs.com/es5/ch17.html" \l "two_prototypes)
           3. [The constructor Property of Instances](http://speakingjs.com/es5/ch17.html" \l "constructor_property)
           4. [The instanceof Operator](http://speakingjs.com/es5/ch17.html" \l "operator_instanceof)
           5. [Tips for Implementing Constructors](http://speakingjs.com/es5/ch17.html" \l "constructor_tips)
       11. [Data in Prototype Properties](http://speakingjs.com/es5/ch17.html" \l "_data_in_prototype_properties)
           1. [Avoid Prototype Properties with Initial Values for Instance Properties](http://speakingjs.com/es5/ch17.html" \l "_avoid_prototype_properties_with_initial_values_for_instance_properties)
           2. [Avoid Nonpolymorphic Prototype Properties](http://speakingjs.com/es5/ch17.html" \l "_avoid_nonpolymorphic_prototype_properties)
           3. [Polymorphic Prototype Properties](http://speakingjs.com/es5/ch17.html" \l "_polymorphic_prototype_properties)
       12. [Keeping Data Private](http://speakingjs.com/es5/ch17.html" \l "private_data_for_objects)
           1. [Private Data in the Environment of a Constructor (Crockford Privacy Pattern)](http://speakingjs.com/es5/ch17.html" \l "private_data_constructor_environment)
           2. [Private Data in Properties with Marked Keys](http://speakingjs.com/es5/ch17.html" \l "private_data_marked_keys)
           3. [Private Data in Properties with Reified Keys](http://speakingjs.com/es5/ch17.html" \l "private_data_reified_keys)
           4. [Keeping Global Data Private via IIFEs](http://speakingjs.com/es5/ch17.html" \l "iife_for_objects)
       13. [Layer 4: Inheritance Between Constructors](http://speakingjs.com/es5/ch17.html" \l "constructor_inheritance)
           1. [Inheriting Instance Properties](http://speakingjs.com/es5/ch17.html" \l "_inheriting_instance_properties)
           2. [Inheriting Prototype Properties](http://speakingjs.com/es5/ch17.html" \l "_inheriting_prototype_properties)
           3. [Ensuring That instanceof Works](http://speakingjs.com/es5/ch17.html" \l "_ensuring_that_instanceof_works)
           4. [Overriding a Method](http://speakingjs.com/es5/ch17.html" \l "_overriding_a_method)
           5. [Making a Supercall](http://speakingjs.com/es5/ch17.html" \l "_making_a_supercall)
           6. [Avoiding Hardcoding the Name of the Superconstructor](http://speakingjs.com/es5/ch17.html" \l "_avoiding_hardcoding_the_name_of_the_superconstructor)
           7. [Example: Constructor Inheritance in Use](http://speakingjs.com/es5/ch17.html" \l "_example_constructor_inheritance_in_use)
           8. [Example: The Inheritance Hierarchy of Built-in Constructors](http://speakingjs.com/es5/ch17.html" \l "_example_the_inheritance_hierarchy_of_built_in_constructors)
           9. [Antipattern: The Prototype Is an Instance of the Superconstructor](http://speakingjs.com/es5/ch17.html" \l "_antipattern_the_prototype_is_an_instance_of_the_superconstructor)
       14. [Methods of All Objects](http://speakingjs.com/es5/ch17.html" \l "methods_of_all_objects)
           1. [Conversion to Primitive](http://speakingjs.com/es5/ch17.html" \l "Object.prototype.valueOf)
           2. [Object.prototype.toLocaleString()](http://speakingjs.com/es5/ch17.html" \l "_object_prototype_tolocalestring)
           3. [Prototypal Inheritance and Properties](http://speakingjs.com/es5/ch17.html" \l "_prototypal_inheritance_and_properties)
       15. [Generic Methods: Borrowing Methods from Prototypes](http://speakingjs.com/es5/ch17.html" \l "generic_method)
