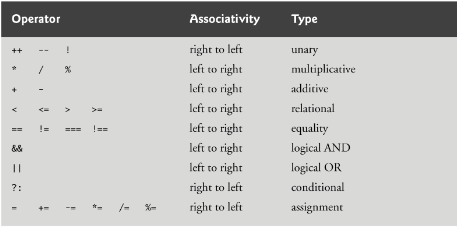
Exam 2

<https://js.do/> 🡨 run code questions

Module 8

* Counter – controlled repletion requires
  + Name of a control variable
  + Initial value of the control variable
  + The increment (or decrement) by which the control variable is modified each time through the loop
  + The condition that tests for the final value of the control variable to determine whether looping should continue
* Double-quote character delimits the beginning and the end of a string literal in JavaScript
  + Cannot be used in a string unless it is preceded by a \ to create the escape sequence \”
* HTML5 allows either single quotes or double quotes to be placed around the value specified for an attribute
* JavaScript allows single quotes to be placed in a string literal
* For Repetition statement
* For statement
  + Specifies each of the items needed for counter-controlled repetition with a control variable
  + Can use a block to put multiple statements into the body
* If the loop’s condition uses a < or > instead of a <= or >=, or vice-versa, it can result in an off-by-one error
* For statement header contains three expressions
  + Initialization
  + Condition
  + Increment Expression
* The increment expression is the for statement acts like a stand-alone statement at the end of the body of the for statement
* Place only expressions involving the control variable in the initialization and increment sections of a for statement
* The three expressions in the for statement are optional
* The two semicolons in the for statement are required
* The initialization, loop-continuation condition and increment portions of a for statement can contain arithmetic expressions
* The part of a script in which a variable name can be used in known as the variable’s scope
* The increment of a for statement may be negative, in which case it is called a decrement and the loop actually counts downward
* If the loop-continuation condition initially is false, the body of the for statement is not performed
  + Execution proceeds with the statement following the for statement
* Error-prevention tip
  + Although the value of the control variable can be changed in the body of a for statement, avoid changing it, because doing so can lead to subtle errors
* Common Programming Error
  + Not using the proper relation operator in the loop-continuation condition of a loop that counts downward ( using I <= 1 instead of I >= 1 in a loop that counts down to 1) is a logical error.
* Good programing practice
  + Although statements preceding a for statement and in the body of a for statement can often be merged into the for header, avoid doings so, because it makes the program more difficult to read
* JavaScript does not include an exponentiation operator
  + Math object’s pow method for this purpose. Math.pow (x, y) calculates the value of x raised to the yth power
* Switch multiple-selection statement
  + Tests a variable or expression separately for each of the values it may assume
  + Different actions are taken for each value
* CSS property list-style-type
  + Allows you to set the numbering system for a list
  + Possible values include
    - Decimal (numbers- the default)
    - Lower-roman (lowercase roman numerals)
    - Upper-roman (uppercase roman numerals)
    - Lower-alpha (lowercase letters)
    - Upper-alpha (uppercase letters)
    - Others
* Switch Statement
  + Consists of a series of case labels and an optional default case
  + When control reaches a switch statement
    - The script evaluates the controlling expression in the parentheses
    - Compares this value with the value in each of the case labels
    - If the comparison evaluates to true, the statements after the case label are executed in order until a break statement is reached
* The break statement is used as the last statement in each case to exit the switch statement immediately
* The default case allows you to specify a set of statements to execute if no other case is satisfied
  + Usually the last case in the switch statement
* Each case can have multiple actions (statements)
* Braces are not required around the multiple actions in case of a switch
* The break statement is not required for the last case because program control automatically continues with the next statement after the switch
* Having several case labels listed together (e.g., case 1: case 2: with no statements between the cases) executes the same set of actions for each case
* Do… while statement
  + Tests the loop-continuation condition after the loop body executes
  + The loop body always executes at least once
* Common programming error
  + Infinite loops are caused when the loop-continuation condition never becomes false in a while, for or do… while statement. To prevent this, make sure that there’s not a semicolon immediately after the header of a while or for statement. In a counter-controlled loop, make sure that the control variable is incremented (or decremented) in the body of the loop. In a sentinel-controlled loop, the sentinel value should eventually be input
* Break statement in a while, for, do… wile or switch statement
  + Causes immediate exit from the statement
  + Execution continues with the next statement in sequence
* Break statement common uses
  + Early escape from a loop
  + Skip the remainder of a switch statement
* Continue statement in a while, for or do…while
  + Skips the remaining statements in the body of the statement and proceeds with the next iteration of the loop
  + In while and do… while statements, the loop-continuation test evaluates immediately after the continue statements executes
  + In for statements, the increment expression executes, then the loop-continuation test evaluates
* Logical operators can be used to form complex conditions by combining simple conditions
  + && (logical AND)
  + || (logical OR)
  + ! (logical NOT, also called logical negation)
* The && operator is used to ensure that two conditions are both true before choosing a certain path of execution
* JavaScript evaluates to false or true all expressions that include relational operators, equality operators and/or logical operators
* The || (logical OR) operator is used to ensure that either or both of two conditions are true before choosing a certain path of execution
* The && operator has a higher precedence than the || operator
* Both operators associate from left to right
* An expression containing && or || operators is evaluated only until truth or falsity is known
  + This is called short-circuit evaluation
* ! (logical negation) operator
  + Reverses the meaning of a condition (i.e., a true value becomes false, and a false value becomes true)
  + Has only a single condition as an operand (ie., it is a unary operator)
  + Placed before a condition to evaluate to true if the original condition (without the logical negation operator) is false
* Most nonboolean values can be converted to a Boolean true or false value
* Nonzero numeric values are considered to be true
* The numeric value zerio is considered to be false
* Any string that contains characters is considered to be true
* The empty string is considered to be false
* The value null and variables that have been declared but not initialized are considered to be false
* All objects are considered to be true
* k

