

JSP

[04] Servlet 웹프로그래밍 II

- **Get방식** – URL값으로 정보가 전송되어 보안에 약함
 - 주소와 매개변수를 붙여서 주소 표시줄에 입력하는 방법(?로 구분)
 - a 태그를 이용해서 페이지를 요청하는 경우
 - form태그에서 명시적으로 GET방식으로 요청하는 경우(method="get")
 - 매개변수의 데이터는 255자 이내이며 보안성이 없음
 - **get으로 넘어가는 경우가 더 많음 post로 넘어가는 경우는 한경우밖에 없음**
- **Post방식** – header를 이용해 정보가 전송되어 보안에 강함
 - form태그에서 명시적으로 POST 방식으로 요청(method="post")
 - 데이터의 크기에 제한이 없으며 URL에 표시되지 않으므로 보안성이 우수
 - 한글 파라미터 전달시 request.setCharacterEncoding("utf-8")추가
 - **post방식은 url에 안나오기 때문에 보안에 강한 것**

checkbox 할 때 체크가 되어 있지 않을 경우 넘어가지 않는 로직은 자바스크립트를 통해서 해야 됨 ..

```
<script>
//함수는 queryselect만 사용
window.onload=function(){
    // submit 디폴트 이벤트 제한 : checkbox가 한개도 체크되어 있지 않을 때
    document.querySelector('form').onsubmit= function(){
        var hobby = document.querySelectorAll('input[name="hobby"]'); //배열
        var cnt = 0; //체크박스가 몇개 체크 될지 저장하기 위한 변수
        for(var i=0; i<hobby.length; i++){
            /*if(hobby[i].checked){ //체크되어 있으면 true 안되어 있으면 false 자바스크립트는 true를 연산하려고 하면 1이 됨 false는 0
                cnt++;
            }*/ //이렇게 해도 되지만
            cnt = cnt+ hobby[i].checked;

        }
        if(!cnt){ //cnt가 0이면 if문이 참일 것
            alert('취미를 한개 이상 선택하셔야 합니다');
            return false;
        }
        return true;
    }; //onsubmit
}; //onload
```

Window.onload = function = 윈도우가 실행될 때 함수안에다가

Query가 form인 아이의 onsubmit을 실행시키면 ?

Hobby 종류를 hobby에 넣고 , 체크박스의 개수를 저장할 변수를 하나 지정해놓은 다음 for문을 돌리기

For문 안에서 hobby[i].checked 가 true일 경우 1 false일 경우 0이므로 한 개라도 선택하게 되면 값이 1이상일 즉 cnt안에 그 수를 누적시켜 cnt 가 1이상일 경우엔 submit이 일어나고 0일경우엔 submit이 일어나지 않도록 하면 됨

Radio 타입 할 때 아무것도 체크가 되어 있지 않으면 서브밋이 안되도록 하게 하는 것

```
-----
<script>
  window.onload = function(){
    document.querySelector('form').onsubmit = function(){
      //디폴트 이벤트 제한 : 동의합니다를 check하지 않으면 제한
      var agreeYes = document.querySelector('input[name="agree"]'); //name=agree가 두개라도, 맨 첫번째것!
      if(!agreeYes.checked){
        alert('동의하셔야 진행 가능합니다');
        return false;
      }
    };
  };
</script>
</head>
```

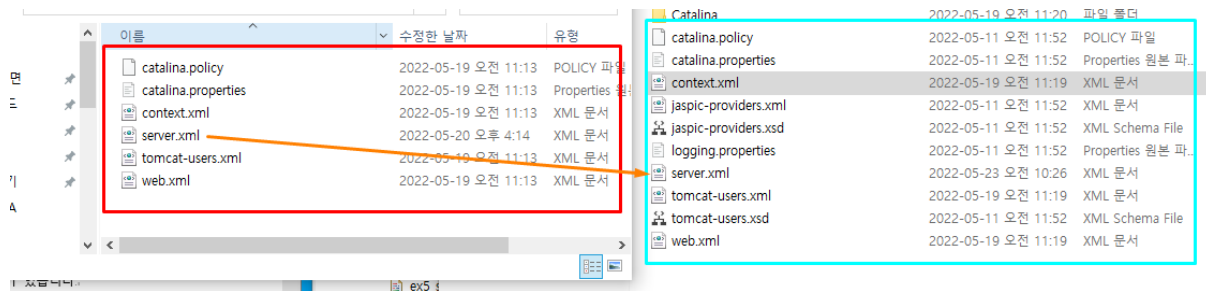
Window.onload의 function을 만든 후 그 안에 form query를 가지고오고 form에 있는 onsubmit을 실행했을 때 필요한 로직을 넣을 function 만들기

