| **Term** | **Definition** | **Slide** |
| --- | --- | --- |
| JavaScript | The original web-based programming language, initially run only in browsers to make webpages come alive, but now used also in other environments thanks to platforms like NodeJS | F5 |
| engine | A tool that converts (compiles) JavaScript to machine-executable binaries, allowing JS to run in various environments. Browsers and platforms such as NodeJS have different engines. | F7 |
| statement | A single JS command, terminated by either a semi-colon or a newline | F8 |
| comment | Human-readable text, not compiled by the JS engine, indicated by either // for a single line comment or /\* multi-line comment \*/ | F8 |
| variable | Named storage for a piece of data, created using either the var, let or const keyword | F10 |
| number | Data type that includes both integers and decimal (floating-point) numbers | F11 |
| string | Data type that stores a sequence of characters, delimited with quote marks | F14 |
| backtick | A backwards facing apostrophe (usually to the left of the number 1 on a standard keyboard) used to delimit strings containing JS expressions | F14 |
| boolean | A data type that has one of two possible values (usually denoted true and false) | F15 |
| null | A special value indicating that a variable contains no value (empty) | F16 |
| undefined | A special value indicating that a variable contains an unknown value, or has not yet been assigned a value | F17 |
| object | A complex data type that can store multiple values as a series of key-value pairs | F18 |
| array | A special type of object containing a series of ordered values, using numeric indexes (keys) beginning at 0 | F18 |
| typeof | Special operator used to obtain the type of a given variable | F20 |
| concatenate | Joining two variables (at least one of which is a string) together to make a single long string combining both | F22 |
| function | A named block of code that can be executed and re-used, which may accept some variables (parameters) passed to it when executed (called) | F29 |
| DRY | Don’t Repeat Yourself: a pattern or strategy for writing efficient code which aims to re-use and not repeat code | F29 |
| return | The last statement inside a function before the function ends. Used to specify a value that should be returned to the code from where the function was called | F31 |
| Function declaration | Standard/original syntax for creating a function, which uses the keyword ‘function’ followed by the name of the function, then the parameters, then the function body | F29 |
| Function expression | Syntax for creating a function that mirrors a variable expression. Uses the keyword const followed by the function name, then an equals sign followed by parameters and function body. | F32 |
| hoisting | Accessing a function or variable before it is defined. Only works with var and function declarations, not with let/const variables. | F33 |
| Arrow function | Newer, more concise syntax for creating a function that uses an arrow => to indicate the function body, often used when passing functions as arguments. | F34 |
| object | A complex variable that stores multiple pieces of data as key-value pairs using curly bracket notation | F36 |
| iterate | The process of repeatedly accessing each variable or value within a variable and advancing to the next value | F42 |
| reference | A location in memory that stores a value. Object variables store the reference (or memory address) not the value itself. | F44 |
| Shallow copy | A method of creating a copy (clone) of an object that duplicates the top-level properties into a new variable with its own reference | F45 |
| Deep copy | Similar to shallow copy, except that it duplicates all properties at potentially deeply nested levels within the object | F47 |
| method | A function that belongs to an object or class | F48 |
| context | Determined at run-time - the specific instance of an object used when generically accessing its properties or methods | F49 |
| this | A special keyword used to access the context within a function or object | F50 |
| constructor function | A special type of function (now legacy) used to construct or create potentially many objects that all follow the same pattern | F51 |
| new | Special keyword used to create a new object based on a constructor function or class template | F51 |
| class | Modern method of setting up a template to create multiple objects with the same properties and methods | F53 |
| primitive | A variable that contains a single value, eg. string, number, boolean, which is stored by value in memory | I4 |
| hexadecimal | A number system using base 16, which includes the 10 digits 0-9 as well as the characters A-F to represent numbers, often used for colours and complex character encoding | I9 |
| octal | A number system using base 8, which includes the 8 digits 0-7 to represent numbers, rarely used | I10 |
| binary | A number system using base 2, which includes the 2 digits 0-1 to represent numbers, rarely used | I10 |
| Infinity | Special numeric value representing infinity, which is any number divided by zero, or very large numbers that overflow the 64bit memory storage allocation | I12 |
| NaN | Stands for Not a Number - special value returned when attempting to do mathematical operations on values that are not numbers | I14 |
| float | A floating-point (decimal) number | I15 |
| Escape sequence | Uses a backslash character inside any string to escape the subsequent characters, used to print special values such as newlines, tabs, quotes, emojis, non-latin characters, etc | I17 |
| indexOf | String function that returns the index of the given character in a string (or -1 if not found) | I19 |
| substring | String function that returns a section of a string beginning and ending at the given index values | I19 |
