[3:50 PM] Aleksandar Aleksiev

Write a JavaScript program to list the properties of a JavaScript object  
    Sample object:  
    var student = {  
    name : "David Rayy",  
    sclass : "VI",  
    rollno : 12 };  
    Sample Output: name,sclass,rollno

[3:50 PM] Aleksandar Aleksiev

   Create TV classes.

   1) Create a TV class with properties like brand, channel and volume.  
        Specify brand in a constructor parameter. Channel should be 1 by default. Volume should be 50 by default.  
    2) Add methods to increase and decrease volume. Volume can't never be below 0 or above 100.  
    3) Add a method to set the channel. Let's say the TV has only 50 channels so if you try to set channel 60 the TV will stay at the current channel.  
    4) Add a method to reset TV so it goes back to channel 1 and volume 50. (Hint: consider using it from the constructor).  
    5) It's useful to write a status, that returns info about the TV status like: "Panasonic at channel 8, volume 75".

[3:50 PM] Aleksandar Aleksiev

за прототипите

[3:50 PM] Aleksandar Aleksiev

/\*

  EXAMPLE TASK:

    - Write an Airplane constructor that initializes `name` from an argument.

    - All airplanes built with Airplane should initialize with an `isFlying` of false.

    - Give airplanes the ability to `.takeOff()` and `.land()`:

        + If a plane takes off, its `isFlying` property is set to true.

        + If a plane lands, its `isFlying` property is set to false.

\*/

/\*

  TASK 1

    - Write a Person Constructor that initializes `name` and `age` from arguments.

    - All instances of Person should initialize with an empty `stomach` array.

    - Give instances of Person the ability to `.eat("someFood")`:

        + When eating an edible, it should be pushed into the `stomach`.

        + The `eat` method should have no effect if there are 10 items in the `stomach`.

    - Give instances of Person the ability to `.poop()`:

        + When an instance poops, its `stomach` should empty.

    - Give instances of Person a method `.toString()`:

        + It should return a string with `name` and `age`. Example: "Mary, 50"

\*/

/\*

  TASK 2

    - Write a Car constructor that initializes `model` and `milesPerGallon` from arguments.

    - All instances built with Car:

        + should initialize with an `tank` at 0

        + should initialize with an `odometer` at 0

    - Give cars the ability to get fueled with a `.fill(gallons)` method. Add the gallons to `tank`.

    - STRETCH: Give cars ability to `.drive(distance)`. The distance driven:

        + Should cause the `odometer` to go up.

        + Should cause the the `tank` to go down taking `milesPerGallon` into account.

    - STRETCH: A car which runs out of `fuel` while driving can't drive any more distance:

        + The `drive` method should return a string "I ran out of fuel at x miles!" x being `odometer`.

\*/

/\*

  TASK 3

    - Write a Baby constructor subclassing Person.

    - Besides `name` and `age`, Baby takes a third argument to initialize `favoriteToy`.

    - Besides the methods on Person.prototype, babies have the ability to `.play()`:

        + Should return a string "Playing with x", x being the favorite toy.

\*/