Module 9

* To develop and maintain a large program
  + Construct it from small, simple pieces
  + Divide and conquer
* You’ll combine new functions that you write with prepackaged functions and objects available in JavaScript
* The prepackaged functions that belong to JavaScript objects are called methods (things like Math.pow)
* JavaScript provides several objects that have a rich collection of methods for performing a common mathematical calculation, string manipulation, date and time manipulation, and manipulation of collections of data called arrays.
* You can define programmer-defined functions that perform specific tasks and use them at many points in a script
  + The actual statements defining the function are written only once and are hidden from other functions
* Functions are invoked by writing the name of the function, followed by a left parenthesis, followed by a comma-separated list of zero or more arguments, followed by a right parenthesis
* Methods are called in the same way as functions, but require the name of the objects to which the method belongs and a dot preceding the method name
* Function (and method) arguments may be constants, variables or expressions
* Return statement
  + Passes information from inside a function back to the point in the program where it was called
  + A function must be called explicitly for the code in its body to execute
  + The format of a function definition is function *function-name( parameter-list)*
  + { Declarations and statements }
* Common programming error
  + Forgetting to return a value from a function that’s supposed to return a value is a logic error
* Three ways to return control to the point at which a function was invoked
  + Reaching the function-ending right brace
  + Executing that statement return;
  + Executing the statement “return expression;” to return the value of expression to the caller
* When a return statement executes, control returns immediately to the point at which the function was invoked
* The script in our next example uses a programmer-defined function called maximum to determine the largest of the three floating-point values
* All variables declared with the keyword var in functions definitions are local variables- this means that they can be accessed only in the function in which they’re defined
* A function’s parameters are also considered to be local variables
* There are several reasons for modularizing a program with functions
  + Divide and conquer approach makes program development more manageable
  + Software reusability
  + Avoid repeated code in a program
* Software engineering observation
  + If a function’s task cannot be expressed concisely, perhaps the function is performing too many different tasks. It’s usually best to break such a function into several smaller functions.
  + To promote software reusability, every function should be limited to performing a single, well-defined task, and the name of the function should describe that task effectively. Such functions makes programs easier to write, debug, maintain, and modify.
  + Variables declared inside the body of a function are known only in that function. If the same variable names are used elsewhere in the program, they’ll be entirely separate variables in memory.
* Common Programming Error
  + Redefining a function parameter as a local variable in the function is a logic error
* Good Programming Practice
  + Do not use the same name for an argument passed to a function and the corresponding parameter in the function definition. Using different names avoids ambiguity
* Random method generates a floating-point value from 0 up to but no including 1
* Random integers in a certain range can be generated by scaling and shifting the values returned y random, then using math.floor to convert them to integers
  + The scaling factor determines the size of the range (i.e. a scaling factor of 4 means four possible integers)
  + The shift number is added to the result to determine where the range begins (i.e. shifting the numbers by 3 would give numbers between 3 and 7)
* Method math.floor rounds its argument down to the closet integer
* Error-prevention tip
  + Initializing variables when they’re declared in functions helps avoid incorrect results and interpreter messages warning of uninitialized data
* JavaScript provides nine global functions as part of a Global object
* This contains
  + All the global variables in the script
  + All the user-defined functions in the script
  + All the built-in global functions listed in the following slide
* You do not need to use the global object directly; JavaScript uses it for you
* Global Function
  + isFinite Takes a number argument and returns true if the value of the argument is NaN, Number, Positive\_Infinity or number, Negative\_Infinity (values that are not numbers or numbers outside the range that JavaScript supports)- otherwise the function returns false
  + isNaN takes a numeric argument and returns true if the value of the argument is not a number, otherwise it returns false. The function is commonly used with the return value of parseInt or parseFloast to determine whether the result is a proper numeric value
  + parseFloat takes a string argument and attempts to convert the beginning of the string into a floating-point value. If the conversion is unsuccessful the function returns NaN; otherwise it returns the converted value
  + parseInt takes a string argument and attempts to convert the beginning of the string into an integer value. If the conversion is unsuccessful, the function returns NaN; otherwise it returns the converted value.
* A recursive function calls itself, either directly, or indirectly through another function
* A recursive function knows how to solve only the simplest case, or base case
  + If the function is called with a base case, it returns a result
  + If the function is called with a more complex problem, it divides the problem into two conceptual pieces- a piece that the function knows how to process (the base case) and a simpler or smaller version of the original problem
* The function invokes (calls) a fresh copy of itself to go to work on the smaller problem; this invocation is referred to as a recursive call, or the recursion step
* The recursion step executes while the original call to the function is still open
* For recursion eventually to terminate, each time the function calls itself with a simpler version of the original problem, the sequence of smaller and smaller problems must converge on the base case
  + At that point the function recognizes the base case, returns a result to the previous copy of the function, and a sequence of returns ensues up the line until the original function call eventually returns the final result to the caller
* Common Programming Error
  + Omitting the base case and writing the recursion step incorrectly so that it does not converge on the base case are both erros that cause infinite recursion, eventually exhausting memory. This situation is analogous to the problem of an infinite loop in an iterative (non-recursive) solution.