           1. [Accessing Object.prototype and Array.prototype via Literals](http://speakingjs.com/es5/ch17.html" \l "_accessing_object_prototype_and_array_prototype_via_literals)
           2. [Examples of Calling Methods Generically](http://speakingjs.com/es5/ch17.html" \l "_examples_of_calling_methods_generically)
           3. [Array-Like Objects and Generic Methods](http://speakingjs.com/es5/ch17.html" \l "array-like_objects)
           4. [A List of All Generic Methods](http://speakingjs.com/es5/ch17.html" \l "list_of_generic_methods)
       16. [Pitfalls: Using an Object as a Map](http://speakingjs.com/es5/ch17.html" \l "_pitfalls_using_an_object_as_a_map)
           1. [Pitfall 1: Inheritance Affects Reading Properties](http://speakingjs.com/es5/ch17.html" \l "_pitfall_1_inheritance_affects_reading_properties)
           2. [Pitfall 2: Overriding Affects Invoking Methods](http://speakingjs.com/es5/ch17.html" \l "_pitfall_2_overriding_affects_invoking_methods)
           3. [Pitfall 3: The Special Property \_\_proto\_\_](http://speakingjs.com/es5/ch17.html" \l "_pitfall_3_the_special_property___proto)
           4. [The dict Pattern: Objects Without Prototypes Are Better Maps](http://speakingjs.com/es5/ch17.html" \l "dict_pattern)
           5. [Best Practices](http://speakingjs.com/es5/ch17.html" \l "_best_practices)
       17. [Cheat Sheet: Working with Objects](http://speakingjs.com/es5/ch17.html" \l "oop_cheat_sheet)
   12. [18. Arrays](http://speakingjs.com/es5/ch18.html)
       1. [Overview](http://speakingjs.com/es5/ch18.html" \l "_overview)
          1. [Arrays Are Maps, Not Tuples](http://speakingjs.com/es5/ch18.html" \l "_arrays_are_maps_not_tuples)
          2. [Arrays Can Also Have Properties](http://speakingjs.com/es5/ch18.html" \l "_arrays_can_also_have_properties)
       2. [Creating Arrays](http://speakingjs.com/es5/ch18.html" \l "creating_arrays)
          1. [The Array Constructor](http://speakingjs.com/es5/ch18.html" \l "array_constructor)
          2. [Multidimensional Arrays](http://speakingjs.com/es5/ch18.html" \l "_multidimensional_arrays)
       3. [Array Indices](http://speakingjs.com/es5/ch18.html" \l "array_indices)
          1. [The in Operator and Indices](http://speakingjs.com/es5/ch18.html" \l "_the_in_operator_and_indices)
          2. [Deleting Array Elements](http://speakingjs.com/es5/ch18.html" \l "_deleting_array_elements)
          3. [Array Indices in Detail](http://speakingjs.com/es5/ch18.html" \l "_array_indices_in_detail)
       4. [length](http://speakingjs.com/es5/ch18.html" \l "array_length)
          1. [Manually Increasing the Length of an Array](http://speakingjs.com/es5/ch18.html" \l "_manually_increasing_the_length_of_an_array)
          2. [Decreasing the Length of an Array](http://speakingjs.com/es5/ch18.html" \l "_decreasing_the_length_of_an_array)
          3. [The Maximum Length](http://speakingjs.com/es5/ch18.html" \l "_the_maximum_length)
       5. [Holes in Arrays](http://speakingjs.com/es5/ch18.html" \l "array_holes)
          1. [Creating Holes](http://speakingjs.com/es5/ch18.html" \l "_creating_holes)
          2. [Sparse Arrays Versus Dense Arrays](http://speakingjs.com/es5/ch18.html" \l "_sparse_arrays_versus_dense_arrays)
          3. [Which Operations Ignore Holes, and Which Consider Them?](http://speakingjs.com/es5/ch18.html" \l "_which_operations_ignore_holes_and_which_consider_them)
          4. [Removing Holes from Arrays](http://speakingjs.com/es5/ch18.html" \l "_removing_holes_from_arrays)
       6. [Array Constructor Method](http://speakingjs.com/es5/ch18.html" \l "_array_constructor_method)