| toUpperCase | String function that returns the original string with every character converted to upper case (also toLowerCase equivalent) | I19 |
| startsWith | String function that returns true if the original string starts with the given string and false otherwise (also endsWith equivalent) | I20 |
| split | String function that returns an array of values from the original string, separated using the given character | I20 |
| slice | String function that returns a portion of the original string beginning and ending with the given index values (see substring) | I20 |
| replace | String function that returns a new version of the original string with the first occurrence of the given characters replaced by the given replacement string | I20 |
| replaceAll | String function that returns a new version of the original string with all occurrences of the given characters replaced by the given replacement string | I20 |
| trim | String function that returns a new version of the original string with all leading and trailing white space characters removed | I20 |
| queue | First-in-first-out (FIFO) method of storing data in an array | I22 |
| stack | Last-in-first-out (LIFO) method of storing data in an array | I22 |
| push | Array function that modifies the original array by adding the given value to the end | I23 |
| pop | Array function that modifies the original array by removing the last item and returning it | I23 |
| shift | Array function that modifies the original array by removing the first item and returning it | I24 |
| unshift | Array function that modifies the original array by adding the given value to the beginning | I24 |
| matrix | A multidimensional array | I26 |
| toString | Prototype function that returns the given variable in string format | I27 |
| splice | Array function that modifies the original array by inserting the given values into the original at the specified index, potentially removing the original element/s at that index and returning them | I28 |
| slice | Array function that returns a new array containing elements from the original array between the given index values | I29 |
| concat | Array function that combines multiple given values into the original array and returns a new array including everything | I30 |
| forEach | Array function that takes a function as a parameter and runs it once for each of the original array elements | I31 |
| indexOf | Array function that returns the first index of the given element in the original array, or -1 if not found | I32 |
| lastIndexOf | Array function that returns the last index of the given element in the original array, or -1 if not found | I33 |
| find | Array function that takes a function as a parameter and runs it once for each of the original array elements - if this function returns true, find will return this matching element | I34 |
| filter | Array function that takes a function as a parameter and runs it once for each of the original array elements - if this function returns true, filter will include this matching element in its returned array of matching elements | I35 |
| map | Array function that takes a function as a parameter and runs it once for each of the original array elements, returning a new array with the same change having been made to each element | I36 |
| sort | Array function that modifies the original array by changing the order of its elements, sorting them alphabetically by default. Can take a function parameter which allows a custom sort order to be specified | I37 |
| reverse | Array function that modifies the original array by reversing the order of its elements | I39 |
| join | Array function that joins all elements together into a string, separated by the given character/s | I40 |
| reduce | Array function that accumulates all elements together into a single value, eg. for calculating a total | I41 |
| iterable | Special objects containing multiple elements, which can be iterated over using a for … of loop | I43 |
| Map | Special iterable data type that stores a collection of key-value pairs, where both keys and values can be of any data type | I45 |
| Set | Special iterable data type that stores a collection of unique elements of any type | I50 |
| WeakMap | Special type of Map where all keys are weak references to objects, allowing entries to be garbage collected | I53 |
| destructuring | Syntax using the … operator that unpacks elements of an iterable variable into individual values/variables | I57 |
| rest | Alternative use for the … operator, which can be used to combine and store ‘the rest’ of multiple values/parameters into a single object/array | I60 |
| epoch | Special baseline time in programming - January 1 1970 UTC | I64 |
| UTC | Universal Time Coordinated (previously known as GMT) - baseline time zone | I64 |
| Date | Special JS data type for storing and managing date and time | I64 |
| toLocaleString | Date function to format the internal date and time according to the current locale and timezone | I68 |
| toLocaleDateString | Date function to format the internal date according to the current locale and timezone | I68 |
| toLocaleTimeString | Date function to format the internal time according to the current locale and timezone | I68 |
| JSON | Javascript Object Notation - a popular format for storing object data as strings in order to store in a file and/or send data between applications | I69 |
| stringify | JSON function to convert a variable to valid JSON string format | I69 |
| parse | JSON function to convert a JSON string back into a variable | I74 |
| replacer | Optional argument passed to stringify or parse to allow customisation of the stringify/parse process | I72 |
| scope | Section of code (context) which can access variables - usually either global or local (within curly braces of a function/block) | A4 |