Module 10

* Arrays
  + Data Structures consisting of related data items
* JavaScript arrays
  + “dynamic” entities that can change size after they are created
* An array is a group of memory locations
  + All have the same name and normally are of the required same type (although this attribute is not required in JavaScript)
* Each individual location is called an element
* We may refer to any one of these elements by giving the array’s name followed by the position number of the element in square brackets ([])
* The first element in every array is the zeroth element
* The ith element of array c is referred to as c[i-1]
* Array names follow the same conventions as other identifiers
* A subscripted array name
  + Can be used on the left side of an assignment to place a new value into an array element
  + Can be used on the right side of an assignment operation to use its value
* Each array in JavaScript knows its own length, which it stores in its length attribute and can be found with expression arrayname.length
* Javacript arrays are array objects
* You use the new operator to create an array and to specify the number of elements in an array
* The new operator creates an object as the script executes by obtaining enough memory to store an object of the type specified to the right of new
* Zero-based counting is usually used to iterate through arrays
* JavaScript reallocates an array when a value is assigned to an element that is outside the bounts of the original array
* Software engineering observation
  + It’s considered good practice to separate your javascript scripts into separate files so that they can be reused in multiple web pages
  + Javscript automatically reallocates an array when a value is assigned to an element that’s outside the bounds of the array. Elements between the last element of the original array and the new element are undefined
  + Error-prevention tip
    - When accessing array elements, the index values should never go below 0 and should be les than the number of elements in the array unless its your explicit intent to grow the array by assigning a value to a nonexistent element
* Arrays can be created using a comma-separated initializer list enclosed in square brackets
  + The array’s size is determined by the number of values in the initializer list
* The initial values of an array can be specified as arguments in the parentheses following new array
  + The size of the array is determined by the number of values in the parenthesis
* Error prevention tip
  + When iterating over all the elements of an array, use a for … in statement to ensure that you manipulate only the existing elements. The for… in statement skips any undefined elements in the array
* Sorting data
  + Putting data in a particular order, such as ascending or descending
  + One of the most important computing functions
* Array object in JavaScript has a built-in method sort
  + With no arguments, the method uses string comparisons to determine the sorting order of the array elements
  + Method sort takes as its argument the name of a function that compares its two arguments is less than the second argument
  + Zero if the arguments are qual, or
  + A positive value if the first argument is greater than the second
* Software engineering observation
  + Functions in javascript are considered to be data
  + Therefore functions can be assigned to variables, stored in arrays and passed to functions just like the other data types
* It’’s often necessary to determine whether an array contains a value that matches a certain key value
* The process of locating a particular element value in an array is called searching
* The array object in Javascript has built-in methods indexOf and lastindexOf for searching arrays
  + Method indexOf searches for the first occurrence of the specified key value
  + Method lastIndexOf searches for the last occurrence of the specified key value
* If they key value is found in the array each method returns the index of that value; otherwise, -1 is returned
* Every input element has a value property that can be used to get or set the element’s value
* You can pass an optional second argument to methods indexof and lastindexof that represents the index from which to start the search
* By default, this argument’s value is 0 and the methods search the entire array
* If the argument is greater than or equal to the array’s length, the methods simply return -1
* If the argument’s value is negative, it’s used as an offset from the end of the array
* To identify a particular two-dimensional multidimensional array element
  + Specify the two indices
  + By convention, the first identifies the element’s row, and the second identifies the element’s column
  + In general, an array with m rows and n columns is called an m-by-n array
  + Two dimensional array element accessed using an element name of the form a [row] [column]
    - A is the name of the array
    - Row and column are the indices that uniquely identify the row and column
* Multidimensional arrays can be initialized in declarations like a one-dimensional array, with values grouped by row in square brackets
  + The interpreter determines the number of rows by counting the number of sub initializer
  + The interpreter determines the number of columns in each row by counting the number of values in the sub-array that initializes the row
* The rows of a two-dimensional array can vary in length
* A multidimensional array in which each row has a different number of columns can be allocated dynamically with operator new