       7. [Array Prototype Methods](http://speakingjs.com/es5/ch18.html" \l "array_prototype_methods)
       8. [Adding and Removing Elements (Destructive)](http://speakingjs.com/es5/ch18.html" \l "Array.prototype.push)
       9. [Sorting and Reversing Elements (Destructive)](http://speakingjs.com/es5/ch18.html" \l "Array.prototype.sort)
          1. [Comparing Numbers](http://speakingjs.com/es5/ch18.html" \l "_comparing_numbers)
          2. [Comparing Strings](http://speakingjs.com/es5/ch18.html" \l "_comparing_strings)
          3. [Comparing Objects](http://speakingjs.com/es5/ch18.html" \l "_comparing_objects)
       10. [Concatenating, Slicing, Joining (Nondestructive)](http://speakingjs.com/es5/ch18.html" \l "Array.prototype.slice)
       11. [Searching for Values (Nondestructive)](http://speakingjs.com/es5/ch18.html" \l "_searching_for_values_nondestructive)
       12. [Iteration (Nondestructive)](http://speakingjs.com/es5/ch18.html" \l "array_iteration_methods)
           1. [Examination Methods](http://speakingjs.com/es5/ch18.html" \l "Array_prototype_forEach)
           2. [Transformation Methods](http://speakingjs.com/es5/ch18.html" \l "Array.prototype.map)
           3. [Reduction Methods](http://speakingjs.com/es5/ch18.html" \l "_reduction_methods)
       13. [Pitfall: Array-Like Objects](http://speakingjs.com/es5/ch18.html" \l "_pitfall_array_like_objects)
       14. [Best Practices: Iterating over Arrays](http://speakingjs.com/es5/ch18.html" \l "array_iteration)
   13. [19. Regular Expressions](http://speakingjs.com/es5/ch19.html)
       1. [Regular Expression Syntax](http://speakingjs.com/es5/ch19.html" \l "_regular_expression_syntax)
          1. [Atoms: General](http://speakingjs.com/es5/ch19.html" \l "regex_atoms_general)
          2. [Atoms: Character Classes](http://speakingjs.com/es5/ch19.html" \l "_atoms_character_classes)
          3. [Atoms: Groups](http://speakingjs.com/es5/ch19.html" \l "_atoms_groups)
          4. [Quantifiers](http://speakingjs.com/es5/ch19.html" \l "regexp_quantifiers)
          5. [Assertions](http://speakingjs.com/es5/ch19.html" \l "regexp_assertions)
          6. [Disjunction](http://speakingjs.com/es5/ch19.html" \l "_disjunction)
       2. [Unicode and Regular Expressions](http://speakingjs.com/es5/ch19.html" \l "_unicode_and_regular_expressions)
       3. [Creating a Regular Expression](http://speakingjs.com/es5/ch19.html" \l "creating_regexps)
          1. [Literal Versus Constructor](http://speakingjs.com/es5/ch19.html" \l "_literal_versus_constructor)
          2. [Flags](http://speakingjs.com/es5/ch19.html" \l "regexp_flags)
          3. [Instance Properties of Regular Expressions](http://speakingjs.com/es5/ch19.html" \l "_instance_properties_of_regular_expressions)
          4. [Examples of Creating Regular Expressions](http://speakingjs.com/es5/ch19.html" \l "_examples_of_creating_regular_expressions)
       4. [RegExp.prototype.test: Is There a Match?](http://speakingjs.com/es5/ch19.html" \l "RegExp.prototype.test)
       5. [String.prototype.search: At What Index Is There a Match?](http://speakingjs.com/es5/ch19.html" \l "String.prototype.search)
       6. [RegExp.prototype.exec: Capture Groups](http://speakingjs.com/es5/ch19.html" \l "RegExp.prototype.exec)
          1. [First Match (Flag /g Not Set)](http://speakingjs.com/es5/ch19.html" \l "_first_match_flag_g_not_set)
          2. [All Matches (Flag /g Set)](http://speakingjs.com/es5/ch19.html" \l "_all_matches_flag_g_set)
       7. [String.prototype.match: Capture Groups or Return All Matching Substrings](http://speakingjs.com/es5/ch19.html" \l "String.prototype.match)