| Lexical environment | Block of code contained within curly braces, within which both locally and globally scoped variables can be accessed | A4 |
| closure | A function that returns an inner function which, when executed, can access variables in both the inner and outer function scopes | A6 |
| Function object | The ability to treat functions as objects in JS, by passing them by reference, accessing built-in properties such as name and length, and adding our own custom properties | A8 |
| setTimeout | Built-in JS function that delays the execution of a given function for a specified period of time, executing it asynchronously, returning a unique reference to the timeout | A11 |
| clearTimeout | Built-in JS function that uses the timeout reference from setTimeout to cancel the delayed code and prevent it from ever executing | A12 |
| synchronous | Standard code execution, where each statement executes in turn, one after the other until the script is complete | A13 |
| asynchronous | Code (often long-running or of unknown duration) that does not need to wait until completion before other statements can be executed in the meantime | A13 |
| setInterval | Built-in JS function that repeats execution of a given function with a given fixed amount of time between each execution, and returns a unique reference to the interval | A14 |
| clearInterval | Built-in JS function that uses the interval reference from setInterval to stop the code from repeatedly executing | A14 |
| decorator | Wrapper function that returns an altered version of a given function after adding some extra functionality | A17 |
| call | Built-in function for all JS functions that allows it to be called with a given context and list of parameters | A21 |
| apply | Built-in function for all JS functions that allows it to be applied to a given context and array of parameters | A21 |
| bind | Built-in function for all JS functions that allows a given context to be bound into the function, so that internal references to ‘this’ refer to that given context | A28 |
| prototype | Parent object containing various properties and methods that is inherited (either implicitly or explicitly) by one or more children | A29 |
| getPrototypeOf | Static method of Object that accesses the prototype for a given object | A30 |
| setPrototypeOf | Static method of Object that sets the prototype for a given object to the given parent object | A31 |
| enumerable | Option to control whether a custom object property is able to be iterated over, or included in a for…in loop | A31 |
| own | Used to distinguish which properties belong to the object itself (‘own’ properties) and which are inherited from its prototype | A31 |
| polyfilling | A piece of code used to provide modern JS functionality on older browsers that do not natively support it. | A34 |
| object-oriented | A programming paradigm that organises code into objects, each representing an instance of a class with its own data and behaviour. | A37 |
| functional | A programming paradigm that organises code into functions, emphasising immutability and avoiding side effects. | A37 |
| extends | Keyword used to specify that one class should inherit properties and methods from a parent | A38 |
| super | Keyword used to refer to a parent class within child methods | A39 |
| constructor | Special function of a class called when creating a new instance of the class, used to set up the new instance with required property values | A40 |
| static | Keyword used to specify a class property or method as belonging to the class itself, and not to each instance of the class | A42 |
| public | Default visibility of class properties and methods - accessible to any code whether internal or external to the class | A43 |
| protected | Naming convention using an underscore to prefix any property or method of a class that is designed to be used only internally within the class or its children, not externally from an instance | A43 |
| private | Syntax using a # to prefix any property or method of a class that will be enforced as only visible within the class itself, not from any children or externally from an instance | A44 |
| try | Keyword used before an enclosed block of code that can be used to capture any errors caused by the code block to enable them to be handled | 46 |
| catch | Keyword used to specify the second half of a try block, used to catch any errors that occurred and handle them without the code crashing | A46 |
| throw | Keyword used to deliberately trigger (throw) an error | A48 |
| finally | Keyword used for the final section of a try…catch block, to define any code that needs to be executed in the event of either success (no errors) or failure (catching an error) | A49 |
| promise | Special modern JS object used to handle asynchronous code by making a request and responding only when it resolves, without halting the script execution while waiting | A50 |
| resolve | Function executed when a promise has successfully completed, used to pass back any resulting data | A50 |
| reject | Function executed when a promise has completed but failed, used to pass back any resulting error data | A50 |
| then | Function of a consumed promise, executed when it resolves | A51 |
| fetch | Built-in browser JS function for sending a HTTP request to a server and asynchronously processing the results via a promise | A53 |
| chaining | Using the result of one function to call another function immediately | A54 |
| async | Keyword used to indicate that a function returns a promise, and contains asynchronous code that needs to wait until promise resolution before a result is returned | A57 |
| await | Keyword used inside an async function to make asynchronous, promise-based code behave synchronously by waiting, or halting the script, until the result is available | A57 |