JavaScript Fundamental 02

- Core Language
    - 01 Math
        - Math Object to provide some math method and property
        - Math.pow()
        - Math.sqrt()
        - Math.round()
        - Math.ceil()
        - Math.floor()
        - Math.max()
        - Math.min()
        - Math.abs()
        - Math.sin()
        - Math.cos()
        - Math.random()
        - Math.PI
        - Math.SQRT2
        - Math.SQRT1_2
    - 02 Date
        - create date object
            - var date = new Date(); // the time when the object is created
            - use number to create a specific date object
                - var date = new Date(1997, 0, 1, 22, 5);
            - use javascript time stamp
                - var date = new Date(865220700000);
        - method
            - get
                - date.getFullYear(); // 1997
                - date.getMonth(); // numeric value
                    - use array to get month string value
                - date.getDate()
                - date.getDay() // numeric value in week
                    - use array
                - getHours()
                - getMinutes
                - getSeconds
                - getMilliseconds

            - getTime // return time stamp
            - getTimezoneOffset() // return offset in minutes
        - set
            - setMonth()
    - use < or > on date object, can't use == or ===
        - use == or === on date.getTime()
- 03 Advanced Functions
    - two ways to create function
        - function declaration is hoisted to the top
        - function expression explicitly demonstrate object are values
            - create a function object and assign it to a variable
    - "this" variable
        - every function is actually a method of object
        - "this" refers to the object that the function is attached to
        - global functions are methods of window object
    - "arguments object"
        - traditional way: use parameter to reference the argument
        - use arguments object
            - arguments are the value passed to the function
            - is a object, behave like an array
        - use arguments object to reference the function we are calling
            - arguments.callee
            - used in recursive function
                - because we can accidentally dereference the function identifier
    - closure
        - when a function executes, execution context is created for each execution, and is cleared after the execution is done
        - a function return anonymous function when executed, the anonymous function has the ability to access the father function's parameter and local variables
        - because the memory is not freed, closure is bad sometimes
        - use immediately invoked function return a object, then the object has access to variables defined in the immediately invoked function
            - create a object with some external variables
- 04 Arrays
    - passing function to other function or method
    - array
        - indexOf() lastIndexOf()

            - iteration method
                - need create special functions to use, call back function
                    - function name(value, index, array) {}
                - fruit.every() // return true or false
                - fruit.some() // return ture or false
                - fruit.filter() // return a new array with the elements passed the test
                - forEach() // execute function for each element
                - map() // create a new array and modify that array
    - 05 Creating Custom Objects
        - object literal syntax to create single object
        - create a lot of object with the same interface
            - factory function
                - accept inputs as parameters and return a object
            - constructor function
                - like String Number Boolean Object Array Date
                - custom datatypes
                - when a function begins with a uppercase letter, there should be a "new" keyword before it
                - constructor function automatically return value
                - define method on the prototype property of a constructor function
                    - every other constructor function does that
                    - alert(Date.prototype.getDate);
    - 06 Functions As Objects
        - prototype property
            - constructor functions has prototype property
            - datatype in JavaScript use prototype define the method for that datatype
        - method
            - call a function as if they are method of other object
            - apply()
            - call()
        - arguments object
- DOM
    - 07 Keyboard Events
        - keypress event
            - not occur for modifier key
                - shift ctrl capslock
            - design for user input

            - event.charCode for standard
            - event.keyCode for legacy IE
            - event.charCode and event.keyCode return ascii numeric value for lowercase character, uppercase character, number, and punctuation
        - keydown and keyup event
            - event.keyCode // store the uppercase character ascii number for both uppercase character and lowercase character
            - event.ctrlKey // true or false
            - event.shiftKey
            - event.altKey
    - 08 Scripting Forms
        -

- Browser Features
    - 09 XMLHttpRequest
        - ajax : “Asynchronous JavaScript And XML” (异步JavaScript和XML)
        - use JavaScript to make a http request to the server and do something with the server's response
        - the "XMLHttpRequest" object
        - asynchronous mode and ready state code
            - can't rely on the order of execution
            - go through 5 state
                - xhr.readyState property
                - from 0 to 4, change value upon the request is going through
                - state 4 means the request was sent and we have a complete response from the server
                - when readyState change value, a readystatechange event occurs
                - handle the ready state change event
        - response status code
            - OK code
            - (status >= 200 && status < 300) || status === 304
        - steps
            - create a xhr object
            - xhr.open()
                - specify request type
                - specify URL
                - specify mode
                    - true = asynchronous mode
                    - false = synchronous mode
            - use onreadystatechange event handler check for the ready state

        - check for the response status
        - send the request
    - POST
        - specify the http header
            - xhr.setRequestHeader()
            - "header" "content"
        - send data
            - assemble all the data
            - form object has a "elements" property which contains all the form control elements within that form
            - format : fieldName=value&fieldName=value
            - encodeURIComponent()
- 10 JSON
    - JavaScript Object Notation
    - xml
    - js object literal -> json (everything is string or number)
    - JSON.parse()
    - JSON.stringify()
    - retrieve data
    - send data
    - IE 7 does not support, need external libray
- 11 Local Storage
    - cokies
    - not supported by IE 7
    - localStorage object
    - name and value pair format
    - two way to store or retrieve data
        - store
            - localStorage.firstName = "John"
            - localStorage.setItem("lastName", "Doe")
        - retrieve
            - localStorage.getItem("firstName")
            - localStorage.lastName
    - remove data
        - localStorage.removeItem("lastName")
    - localStorage.clear()
    - number is converted to string
    - restrict to one domain
- 12 Error handling
    - try {code may fail in here, when user input data} catch(err) {display the

error message} finally {this code will always execute}