       8. [String.prototype.replace: Search and Replace](http://speakingjs.com/es5/ch19.html" \l "String.prototype.replace)
          1. [Replacement Is a String](http://speakingjs.com/es5/ch19.html" \l "_replacement_is_a_string)
          2. [Replacement Is a Function](http://speakingjs.com/es5/ch19.html" \l "_replacement_is_a_function)
       9. [Problems with the Flag /g](http://speakingjs.com/es5/ch19.html" \l "tips_flag_g)
       10. [Tips and Tricks](http://speakingjs.com/es5/ch19.html" \l "_tips_and_tricks)
           1. [Quoting Text](http://speakingjs.com/es5/ch19.html" \l "_quoting_text)
           2. [Pitfall: Without an Assertion (e.g., ^, $), a Regular Expression Is Found Anywhere](http://speakingjs.com/es5/ch19.html" \l "_pitfall_without_an_assertion_e_g_a_regular_expression_is_found_anywhere)
           3. [Matching Everything or Nothing](http://speakingjs.com/es5/ch19.html" \l "_matching_everything_or_nothing)
           4. [Manually Implementing Lookbehind](http://speakingjs.com/es5/ch19.html" \l "regexp-look-behind)
       11. [Regular Expression Cheat Sheet](http://speakingjs.com/es5/ch19.html" \l "_regular_expression_cheat_sheet)
   14. [20. Dates](http://speakingjs.com/es5/ch20.html)
       1. [The Date Constructor](http://speakingjs.com/es5/ch20.html" \l "date_constructors)
       2. [Date Constructor Methods](http://speakingjs.com/es5/ch20.html" \l "_date_constructor_methods)
       3. [Date Prototype Methods](http://speakingjs.com/es5/ch20.html" \l "date_prototype_methods)
          1. [Time Unit Getters and Setters](http://speakingjs.com/es5/ch20.html" \l "_time_unit_getters_and_setters)
          2. [Various Getters and Setters](http://speakingjs.com/es5/ch20.html" \l "_various_getters_and_setters)
          3. [Convert a Date to a String](http://speakingjs.com/es5/ch20.html" \l "_convert_a_date_to_a_string)
       4. [Date Time Formats](http://speakingjs.com/es5/ch20.html" \l "_date_time_formats)
          1. [Date Formats (No Time)](http://speakingjs.com/es5/ch20.html" \l "_date_formats_no_time)
          2. [Time Formats (No Date)](http://speakingjs.com/es5/ch20.html" \l "_time_formats_no_date)
          3. [Date Time Formats](http://speakingjs.com/es5/ch20.html" \l "date_time_formats)
       5. [Time Values: Dates as Milliseconds Since 1970-01-01](http://speakingjs.com/es5/ch20.html" \l "date_milliseconds)
          1. [Converting a Date to a Number](http://speakingjs.com/es5/ch20.html" \l "_converting_a_date_to_a_number)
   15. [21. Math](http://speakingjs.com/es5/ch21.html)
       1. [Math Properties](http://speakingjs.com/es5/ch21.html" \l "_math_properties)
       2. [Numerical Functions](http://speakingjs.com/es5/ch21.html" \l "Math.pow)
       3. [Trigonometric Functions](http://speakingjs.com/es5/ch21.html" \l "Math.atan2)
       4. [Other Functions](http://speakingjs.com/es5/ch21.html" \l "Math_max)
   16. [22. JSON](http://speakingjs.com/es5/ch22.html)
       1. [Background](http://speakingjs.com/es5/ch22.html" \l "_background_2)
          1. [Data Format](http://speakingjs.com/es5/ch22.html" \l "_data_format)
          2. [History](http://speakingjs.com/es5/ch22.html" \l "_history)
          3. [Grammar](http://speakingjs.com/es5/ch22.html" \l "_grammar)
       2. [JSON.stringify(value, replacer?, space?)](http://speakingjs.com/es5/ch22.html" \l "JSON.stringify)
          1. [Data Ignored by JSON.stringify()](http://speakingjs.com/es5/ch22.html" \l "_data_ignored_by_json_stringify)