Module 11

* WE use HTML5’s new web storage capabilities to create a web application that stores a user’s favorite twitter searches on the computer for easy access at a later time
* We also provide a brief introduction to JSON, a means for creating javascript objects typically for transferring data over the internet between client-side and server-side programs
* Math object methods enable you to conveniently perform many common mathematical calculations
* An objects methods are called by writing the name of the object followed by a dot operator and the name of the method
* In parentheses following the method name is are arguments to the method
* Software engineering observation
  + The difference between invoking a stand0alone function and invoking a method of an a object is that an object name and a dot are not required to call a stand-alone function
* Math Object Methods
  + Abs(x)
    - Absolute value of x
  + Ceil(x)
    - Rounds x to the smallest integer not less than x
  + Cos(x)
    - Trigonometric cosine of x
  + Exp(x)
    - Exponential method x
  + Floor(x)
    - Rounds x to the largest integer not greater than x
  + Log(x)
    - Natural logarithm of x
  + Max (x,y)
    - Larger value of x and y
  + Min(x,y)
    - Smaller value of x and y
  + Pow(x,y)
    - X raised to the power y
  + Round (x)
    - Rounds x to the closest integer
  + Sin(x)
    - Trigonometric sine of x
  + Sqrt(x)
    - Square root of x
  + Tan(x)
    - Trigonometric tangent of x
* Properties of the math object
  + Math.e
    - Base of a natural logarithm
  + Math.LN2
    - Natural logarithm of 2
  + Math.LN10
    - Natural logarithm of 10
  + Math.LOG2E
    - Base 2 logarithm of e
  + Math.LOG10E
    - Base 10 logarithm of e
  + Math.PI
    - PI the ratio of a circle’s circumference to its diameter
  + Math.SQRT1\_2
    - Square root of .5
  + Math.SQRT2
    - Square root of 2.0
* Characters are the building blocks of JavaScript programs
* Every program is composed of a sequence of characters grouped together meaningfully that is interpreted by the computer as a series of instructions used to accomplish a task
* A string is a series of characters treated as a single unit
* A string may include letters, digits and various special characters
* Javascript supports Unicode, which represents a large portion of the world’s languages
* String literals or string constants are written as a sequence of characters in double or single quotation marks
* Combining strings is called concatenation
* Some String-Object methods
  + Char(At) index)
    - Returns a string containing the character at the specified index,.. If there’s no character at the index charAt returns an empty strig. The first character is located at index 0
  + CharCodeAT(index)
    - Returns the Unicode value of the character at the specified index or NaN if there’s no character at that index
  + Concat(string)
    - Concatenates its arguments to the end of the string on which the method is invoked. The original string is not modified; instead a new string is returned. This method is the same as adding two strings with the string-concatenation operator
  + FromCharCode(value 1, Value2, …)
    - Converts a list of Unicode values into a string containing the corresponding characters
  + IndexOf(substring, Index)
    - Searches for the first occurrence of substring starting from position index is the string that invokes the method. The method returns the starting index of substring in the source string of -1 if substring is not found. If the index argument is not provided, the method begins searching from index 0 is the source string
  + Lastindexof( substring, index)
    - Searches for the last occurrence of substring starting from position index and searching toward the beginning of the string that invokes the method. The method returns the starting index of substring in the source string or -1 if substring is not found. If the index argument is not provided the method begins searching from the end of the source string
  + Replace(searchString, replaceString)
    - Searches for the substring searchstring, replaces the first occurrence with replacestring and returns the modified string or returns the original string if no replacement was made
  + Slice(start, end )
    - returns a string containing the portion of the string from index start through index end, if the end index is not specified the method returns a string from the start index to the end source string. A negative end index specifies an offset from end of the string, starting from a position one past the end of the last character
