

❖ document 객체

◆ form 객체

- * 입력 태그들의 입력 내용을 웹 프로그램에 전달하고 입력 내용을 제어

❖ form 객체의 접근은 name 값으로 하므로 name속성을 지정해주자.

- * id 값에 의한 객체 획득 방법

```
변수 = document.getElementById("id이름");
```

- * name값에 의한 객체 획득 방법

```
변수 = document.form-name값;
```

❖ document 객체

◆ form 객체

- * 입력값 처리 - value 속성으로 읽고 쓴다.

```
변수 = document.form_name.태그name.value;    // 읽기  
document.form_name.태그name.value = 데이터; // 쓰기
```

- ❖ value 속성은 input 태그의 입력 여부 체크에도 사용할 수 있다.

```
var myform = document.form1;  
var username = myform.user_name;  
  
if (!username.value){  
    alert( "내용을 입력하세요");  
}
```

❖ document 객체

◆ form 객체

- * 선택 항목 처리
 - select 태그에서 사용
 - selectedIndex 속성 사용 (index는 0 부터 시작)
- * 선택 항목 Index 얻기

```
var no = document.form_name.태그name.selectedIndex;
```

- * 선택 항목 데이터 얻기

```
var no = document.form_name.태그name.selectedIndex;  
var str = document.form_name.태그name[no].value;
```

- * 선택 여부 검사

```
<form name="form1">  
  <select name="foo">  
    <option>###선택하세요!###</option>  
    <option>항목 1</option>  
    <option>항목 2</option>  
  </select>  
</form>
```

```
var myform = document.form1;  
if(myform.foo.selectedIndex < 1){  
  ## 처리내용 ##  
}
```

❖ document 객체

◆ form 객체

- * 라디오 버튼과 체크박스의 체크 상태
 - checked 속성을 사용 - 체크되면 true
- * focus 지정 - focus() 함수로 강제 지정

```
document.form1.foo.focus();
```

- * 작성된 내용 리셋 - reset() 함수 사용

```
document.form1.reset();
```

- * 작성된 내용 전송 - submit() 함수 사용

```
document.form1.submit();
```

- * 작성된 내용 전송 하기전 입력 여부 검사 - onsubmit() 이벤트 함수 사용

```
<form name="form1" onsubmit="함수이름(); return false;">
```

❖ JSON 표기법

- * 경량의 데이터 교환 형식
- * 구문형식은 Javascript를 따르지만 언어로서 독립적
- * 프로그래밍언어간 데이터 교환을 위해 가장 널리 쓰이는 표기법
- * Javascript 문법으로 사용
- * 자체가 하나의 데이터로 사용

◆ 기본 형식

```
var 객체이름 = { 이름: 값, 이름: 값, ... };
```

```
var 객체이름 = { };  
객체이름.이름 = "새로운값";
```