          2. [The toJSON() Method](http://speakingjs.com/es5/ch22.html" \l "_the_tojson_method)
       3. [JSON.parse(text, reviver?)](http://speakingjs.com/es5/ch22.html" \l "JSON.parse)
       4. [Transforming Data via Node Visitors](http://speakingjs.com/es5/ch22.html" \l "node_visitors)
          1. [JSON.stringify()](http://speakingjs.com/es5/ch22.html" \l "_json_stringify)
          2. [JSON.parse()](http://speakingjs.com/es5/ch22.html" \l "_json_parse)
   17. [23. Standard Global Variables](http://speakingjs.com/es5/ch23.html)
       1. [Constructors](http://speakingjs.com/es5/ch23.html" \l "_constructors)
       2. [Error Constructors](http://speakingjs.com/es5/ch23.html" \l "_error_constructors)
       3. [Nonconstructor Functions](http://speakingjs.com/es5/ch23.html" \l "_nonconstructor_functions)
          1. [Encoding and Decoding Text](http://speakingjs.com/es5/ch23.html" \l "_encoding_and_decoding_text)
          2. [Categorizing and Parsing Numbers](http://speakingjs.com/es5/ch23.html" \l "_categorizing_and_parsing_numbers)
       4. [Dynamically Evaluating JavaScript Code via eval() and new Function()](http://speakingjs.com/es5/ch23.html" \l "_dynamically_evaluating_javascript_code_via_eval_and_new_function)
          1. [Evaluating Code Using eval()](http://speakingjs.com/es5/ch23.html" \l "eval)
          2. [Evaluating Code Using new Function()](http://speakingjs.com/es5/ch23.html" \l "function_constructor)
          3. [eval() Versus new Function()](http://speakingjs.com/es5/ch23.html" \l "eval_vs_new_function)
          4. [Best Practices](http://speakingjs.com/es5/ch23.html" \l "_best_practices_2)
          5. [Conclusion](http://speakingjs.com/es5/ch23.html" \l "_conclusion_2)
       5. [The Console API](http://speakingjs.com/es5/ch23.html" \l "console_api)
          1. [How Standardized Is the Console API Across Engines?](http://speakingjs.com/es5/ch23.html" \l "_how_standardized_is_the_console_api_across_engines)
          2. [Simple Logging](http://speakingjs.com/es5/ch23.html" \l "_simple_logging)
          3. [Checking and Counting](http://speakingjs.com/es5/ch23.html" \l "_checking_and_counting)
          4. [Formatted Logging](http://speakingjs.com/es5/ch23.html" \l "_formatted_logging)
          5. [Profiling and Timing](http://speakingjs.com/es5/ch23.html" \l "_profiling_and_timing)
       6. [Namespaces and Special Values](http://speakingjs.com/es5/ch23.html" \l "_namespaces_and_special_values)
   18. [24. Unicode and JavaScript](http://speakingjs.com/es5/ch24.html)
       1. [Unicode History](http://speakingjs.com/es5/ch24.html" \l "_unicode_history)
       2. [Important Unicode Concepts](http://speakingjs.com/es5/ch24.html" \l "_important_unicode_concepts)
       3. [Code Points](http://speakingjs.com/es5/ch24.html" \l "_code_points)
       4. [Unicode Encodings](http://speakingjs.com/es5/ch24.html" \l "_unicode_encodings)
       5. [JavaScript Source Code and Unicode](http://speakingjs.com/es5/ch24.html" \l "_javascript_source_code_and_unicode)
          1. [Source Code Internally](http://speakingjs.com/es5/ch24.html" \l "_source_code_internally)
          2. [Source Code Externally](http://speakingjs.com/es5/ch24.html" \l "_source_code_externally)
       6. [JavaScript Strings and Unicode](http://speakingjs.com/es5/ch24.html" \l "_javascript_strings_and_unicode)
          1. [Escape Sequences](http://speakingjs.com/es5/ch24.html" \l "_escape_sequences)
          2. [Refering to Astral Plane Characters via Escapes](http://speakingjs.com/es5/ch24.html" \l "_refering_to_astral_plane_characters_via_escapes)
