**Animal User**

1. Write a function, AnimalTestUser, that has one string parameter, username. It returns an object with a username property.
   1. var testSheep = AnimalTestUser('CottonBall');

console.log(testSheep); **//{ username: 'CottonBall' }**

1. In your AnimalTestUser function, create a check that sees how many arguments are passed. If there is more than one argument, create a property, otherArgs that is an array of the remaining arguments. Note: the arguments keyword is not a true array. Remember, it is an array-like object.
2. var testSheep = AnimalTestUser('CottonBall', {'loves dancing': true}, [1,2,3] );

console.log(testSheep); //{ username: 'CottonBall', otherArgs: [ {'loves dancing': true}, [1,2,3] ] }

1. Write a constructor function, AnimalCreator that returns a single animal object. The constructor function has 4 parameters: username, species, tagline, and noises. The animal object should have at least 5 properties: username,species, noises, tagline, and friends. The values should all be strings except noises and friends, which are arrays.

var sheep = AnimalCreator('Cloud', 'sheep', 'You can count on me!', ['baahhh', 'arrgg', 'chewchewchew']);

console.log(sheep);

// { username: 'Cloud',

// species: 'sheep',

// tagline: 'You can count on me!',

// noises: ['baahhh', 'arrgg', 'chewchewchew'],

// friends: []

// }

1. Write a function, addFriend that takes an animal object (like the one returned from the AnimalCreator function) and adds another animal object as a friend.

addFriend(sheep, cow);

console.log(sheep);

// { username: 'Cloud',

// species: 'sheep',

// tagline: 'You can count on me!',

// noises: ['baahhh', 'arrgg', 'chewchewchew'],

// friends: [{username: 'Moo', species: 'cow'...}]

// }

addFriend(sheep, llama);

console.log(sheep);

// { username: 'Cloud',

// species: 'sheep',

// tagline: 'You can count on me!',

// noises: ['baahhh', 'arrgg', 'chewchewchew'],

// friends: [{username: 'Moo', species: 'cow'...}, {username: 'Zeny', species: 'llama'...}]

// }

1. Change your addFriend function to only add the username of the friend, not the whole object.

addFriend(sheep, cow);

console.log(sheep);

// { username: 'Cloud',

// species: 'sheep',

// tagline: 'You can count on me!',

// noises: ['baahhh', 'arrgg', 'chewchewchew'],

// friends: ['Moo']

// }

addFriend(sheep, llama);

console.log(sheep);

// { username: 'Cloud',

// species: 'sheep',

// tagline: 'You can count on me!',

// noises: ['baahhh', 'arrgg', 'chewchewchew'],

// friends: ['Moo', 'Zeny']

// }

1. Create a myFarm collection of at least 3 animal objects. Give them some friends using addFriend, too!

console.log(myFarm) //[{username: 'Cloud'...},{username: 'Zeny'...},{username: 'CottonBall'...}]

1. Create a function, addMatchesArray, that takes a farm (array of animal objects) and adds a new property to each animal object called matches that is an empty array. Hint: you will need a loop.
2. addMatchesArray(myFarm);
3. console.log(myFarm[0]); // { username: 'Cloud',
4. // species: 'sheep',
5. // tagline: 'You can count on me!',
6. // noises: ['baahhh', 'arrgg', 'chewchewchew'],
7. // friends: ['Moo', 'Zeny'],
8. // matches: []

// }

1. Create a function, giveMatches, that takes a farm collection (aka an array of animal objects) that already has a matches property. It selects a name from the friends array and adds it to the matches array. You can choose how the selection is made (random, the first element, etc). Make sure all your animal objects have friends.
2. giveMatches(myFarm);
3. console.log(myFarm[0]); // { username: 'Cloud',
4. // species: 'sheep',
5. // tagline: 'You can count on me!',
6. // noises: ['baahhh', 'arrgg', 'chewchewchew'],
7. // friends: ['Moo', 'Zeny'],
8. // matches: ['Zeny']

// }