

# 2023-12-01

[회원가입]

회원 가입

Home | 회원가입

CREATE ACCOUNT

이름\*

아이디\*

중복 확인

비밀번호\*

비밀번호 확인\*

주소\*

우편번호

우편번호 검색

주소

상세주소

전화번호\*

Email\*

이용약관 동의 (필수)

제1조(목적) 이 약관은 협세 회사(전자상거래 사 업자)가 운영하는 협세 사이버 물(이하 "물"이라 한다)에서 제공하는 인터넷 관련 서비스(이하 "서 비스"라 한다)를 이용함에 있어 사이버 물과 이용 자의 권리, 의무 및 책임사항을 규정함을 목적으로 합니다. ※"PC통신, 무선 등을 이용하는 전자상가 래에 대해서도 그 성질에 반하지 않는 한 이 약관

개인정보 수집 및 이용 동의 (필수)

개인정보처리방침 [차별] 1. 총칙 2. 개인정보 수 집에 대한 동의 3. 개인정보의 수집 및 이용목적 4. 수집하는 개인정보 항목 5. 개인정보 자동수집 장치의 설치, 운영 및 그 거부에 관한 사항 6. 목적 외 사용 및 제3자에 대한 제공 7. 개인정보의 열람 및 정정 8. 개인정보 수집, 이용, 제공에 대한 동의 철회 9. 개인정보의 보유 및 이용기간 10. 개인

☐ 서비스 이용 약관에 동의 하셨습니다?  
☐ 개인정보 수집 약관에 동의 하셨습니다?  
☐ 개인정보 이용 약관에 동의 하셨습니다?

회 원 가 입

```
// 회원가입 관련 input들
const username = document.querySelector("input[name='username']");
const userId = document.querySelector("input[name='id']");
const userPw = document.querySelector("input[name='pw']");
const userPwCheck = document.querySelector("input[name='pw_check']");
const userAdress = document.querySelector("input[name='adress']");
const userAdressDetails = document.querySelector("input[name='adress_Details']");
const userZipcode = document.querySelector("input[name='zipcode']");
const userPhone = document.querySelector("input[name='phone']");
const userEmail = document.querySelector("input[name='email']");
```

input들에 기능을 주기 위해 쿼리 셀렉터로 미리 전부 가져왔다.

```
//우편번호 검색, 자동완성 (Daum)
function zipCodeSearch() {
    new daum.Postcode({
        oncomplete: function(data) {
            // 팝업에서 검색결과 항목을 클릭했을때 실행할 코드를 작성하는 부분.

            // 각 주소의 노출 규칙에 따라 주소를 조합한다.
            // 내려오는 변수가 값이 없는 경우엔 공백('')값을 가지므로, 이를 참고하여 분기 한다.
            var addr = ''; // 주소 변수
            var extraAddr = ''; // 참고항목 변수

            //사용자가 선택한 주소 타입에 따라 해당 주소 값을 가져온다.
            if (data.userSelectedType === 'R') { // 사용자가 도로명 주소를 선택했을 경우
                addr = data.roadAddress;
            } else { // 사용자가 지번 주소를 선택했을 경우(J)
                addr = data.jibunAddress;
            }

            // 사용자가 선택한 주소가 도로명 타입일때 참고항목을 조합한다.
```

```

        if(data.userSelectedType === 'R'){
            // 법정동명이 있을 경우 추가한다. (법정리는 제외)
            // 법정동의 경우 마지막 문자가 "동/로/가"로 끝난다.
            if(data.bname !== '' && /[동|로|가]$/g.test(data.bname)){
                extraAddr += data.bname;
            }
        }

        // 우편번호와 주소 정보를 해당 필드에 넣는다.
        document.getElementById('zip_code').value = data.zonecode;
        document.getElementById("adress_input").value = addr;
        // 커서를 상세주소 필드로 이동한다.
        document.getElementById("adress_input_details").focus();
    }
    }).open();
}
}

```

우편번호와 주소를 자동입력 하게 하기위한 api

```

// 회원가입버튼 관련 조건들, if만 사용할시 한줄로 정리 가능하다.
function signUpButton(){
    const accOr = document.querySelector("#acc-or").checked;
    const payment = document.querySelector("#payment").checked;
    const paypal = document.querySelector("#paypal").checked;

    if(!username.value) return alert("이름을 입력 해주세요.");
    if(!userId.value) return alert("아이디를 입력 해주세요.");
    if(checkUserId()) return alert("중복된 아이디 입니다.");
    if(!userPw.value) return alert("비밀번호 확인이 틀립니다.");
    if(!userPwCheck.value) return alert("비밀번호 확인이 틀립니다.");
    if(userPw.value !== userPwCheck.value) return alert("비밀번호 확인이 틀립니다.");
    if(!userZipcode.value) return alert("우편번호를 입력 해주세요.");
    if(!userAddress.value) return alert("주소를 입력 해주세요.");
    if(!userAddressDetails.value) return alert("상세주소를 입력 해주세요.");
    if(!userPhone.value) return alert("전화번호를 입력 해주세요.");
    if(!userEmail.value) return alert("이메일을 입력 해주세요.");
    if(!accOr) return alert("약관에 동의 해주세요.");
    if(!payment) return alert("약관에 동의 해주세요.");
    if(!paypal) return alert("약관에 동의 해주세요.");
    else{
        // 데이터js의 내용이 한번만 저장되게 함
        if(localStorage.userData === null || localStorage.userData === undefined || JSON.stringify(localStorage.userData) === "[]"){
            localStorage.setItem("userData", JSON.stringify(userData));
        }

        const localData = JSON.parse(localStorage.getItem("userData") || "[]");
        console.log(localData)
        localData.push({
            userNo:localData.length+1,
            admin:userId.value == "park" || userId.value == "son" ? 1:0,
            name:username.value,
            phone:userPhone.value,
            id:userId.value,
            pw:userPw.value,
        });
        localStorage.setItem("userData", JSON.stringify(localData));

        alert("회원가입이 완료 되었습니다.")
        return location.href="login.html";
    }
}
}

```

회원가입시 필수 항목들에 입력된 값이 없으면 회원가입 버튼을 눌러도 가입 되지않고 알림창이 뜬다.

if문이 너무 길어져서 한 줄씩으로 정리했다.

return을 주지않으면 밑 if들도 실행된다.

모든 값을 입력했을때 push로 객체 형태로 넣어준다.

유저 넘버는 가입된 순서대로 1씩 증가 하게 했고 admin권한은 id명을 정해냈다.

나머지 정보는 입력했던 값이 저장된다.

```

// 회원가입 버튼에서도 최종 중복체크를 하고, alert이 나오게 하기 위한것.
function checkUserId(){
  const idfind = userData.find((data)=>{
    return data.id == userId.value
  })
  return idfind
}

// 아이디 중복체크 관련 (checkUserId()과 연결할수 있는지 알아보기)
function userIdMatch(){
  const idfind = userData.find((data)=>{
    return data.id == userId.value
  })
  if(idfind) return alert("중복된 아이디 입니다.");
  if(userId.value == "") return alert("아이디를 입력 해주세요.");
  if(!idfind) return alert("사용 가능한 아이디 입니다.");
}

```

find로 로컬스토리지에 id가 같은게 있는지 찾아준다 (중복 체크)