Cartoon Network
Developer Quiz

Question 1:

Using Node.js, write a function or module that returns a specific property from the following object.

```
var obj = {
    title: 'The Heist',
    seasonNumber: '5',
    episodeNumber: '12',
    show: {
        showName: 'Gumball',
        showId: '2000345',
    }
 };
```

Requirements:

- The function should take three parameters: the object, the property, and a default value.
- If the property exists in the object, return that property.
- If the property is undefined or null, return the default value.

Examples:

- yourFunction(obj, "seasonNumber", "8") should return "5".
- yourFunction(obj, "show.showId", "000") should return "2000345".
- yourFunction(obj, "episode.rating", "TV-Y7") should "TV-Y7" without any errors/exceptions.

Question 2:

Using Node.js, perform the tasks below using the data from this JSON file:

https://www.cartoonnetwork.com/test/backend-quiz/games.json

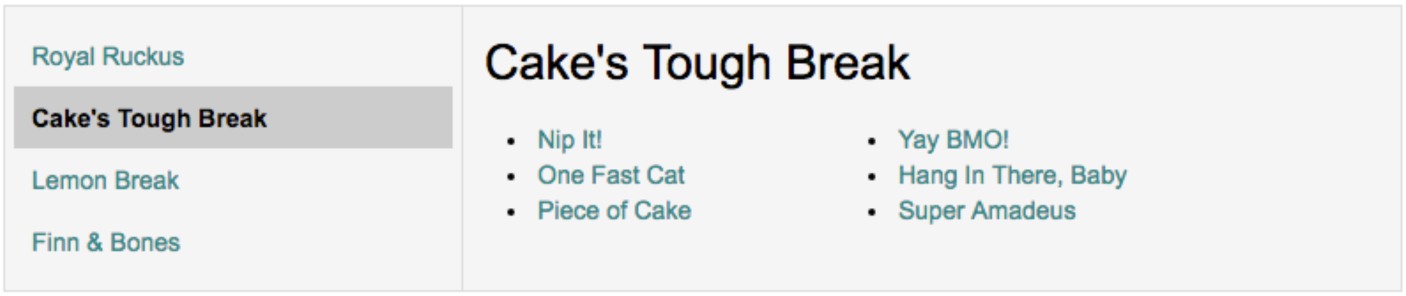
a) Write a function that retrieves the JSON data and returns response data using a callback.
b) Write a function that retrieves the JSON data and returns response data using a promise.

Question 3:

Using your preferred scripting language, create a function that loops through the following array and outputs the resulting HTML. Then, using JavaScript (jQuery is allowed), add the following functionality: clicking one of the games on the left should display the corresponding badges in the panel on the right.

```
var badgeArray = [
    {
        game: "Royal Ruckus",
        badges: ["Approximate Beatdown", "Huge Money", "Taste the Rainbow", "Done & Dungeon", "Let's Rage"]
    },
    {
        game: "Cake's Tough Break",
        badges: ["Nip It!", "Yay BMO!", "One Fast Cat", "Hang In There, Baby", "Piece of Cake", "Super Amadeus"]
    },
    {
        game: "Lemon Break",
        badges: ["Lemon Aid", "Sweet Kicks", "BMO Hope", "Elephant Prowess", "Unacceptable Escape"]
    },
    {
        game: "Finn & Bones",
        badges: ["Rock Family Tree", "Clash of Bones", "Chemistry 101", "Mix Master", "Kiss of Death"]
    }
];
```

Expected output:



Question 4:

Given the following table definition and data, write a SQL statement for each of the tasks below.

```
CREATE TABLE show (
    showid INTEGER PRIMARY KEY,
    showname VARCHAR(64)
);
CREATE TABLE episode (
    titleid INTEGER PRIMARY KEY,
    name VARCHAR(64),
    showid INTEGER,
    active BOOLEAN,
    unlocked BOOLEAN,
    FOREIGN KEY (showid) REFERENCES show (showid)
);
```

show

showid	showname
835928	Gumball
2011404	Steven Universe
2000349	Teen Titans Go

episode

titleid	name	showid	active	unlocked
2127103	At War with Ghosts	835928	TRUE	TRUE
2161404	Doctor Man	835928	TRUE	FALSE
2126651	I Think I Need a Little Change	2011404	TRUE	TRUE
2126687	Onion's Other Friends	2011404	TRUE	FALSE
2126804	Titans vs. Pain Bot	2000349	TRUE	TRUE
2126886	Robin Races Kid Flash	2000349	TRUE	FALSE

a) Disable all episodes from the show "Gumball" by setting their active property to "FALSE".
b) Remove the show "Steven Universe" and its corresponding episodes.
c) Insert the following show and episode records:

```
var newShow = {
    showid: 2012957,
    name: 'Clarence'
};
var newEpisode = {
    titleid: 2124352,
    name: 'Tiny Piggie Hatch',
    showid: 2012957,
    active: true,
    unlocked: true
};
```

d) Create a show called "Unlocked" and add all unlocked episodes to the new show.