  + Split (string)
    - splits the source string into an array of strings, where its string arguments specifies the delimiter
  + Substr( start, length)
    - returns a string containing length characters starting from index start in the source string. IF length is not specified, a string containing characters from start to the end of the source string is returned
  + Substring( start, end)
    - returns a string containing the characters from index strat up to but not including index end in the source string
  + toLowerCase()
    - returns a string in which all uppercase letters are not converted to lowercase letters. Non-letter characters are not change
  + toUpperCase()
    - returns a string in which all lowercase letters are converted to uppercase letters. Non letter characters are not changed .
* Date object provides methods for date and time manipulations
  + Based either on the computer’s local time zone or on world time standard’s coordinated universal time
* Most methods have a local time zone and a UTC version
* Empty parentheses after an object name indicate a call to the object’s constructor with no arguments
  + A constructor is an initializer method for an object
  + Called automatically when an object is allocated with new
  + The date constructor with no argument initializes the date object with the local computer’s current date and time
* Date method parse
  + Receives as its argument a string representing a date and time and returns the number of milliseconds between midnight, January 1, 1970, and the specified date and time
* Date method UTC
  + Returns the number of milliseconds between midnight, January 1, 1970, and the date and time specified as its arguments
  + Arguments include the required year, month and date, and the optional hours, minutes, seconds and milliseconds
  + If an argument is not specified, a 0 is supplied in its place
  + For hours, minutes and seconds, if the argument to the right of any of these arguments is specified that argument must also be specified
* Common programming error
  + Assuming that months are represented as numbers from 1 to 12 leads to off-by-one errors when you’re processing dates
* The Boolean and number objects are object wrappers for Boolean true/false values and numbers, respectively
* When a Boolean value is required in a javascript program, javascript automatically creates a Boolean object to store the value
* Javascript programmers can create Boolean object explicitly
  + Var b = new voolean (Booleanvalue);
  + If Booleanvalue is false, 0, null, number.NaN or the empty string (“”\_, or if no argument is supplied, the new Boolean objects contains false
  + Otherwise, the new Boolean object contains true
* Boolean-object methods
  + toString ()
    - returns the string “true” if the value of the Boolean object is true; otherwise return the string as false
  + Valueof() reutrns the value true if the Boolean object is truel; otherwise, returns false
* Javascript automatically creates number objects to store numeric values in a script
* You can create a number object with the statement
  + Var n = new number (numericvalue); numeric value is the number to store in the object
* Although you can explicitly create number objects, normally they are created when needed by the javascript interpreter
* Document object
  + Provided by the browser and allows javascript code to manipulate the current document in the browser
* JSON( Javascript Object Notation)
  + A simple way to represent javascript objects as strings
  + Introduced as an alternative to XML as a data-exchange technique
* JSON has gained acclaim due to its simple format, making objects easy to read, create and parse
* Each JSON object is represented as a list of property names and values contained in curly braces, in the following format
  + {propertyname1 : value1, propertyname2 : value 2 }
* Arrays are represented in JSON with square brackets in the following format:
  + [value0, value1, value2]
* Each value can be a string, a number, a JSON object, true, false or null
* JSON provides a straightforward way to manipulate objects in javascript, and many other programming languages now support this format
* In addition to simplifying object creation, JSON allows programs to easily extract data and efficiently transmit across the internet

