**11/15 김민식 OJT Day 10**

**Html/css/javascript/jquery 기본교육**

**JavaScript 재입문하기(JS 튜토리얼)**

[생활코딩 - 자바스크립트(JavaScript) 기본 | 학습 페이지 (inflearn.com)](https://www.inflearn.com/course/%EC%A7%80%EB%B0%94%EC%8A%A4%ED%81%AC%EB%A6%BD%ED%8A%B8-%EC%96%B8%EC%96%B4-%EA%B8%B0%EB%B3%B8/unit/2456?tab=curriculum) – 강의 학습

6. 반복문

- for 문

for(var i = 0; i<10; i = i + 1){

            document.write("coding everybody <br/> ");

        }

* 반복문의 제어
  + Break – 반복문의 종료
  + Continue – 조건의 일부 반복문 생략
* 반복문의 중첩, 디버거
  + 크롬 디버거
    - 관리자 페이지 – Breakpoint 설정 후 새로고침, 아래 화살표 내리면 진행, 변수값도 파악 가능

7. 함수 (function)

- 하나의 로직을 재 실행 할 수 있도록 코드의 재 사용성을 높여준다

- 함수의 형식

function numbering(){

            document.write(1);

        }

        numbering();

* 함수의 효용성
  + 재사용성
  + 유지보수의 용이
  + 가독성
* 함수의 출력
  + Return
* 함수의 입력
  + 매개변수 (parameter) , 인자(argument)
* 다양한 함수 정의 방법

numbering = function (){

            i = 0;

            while(i<10){

                document.write(i);

                i += 1;

            }

      }

      numbering();

* 익명함수

(function (){

            i = 0;

            while(i<10){

                document.write(i);

                i += 1;

            }

        })();

8. 배열(Array)

- 배열의 문법

var member = [‘egoing’, ‘k8805’, ‘sorialgi’];

* + 각각의 데이터 – 원소(Element) / ‘egoing’ <-0 색인(Index) ‘k8805’ <-1 ‘sorialgi’ <-2
  + member[0] == egoing / member[1] == k8805
* 배열의 조작 \_추가
  + 배열의 크기 : arr.length;
  + 배열 인자 추가 (맨 뒤에) : arr.push();

var li = ['a','b','c','d','e'];

        li.push('f');

        alert(li);

* + 여러 개의 배열 인자 추가 : arr.concat(‘a’,’b’]);
  + 배열 인자 추가 (맨 앞에) : arr.unshift();
  + 배열의 특정구간 추출, 특정구간에 특정 배열을 추가 : arr.splice(index, howmany, element1. … elementN); (Index, Howmany 는 필수 element는 옵션)
* 배열의 조작 \_ 제거, 정렬
  + 배열의 첫번째 원소를 제거 : arr.shift();
  + 배열의 끝점의 원소를 제거 : arr.pop();
  + 배열 정렬 : arr.sort();
  + 배열 역순으로 정렬 : arr.reverse();

9. 객체 (Object)

var grades = {'egoing':10,'k8805':6,'sorialgi':80};

위의 예제에서 egoing은 key, 10은 value

* 객체의 값을 가져올 때 : grades.egoing or grades[‘k8805’]
* 객체와 반복문
  + For in 문

        var grades = {'egoing':10,'k8805':6,'sorialgi':80};

        for(key in grades){

            document.write("key : " +key+" value : "+grades[key] + "<br />")

        }

* + - Key라는 변수가 key값을 가져온다.
* 객체 지향 프로그래밍
  + 객체엔 함수도 넣을 수 있다.( 함수를 사용하려면 ()를 추가)
* var grades = {
* 'list' : {'egoing' : 10, 'k8805' : 8 , 'sorialgi' : 80},
* 'show' : function(){
* alert('hello world');
* }
* }
* grades['show']();

10. 모듈

* 부품, 코드의 재활용성을 높이고, 유지보수를 쉽게 할 수 있는 기법
* 모듈의 효과
  + 자주 사용되는 코드를 별도의 파일로 만들어 필요할 때마다 재활용
  + 코드 수정 시 필요한 로직을 빠르게 찾을 수 있다.
  + 필요한 로직만을 로드해서 메모리의 낭비를 줄일 수 있다.
* 모듈화

    <script src="greeting.js"></script>

* Node.js에서의 모듈의 로드
  + 호스트 환경에 따라서 모듈을 로드하는 방법이 달라지는 것의 예시
* 라이브러리
  + 모듈과 비슷한 개념
  + 유명 라이브러리 = jQuery

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

<script src="http://code.jquery.com/jquery-latest.js"></script>

    <script>

        $(document).ready(function(){

            alert(welcome());

            $('#execute\_btn').click(function(){

                $('#list li').text('coding everybody');

            });

        });

    </script>

11. UI와 API 그리고 문서보는 법

* API란?
  + Application Programming Interface – 사용자에게 보이진 않지만 시스템을 구축을 도우는 중간 단계 (ex. alert 창의 모양, 위치 등 코드외 자동 구축)
  + 자바스크립트 API는 크게 2가지로 구분
    - 자바스크립트 자체의 API
    - 자바스크립트가 동작하는 호스트 환경의 API
* UI란?
  + User Inteface – 사용자의 의중을 직접 접촉
* 문서 보는법
  + 레퍼런스 – 사전
  + 튜토리얼 – 안내서
  + <https://tc39.es/ecma262/> - 표준 문서
  + <https://opentutorials.org/course/50> - 생활코딩

