

Secret Santa

A function that reproduces the process of drawing names out of a hat for gift assignments.

Problem Statement

One Christmas season a group of friends decides that they don't have money for everyone to give everyone else a gift. Instead they will arrange it so that everyone gives one gift, and everyone gets one gift.

Everyone puts their names on slips of paper and puts them in a hat. Each takes a turn drawing a name out of the hat.

There are three requirements for the gift assignments:

1. Everyone gives a gift.
2. Everyone gets a gift.
3. No one gives to themselves.
4. The assignments are unpredictable.

Write a function named `ssanta` that takes an array of strings—friends—and returns an object describing the gift giving assignments.

For example, if I call the function with the list `['Gene', 'Kim', 'Pat']` then it should return an object like `{Gene: 'Pat', Kim: 'Gene', Pat: 'Kim'}`.

Suitability

Students might come across this problem in their studies but probably won't. For that reason it is reasonably fair when interviewing candidates from various backgrounds.

Implementation Quirks

You can expect applicants to know how to use the language – loops, variables, object literals. You may want to let the student get started with what they already know. Then be prepared to give them information on functions that could be useful. For example, `Math.random()`.

There are many ways to solve this problem. I've often been surprised by novel approaches to the solution. Still, the point isn't to surprise the interviewer. A successful applicant will talk through the problem and show that they can break it down into a solution.

Interview Waypoints

State the problem clearly and consistently every time using the **Problem Statement** above.

- Let the applicant make a start at the problem. Wait for them to make some basic progress like the function signature.
- If they don't know how to randomize the solution ask them to go as far as they can without that.
- When they have made significant progress ask them what questions they have, or what functions they would like to use for help. Offer up documentation on `Math.random()`.
- Before finishing up have them run the program with a list three or four names long.
- Make sure they avoid people giving to themselves, repeat receivers, and missed receivers.

Expanding Beyond JS

Once the function has been defined you can quickly probe into their comfort with assembling web pages by asking them to expand their solution into a simple web page.

Problem Statement:

Given the function you wrote please put it in it's own file. Write the markup for an html page that when viewed in the browser shows an input field with a button. The form should let users type in a list of first names, split on spaces, and print a list of gift assignments. The page should load your reusable function and call it when the button is clicked. Finally, link a CSS file to the page. In the CSS file write a rule that causes the giver names to be red and the receiver names to be red.

Implementation Quirks

Expanding the problem into a simple web page gives you the chance to see how comfortable the applicant is making a complete web page. Do they understand the mechanics? Having the syntax memorized isn't necessary, but do they know the kinds of tags and attributes required?

You might ask them how they would go about finding the proper syntax. Where would they get their boilerplate from?

Conclusion

The secret santa problem is a little more obscure and has a fair amount of freedom in the solution. It isn't very mathematically challenging. It does do some simple list manipulation which can be common in web-applications. It is challenging enough to serve as an effective screening device – especially when used consistently for several interviews.

Ideas for Improvement

- Add helpful documentation of relevant apis to the "Interview Waypoints" section.