Module 12

* jQuery is a JavaScript library
* jQuery greatly simplifies JavaScript programming
* Developed by John Resig at Rochester institute of Technology
* jQuery is a lightweight Javascript library that emphasizes interaction between JavaScript and HTMl
* jQuery is free, open source software
* jQuery is a lightweight “write less, do more” JavaScript library
* The purpose of jQuery is to maek it much easier to use JavaScript on your website
* jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code
* jQuery also simplifies a lot of the complicate things from JavaScript, like AJAX calls and Dom manipulation
* The jQuery library contains the following geatures:
  + HTML/DOM manipulation
  + HTML event methods
  + Effects and animations
  + AJAX
  + Utilities
* What jQuery does
  + Unobtrusive JavaScript – separation of behavior from structure
  + CSS – separation of style from structure
  + Allows adding JavaSccript to your web pages
  + Advantages over just JavaScript
    - Much easier to use
    - Eliminate cross-browser problems
  + HTML to CSS to DHTML
* jQuery Provides
  + Select DOM (Document Object Model) elements on a page – one element or a group of them
  + Set properties of DOM elements, in groups
  + Creates, deletes, shows, hides DOM elements
  + Defines event behavior on a page (click, mouse movement, dynamic styles, animations, dynamic content)
* The DOM
  + Document Object Model
  + jQuery is “DOM scripting”
  + hierarchal structure of a web page
  + you can add and subtract DOM elements on the fly
  + You can change the properties and contents of DOM elements on the fly
  + The DOM is a cross-platform and language-independent convention for representing and interacting with objects in HTML, XHTML, and XML documents.
  + Aspects of the DOM ( such as its elements) may be addressed and manipulated within the syntax of the programming language in use
* jQuery Syntax
  + The jQuery syntax is tailor-made for selecting HTML elements and performing some action the elements
  + Basic syntax is: $(selector)action()
  + A (selector) to “query (or find) HTML elements A jQuery (action) to be performed on the element(s)
  + Examples:
    - $(this).hide() – hides the current element
    - $(“p”).hide() – hides all the <p> elements
* jQuery Selectors
  + jQuery selectors allow you to select and manipulate HTML elements
  + jQuery selectors are used to find or select HTML elements based on their name, id, classes, types, attributes, values of attributes, and much more
  + It’s based on the existing CSS Selectors and in addition, it has some own custom selectors
  + All selectors in jQuery start with the dollar sign and parentheses
* The element Selector
  + The jQuery element selector selects elements based on the element name
  + You can select all <p> elements on a page like this:
    - $(“p”)
* Functions in a separate file
  + If your website contains a lot of pages, and you want your jQuery functions to be easy to maintain, you can put your jQuery functions in a separate .js file.
  + When we demonstrate jQuery in this tutorial, the functions are added directly into the <head> section
  + However sometimes it is preferable to place them in a separate file, like this (use the src attribute to refer to the .js file):
* jQuery events
  + jQuery is tailor-made to respond to events in a HTML page
  + All the different visitors’ actions that a web page can respond to are called events
  + An event represents the precise moment when something happens
  + In jQuery, most DOM events have an equivalent jQuery method
  + To assign a click event to all paragraphs on a page, you can do this:
    - $(“p”).click();
* Commonly used jQuery Event Methods
  + focus()Thefocus()method attaches an event handler function to an HTML form field.The function is executed when the form field gets focus:
  + hover()Thehover()method takes two functions and is a combination of themouseenter()andmouseleave()methods.The first function is executed when the mouse enters the HTML element, and the second function is executed when the mouse leaves the HTML element:
  + mouseleave()Themouseleave()method attaches an event handler function to an HTML element.The function is executed when the mouse pointer leaves the HTML element:Example
  + Themouseenter()method attaches an event handler function to an HTML element.The function is executed when the mouse pointer enters the HTML element:
  + click()
    - Theclick()method attaches an event handler function to an HTML element.
    - The function is executed when the user clicks on the HTML element.

Module 13

* Bootstrap
  + Is a front-end open source framework that helps you build mobile responsive websites more quickly and easily
  + It is also completely free
  + Using bootstrap, you can build customized complex web pages
  + Bootstrap provides a number of jQuery plugins for additional functionality
  + In order to use bootstrap, you first need to integrate it into your web page
  + For that, you have two different possibilities: load it remotely through CDN or download and use bootstrap locally
  + For our purpose, we will use a CDN to access it rather than install it locally