12. Javascript와 정규표현식

* 정규표현식(regular expression)은 문자열에서 특정한 문자를 찾아내는 도구
  + 컴파일 : 패턴을 찾는 것
  + 실행 : 대상에 대해 구체적인 무언가를 하는것
* 패턴 만들기
  + var pattern = new RegExp('a'); === var pattern =/a/;
* RegExp 객체의 사용
  + 정규표현식 메소드 실행 (추출, test, 치환)
    - RegExp.exec(); : 패턴의 대상을 뽑아줌 – 추출
    - RegExp.test() : true, false로 검색 대상에 패턴이 존재 구분 – 테스트
* String과 정규표현식
  + String.match(); : RegExp.exec()와 비슷

var pattern = /a/;

var str = 'abcdef';

alert(str.match(pattern));

* + String.replace() : 변경된 값을 리턴

        var pattern = /a/;

        var str = 'abcdef';

        alert(str.replace(pattern,'A'));

* 옵션
  + I : 대소문자를 구분하지 X

var xi = /a/;

        var oi = /a/i;

        alert('Abcde'.match(oi));

* + G : 검색된 모든 결과를 리턴

       var xg = /a/;

        var og = /a/g;

        alert('abcdea'.match(og));

* 캡처

var pattern = /(\w+)\s(\w+)/ // \w+:1이상 문자, \s : 공백

* + $ : 패턴 그룹 지정
* 치환

        var urlPattern = /\b(?:https?):\/\/[a-z0-9-+&@#\/%?=~\_|!:,.;]\*/gim;

        var content = '생활코딩 : http://opentutorials.org/course/1 입니다. 네이버 : http://naver.com 입니다. ';

      var result = content.replace(urlPattern, function(url){

            return '<a href="'+url+'">'+url+'</a>';

      });

        console.log(result);

13. 유효범위

* 전역변수(global)와 지역변수(local)

var vscope = 'global';

        function fscope(){

            var vscope = 'local';

            alert(vscope);

        }

        fscope();

* 유효범위의 필요성

function a() {

       i = 0;

       }

       for(var i=0; i <5; i++){

      a();

      document.write(i);

      } //무한으로 루프가 돌아감

* 정적 유효 범위 (static scoping or lexical)
  + 자바스크립트는 함수가 선언된 시점에서의 유효범위를 가짐

14. 값으로서의 함수와 콜백

* 함수의 용도 – 1
  + 함수도 객체, 일종의 값

        a = {

            b:function(){

            }

        }

* + - b : key, 변수, 속성, Property
    - 객체의 속성 값으로 담겨진 함수 – method
  + 함수는 값, 다른 함수의 인자로 전달 가능

      function cal(mode){

            var funcs = {

                'plus' : function(left, right) {return left + right},

                'minus' : function(left, right){return left- right}

            }

            return funcs[mode];

        }

        alert(cal('plus')(2,1));

* 콜백

        var numbers = [20, 10, 9, 8, 7, 6,5,4,3,2,1];

        var sortfunc = function(a, b){

            console.log(a,b);

            if(a > b){

                return 1;

            }else if(a < b){

                return -1;

            }else{

                return 0;

          }

        }

        console.log(numbers.sort(sortfunc));

* 비동기 콜백과 Ajax (asynchronous Javascript and Xml)

15. 클로저(closure)

내부함수가 외부함수의 맥락(context)에 접근할 수 있는 것

* 내부함수, 외부함수

    function outter(){

            function inner(){

                var title = 'coding everybody';

                alert(title);

            }

            inner();

        }

        outter();

* 클로저란?

        function outter(){

            var title = 'coding everybody';

            return function(){

                alert(title);

            }

        }

        var inner = outter();

        inner();

* Private variable

function factory\_movie(title){

            return{

                get\_title : function(){

                    return title;

              },

                set\_title : function(\_title){

                    title = \_title

                }

            }

        }

        ghost = factory\_movie('Ghost in the shell');

        matrix = factory\_movie('Matrix');

        alert(ghost.get\_title());

        alert(matrix.get\_title());

        ghost.set\_title('공각기동대');

        alert(ghost.get\_title());

* 클로저의 응용

var arr = [];

        for(var i = 0; i <5; i++){

            arr[i] = function(id){

                return function(){

                    return id;

                }

            }(i);

        }

        for(var index in arr){

            console.log(arr[index]());

        }

16. arguments

* 유사 배열, 특수한 자바스크립트 객체
  + Agrgument.length;

        function sum(){

            var i, \_sum = 0;

            for(i = 0; i < arguments.length; i++){

                document.write(i+' : '+arguments[i]+'<br />');

                \_sum += arguments[i];

            }

            return \_sum;

        }

        document.write('result : ' + sum(1,2,3,4));

17. 함수의 호출

* Apply 소개
  + .apply

        function sum(arg1, arg2){

            return arg1+arg2;

        }

        alert(sum.apply(null,[1,2]));

* Apply 사용

       o1 = {val1:1, val2:2, val3:3}

        o2 = {v1:10, v2:50, v3:100, v4:25}

        function sum(){

            var \_sum = 0;

            for(name in this){

                \_sum += this[name];

            }

            return \_sum;

        }

        alert(sum.apply(o1)) // 6

        alert(sum.apply(o2)) // 185

질문사항

1. Domain, DTO, VO 차이 \_ nShop code
2. ~~MyBatis\_ xml파일 위치 \_ nShop code~~
3. .ajax 처리방법 (json파일의 위치 or dataflow) \_nShop/views/admin/common/code/insert.jsp
4. 시도해볼만한 자격증