그 function 안에 agreeYes 를 했을때만 넘어갈수있도록 만들어야 하니 agreeYes 부분(즉 첫번째 부분이 체크가 되었는지 안되었는지만 확인하면 됨)

Select 타입 할 때 아무것도 체크 되어 있지 않으면 서브밋이 안되도록 하게 하는 것

```
-----
<script>
  window.onload = function(){
    document.querySelector('form').onsubmit = function(){
      var opt = document.querySelectorAll('option'); //배열
      var cnt = 0;
      for(var i=0; i<opt.length; i++){
        cnt = cnt+ opt[i].selected;
      }
      if(!cnt){
        alert('한개이상 선택하세요');
        return false;
      }
    };
  };
</script>
```

Check박스 방식이랑 비슷 함



빨간박스 : 소스폴더의 server

파란박스 : ide의 톰캣폴더 의 conf

톰캣을 켜다 켤때마다 주기적으로 server의 파일이 conf로 업데이트됨

이렇게 업데이트 되는 것 즉 파일을 가는걸 퍼블리쉬라고 부름

톰캣을 안고면 안웁김. 톰캣을 끌때마다 웁기는 것 즉 실행할때마다 톰캣 restart를 하는 걸 말함

```
public Ex3_LifeCycle() {
    //super(); 애 빼도 알아서 매개변수 없는 super생성자 호출
    System.out.println("Ex3_LifeCycle 생성자 - 1");
    // 메모리에 객체를 막 얹어놓으려고 할 때 생성되는 아이
    // 생성자는 객체 만들기 전에 한번 호출됨
    // 생성자 함수는 객체가 막 만들어지려고 할때 호출되는것. init는 생성자 함수가 생성되는 순간 호출
}
@PostConstruct//함수이름은 내 마음대로 해도됨
public void postConstruct() {
    System.out.println("Ex3_LifeCycle 서버릿 객체가 생성되었을때 수행 - 2-1");
}

public void init(ServletConfig config) throws ServletException {
    // init는 객체가 생성될 때 , 최초 한 번 수행 호출됨
    System.out.println("Ex3_LifeCycle 서버릿 객체가 생성되었을때 수행 - 2-2");
}

protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException {
    // service라는 함수가 있으면 doget과 doPost 무시되고 service만 수행
    // 클라이언트의 요청이 들어왔을 때 수행
    System.out.println("Ex3_LifeCycle service() 실행 - 3");
    response.setContentType("text/html; charset=utf-8");
    PrintWriter out = response.getWriter();
    out.print("<h2>service()실행했을 경우 </h2>");
    out.close();
}

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException {
    // 클라이언트 요청이 get 방식으로 들어왔을 때 수행 (service함수가 없을때)
    System.out.println("doGet() 실행");
    response.setContentType("text/html; charset=utf-8");
    PrintWriter out = response.getWriter();
    out.print("<h2>doGet()실행했을 경우 </h2>");
    out.close();
}
```

```

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException
// 클라이언트 요청이 post 방식으로 들어왔을 때 수행 (service함수가 없을때)
System.out.println("doPost() 실행");
response.setContentType("text/html; charset=utf-8");
PrintWriter out = response.getWriter();
out.print("<h2>doPost()실행했을 경우 </h2>");
out.close();

}

public void destroy() {
// Ex3_LifeCycle 서블릿이 메모리 해제될 때
// 메모리에서 없어지는 거기 때문에 브라우저에 쓰는 건 의미가 없음
System.out.println("destroy() 실행(메모리 해제)- 4-1" );
}

@PreDestroy//함수이름은 내 마음대로 해도됨
public void predestroy() {
System.out.println("destroy() 실행(메모리 해제) - 4-2");
}

```