JavaScript Fundamental 01

- JavaScript Basics
    - 01 Getting Ready
        - code editor
        - browser
        - "legacy" Internet Explorer
        - place the script element at the end of body
        - embedding code vs. js files
    - 02 Data and Variables part 1
        - String
            - string literal notation double quote or single quote
            - var s = "hello, world!";
            - var s; // undefined declaration
            - s = "hello, world!"; // initialization
        - variable name
            - store value in memory
            - camel casing
            - hoisting
        - Number
            - var ten = 10;
            - var pi = 3.14;
        - Boolean
            - var isBoolean = false;
    - 03 Data and Variables part 2
        - comments
            - single line comment //
            - block comment /* */
        - mathematical operators
            - addition operator
            - subtraction operator
            - multiplication operator
            - division operator
            - order of operation / parentheses
        - assignment operator
        - concatenation operator
            - var foo = 4 + 5 + "7" + 6 + 5; // 9765
        - convert a number to string

        - var foo = 6 + "";
    - convert string to number
        - parseInt()
        - parseFloat()
    - JavaScript engine execute one statement faster than multi-statement.
        - var sum = 4 + 5,
        -     diff = 4 - 5;
- 04 Functions
    - function declaration
    - function expression anonymous function assigned to variable
      function are values/data
    - hoisting
    - passing function to function
    - anonymous functions as values
    - use function to reuse code
- 05 Scope
    - determines the accessibility of code
    - Global scope
        - any global variables or functions is accessible by all code within the page (including 3rd party code)
        - global scope is default
    - local (OR FUNCTIONAL) scope
        - variables and functions are limited to the function they are defined in
        - variables and functions cannot be accessed from outside the function
    - identifier look-up in scope chain
    - do not over write identifiers(names)
- 06 Working with Objects
    - Properties: data related to the objects
    - Methods: actions that do something with the object or its data
    - string object
        - properties
            - length
        - methods
            - indexOf() // start from beginning
            - lastIndexOf() // start from end
            - substr(start_index, number) // create a new string, not modify the old one
            - substring(start_index_include, end_index_exclude) // create

      a new string, not modify the old one
      - replace()
      - toUpperCase()
      - toLowerCase()

- 07 Creating Objects
  - old ways: using constructor function
    - var obj = new Object();
    - the "this" variable, use in method
  - modern way: object literal notation
  - other datatypes are also objects, have their own constructor function
    - Number(), String(), Boolean()
    - var str = new String("hello, world"); // don't use this, use literal values
    - pascal casing used in creating our own datatype
  - using object oriented programming to logically organize complex data
  - inherent programming
    - Object datatype is the mother of other datatype, and it has a method toString()
    - var num = 8,
    -     str = num.toString();
    - this way is more efficient than the concatenation operator
- 08 Arrays
  - array is a object
  - creating array
    - constructor function
    - array literal syntax
  - length property
  - method
    - array.push(element)
    - array.pop()
    - array1.concat(array2)
    - array.join(", ") // join a array into a string
    - array.reverse()
    - array.sort()
- 09 Conditions and Decisions
  - comparison operator
    - > >= < <=
  - equality operator
    - == !=
  - identity operator

            - === !==
        - if(condition){} else if(condition){} else{}
        - logical operator
            - AND && OR || NOT !
        - 0, "", undefined, null object --> false
        - other number, other string, object, array --> ture
    - 10 Loops
        - for loop
            - iterating arrays
        - while loop
        - do while loop
- JavaScript and the Browser
    - 11 The Window Object and More Scope
        - application on browser need to know the API on broswser
        - global variables and functions are the properties and method on window object
        - alert() --> window.alert() // window is always omitted, parseInt() and parseFloat() are also the method of window object
        - use immediately invoked function to create scope // avoid different code interrupt each other
            - (function(){...})(); OR (function(){...}());
        - more dialog box
            - alert()
            - confirm()
            - prompt()
        - location object
            - get and set URL in current browser window
    - 12 The Document Object and Finding Elements
        - DOM structure of HTML
        - document.getElementByTagName("p")
            - return a NodeList like an array
            - the NodeList is live, the change will be reflected
        - document.getElementById("foo")
        - css query
            - document.querySelector("p")
            - document.querySelectorAll("p")
            - document.querySelectorAll("div p")
            - document.querySelector("#foo")
        - getElementById() is faster than querySelector()
        - etElementByTagName() return a live NodeList, querySelectorAll()

does not

- 13 Creating Elements and Attributes
    - create, remove and modify the content that already loaded into the browser
    - the change is not permanent
    - add content
        - create element
            - var el = document.createElement("p")
        - give it content
            - var content = document.createTextNode("this is a p")
            - el.appendChild(content);
        - set it's attribute
            - el.setAttribute("align", "center")
            - el.id = "bar"
        - add it to the document
            - document.bady.appendChild(el)
    - appendChild(el) / insertBefore(el, pFoo) / replaceChild(el, pFoo)
    - DOM is slow
    - inner HTML property
- 14 Modifying Element Style
    - style attribute as style object
    - apply css sheet to element, use className property
        - className property
            - className.replace
        - classList property
            - classList.remove
            - classList.add
            - classList.toggle
    - get style information
        - standard based browser
        - IE8 and below
        - cross browser function, use feature detection
- 15 Timers and Animation
    - logic to do animation
        - manipulate css property
        - set a delay
    - setTimeout()
        - setTimeout loop
    - get out of setTimeout loop

        - counter
        - clearTimeout()
    - setInterval() / clearInterval()
    - ways to do animation
        - track the position of elements and set boundary
        - based on the steps

- Events
    - 16 Event Basics
        - execute code base on what the user does
        - DOM level 0 event handler
        - put event handler on the target element
        - onclick, onload
    - 17 The Standard Event Model
        - addEventListener("click", fn, false)
        - bubbling and capturing
            - bubbling == false
            - capturing == ture
            - or omit
        - removeEventLIstener("click", fn, false)
            - use the same fn, so can't use anonymous function object
        - some information is passed by parameter to the fn
            - evt.type
            - evt.target
            - evt.preventDefault()
    - 18 The Legacy IE (IE 8 and below) Event Model
        - attachEvent()
            - gobal object event
            - event.type
            - event.srcElement
            - event.returnValue = false
        - detachEvent()
    - 19 Cross Browser Event Handling
        - jQuery
        - eventUtility object
            - feature detection
        - two ways to access the property of a object
            - obj.property
            - obj["property"]
    - 20 Event Delegation

- use event bubbling
- flexiblity
- performance