          3. [Counting Characters](http://speakingjs.com/es5/ch24.html" \l "_counting_characters)
          4. [Unicode Normalization](http://speakingjs.com/es5/ch24.html" \l "_unicode_normalization)
       7. [JavaScript Regular Expressions and Unicode](http://speakingjs.com/es5/ch24.html" \l "_javascript_regular_expressions_and_unicode)
          1. [Matching Any Code Unit and Any Code Point](http://speakingjs.com/es5/ch24.html" \l "_matching_any_code_unit_and_any_code_point)
          2. [Libraries](http://speakingjs.com/es5/ch24.html" \l "_libraries)
          3. [Recommended Reading and Chapter Sources](http://speakingjs.com/es5/ch24.html" \l "_recommended_reading_and_chapter_sources)
   19. [25. New in ECMAScript 5](http://speakingjs.com/es5/ch25.html)
       1. [New Features](http://speakingjs.com/es5/ch25.html" \l "_new_features)
       2. [Syntactic Changes](http://speakingjs.com/es5/ch25.html" \l "_syntactic_changes)
       3. [New Functionality in the Standard Library](http://speakingjs.com/es5/ch25.html" \l "_new_functionality_in_the_standard_library)
          1. [Metaprogramming](http://speakingjs.com/es5/ch25.html" \l "_metaprogramming)
          2. [New Methods](http://speakingjs.com/es5/ch25.html" \l "_new_methods)
          3. [JSON](http://speakingjs.com/es5/ch25.html" \l "_json)
       4. [Tips for Working with Legacy Browsers](http://speakingjs.com/es5/ch25.html" \l "_tips_for_working_with_legacy_browsers)
6. [IV. Tips, Tools, and Libraries](http://speakingjs.com/es5/pt04.html)
   1. [26. A Meta Code Style Guide](http://speakingjs.com/es5/ch26.html)
      1. [Existing Style Guides](http://speakingjs.com/es5/ch26.html" \l "_existing_style_guides)
      2. [General Tips](http://speakingjs.com/es5/ch26.html" \l "_general_tips)
         1. [Code Should Be Consistent](http://speakingjs.com/es5/ch26.html" \l "_code_should_be_consistent)
         2. [Code Should Be Easy to Understand](http://speakingjs.com/es5/ch26.html" \l "_code_should_be_easy_to_understand)
      3. [Commonly Accepted Best Practices](http://speakingjs.com/es5/ch26.html" \l "_commonly_accepted_best_practices)
         1. [Brace Styles](http://speakingjs.com/es5/ch26.html" \l "_brace_styles)
         2. [Prefer Literals to Constructors](http://speakingjs.com/es5/ch26.html" \l "_prefer_literals_to_constructors)
         3. [Don’t Be Clever](http://speakingjs.com/es5/ch26.html" \l "_don_t_be_clever)
         4. [Acceptable Cleverness](http://speakingjs.com/es5/ch26.html" \l "_acceptable_cleverness)
      4. [Controversial Rules](http://speakingjs.com/es5/ch26.html" \l "_controversial_rules)
         1. [Syntax](http://speakingjs.com/es5/ch26.html" \l "style_one_var_per_line)
         2. [Variables](http://speakingjs.com/es5/ch26.html" \l "_variables)
         3. [Object Orientation](http://speakingjs.com/es5/ch26.html" \l "_object_orientation)
         4. [Miscellaneous](http://speakingjs.com/es5/ch26.html" \l "_miscellaneous)
      5. [Conclusion](http://speakingjs.com/es5/ch26.html" \l "_conclusion_3)
   2. [27. Language Mechanisms for Debugging](http://speakingjs.com/es5/ch27.html)
   3. [28. Subclassing Built-ins](http://speakingjs.com/es5/ch28.html)
      1. [Terminology](http://speakingjs.com/es5/ch28.html" \l "_terminology)
      2. [Obstacle 1: Instances with Internal Properties](http://speakingjs.com/es5/ch28.html" \l "_obstacle_1_instances_with_internal_properties)
         1. [Workaround for Obstacle 1](http://speakingjs.com/es5/ch28.html" \l "_workaround_for_obstacle_1)
         2. [Caveats](http://speakingjs.com/es5/ch28.html" \l "_caveats)
      3. [Obstacle 2: A Constructor That Can’t Be Called as a Function](http://speakingjs.com/es5/ch28.html" \l "subclassing_error)
         1. [Workaround for Obstacle 2](http://speakingjs.com/es5/ch28.html" \l "_workaround_for_obstacle_2)
      4. [Another Solution: Delegation](http://speakingjs.com/es5/ch28.html" \l "_another_solution_delegation)
   4. [29. JSDoc: Generating API Documentation](http://speakingjs.com/es5/ch29.html)
      1. [The Basics of JSDoc](http://speakingjs.com/es5/ch29.html" \l "_the_basics_of_jsdoc)
         1. [Syntax](http://speakingjs.com/es5/ch29.html" \l "_syntax_2)
         2. [Naming Types](http://speakingjs.com/es5/ch29.html" \l "_naming_types)
      2. [Basic Tags](http://speakingjs.com/es5/ch29.html" \l "_basic_tags)
      3. [Documenting Functions and Methods](http://speakingjs.com/es5/ch29.html" \l "_documenting_functions_and_methods)
      4. [Inline Type Information (“Inline Doc Comments”)](http://speakingjs.com/es5/ch29.html" \l "_inline_type_information_inline_doc_comments)
      5. [Documenting Variables, Parameters, and Instance Properties](http://speakingjs.com/es5/ch29.html" \l "_documenting_variables_parameters_and_instance_properties)
      6. [Documenting Classes](http://speakingjs.com/es5/ch29.html" \l "_documenting_classes)
         1. [Defining a Class via a Constructor Function](http://speakingjs.com/es5/ch29.html" \l "_defining_a_class_via_a_constructor_function)
         2. [Defining a Class via an Object Literal](http://speakingjs.com/es5/ch29.html" \l "_defining_a_class_via_an_object_literal)
         3. [Defining a Class via an Object Literal with an @constructs Method](http://speakingjs.com/es5/ch29.html" \l "_defining_a_class_via_an_object_literal_with_an_constructs_method)
         4. [Subclassing](http://speakingjs.com/es5/ch29.html" \l "_subclassing)
      7. [Other Useful Tags](http://speakingjs.com/es5/ch29.html" \l "_other_useful_tags)
   5. [30. Libraries](http://speakingjs.com/es5/ch30.html)
      1. [Shims Versus Polyfills](http://speakingjs.com/es5/ch30.html" \l "shim_vs_polyfill)
      2. [Four Language Libraries](http://speakingjs.com/es5/ch30.html" \l "_four_language_libraries)
      3. [The ECMAScript Internationalization API](http://speakingjs.com/es5/ch30.html" \l "i18n_api)
         1. [The ECMAScript Internationalization API, Edition 1](http://speakingjs.com/es5/ch30.html" \l "_the_ecmascript_internationalization_api_edition_1)
         2. [What Kind of Standard Is It?](http://speakingjs.com/es5/ch30.html" \l "_what_kind_of_standard_is_it)
         3. [When Can I Use It?](http://speakingjs.com/es5/ch30.html" \l "_when_can_i_use_it)
         4. [Further Reading](http://speakingjs.com/es5/ch30.html" \l "_further_reading)
      4. [Directories for JavaScript Resources](http://speakingjs.com/es5/ch30.html" \l "_directories_for_javascript_resources)
   6. [31. Module Systems and Package Managers](http://speakingjs.com/es5/ch31.html)
      1. [Module Systems](http://speakingjs.com/es5/ch31.html" \l "module_systems)
      2. [Package Managers](http://speakingjs.com/es5/ch31.html" \l "_package_managers)
      3. [Quick and Dirty Modules](http://speakingjs.com/es5/ch31.html" \l "_quick_and_dirty_modules)
   7. [32. More Tools](http://speakingjs.com/es5/ch32.html)
   8. [33. What to Do Next](http://speakingjs.com/es5/ch33.html)
7. [Index](http://speakingjs.com/es5/ix01.html)
8. [About the Author](http://speakingjs.com/es5/author_bios.html)
9. [Colophon](http://speakingjs.com/es5/co02.html)
10. [Copyright](http://speakingjs.com/es5/copyright.html)