* Global state of object – access to a public
* Public facing functions
* Private – whatever it interacts with/it doesn’t see it

Object = school / school bus

Tires are properties of school bus

Tread, bolts, weight, air, - all interact w/one another

What is private to the child object = tires

What is public to the parent

School bus doesn’t need to know a lot of stuff – just needs to know how to put this object on myself.

Think of each letter as an object = value, filler, guess

Several methods

* Get character
* Change guess status
* Get filler text

Write function if guess status is true, return character, other return filler = public function

* When create object export out that function

Instance = inherit features/properties from a person

* My own person b/c I have my own properties not shared
* Control how ppl interact with you without giving all access

Object is able to interact with object

* Export function
* Array for letters
* For each letter = public function guessed
* Retrieve underscore or the letter (letter object is where it is defined)

Iterate over array that has letters

Each letter = guess prop

Return character or filler

Managing data.

Has been guessed = public function that checks

Guessed – char update status

Word iterate over every letter – if choice matches – update status

Once status is checked with every letter = relies on underlying state of object

Letters, status, other function return letter or filler

Cannot use THIS in arrow functions – it’s completely anonymous – call it and it’s done.

Single line = return

// dependency for inquirer npm package

var inquirer = require("inquirer");

// constructor function used to create programmers objects

function Programmer(name, position, age, language) {

this.name = name;

this.position = position;

this.age = age;

this.language = language;

// creates the printInfo method and applies it to all programmer objects

this.printInfo = function() {

console.log("Name: " + this.name + "\nPosition: " + this.position +

"\nAge: " + this.age + "\nLanguages: " + this.language);

};

}

// runs inquirer and asks the user a series of questions whose replies are

// stored within the variable answers inside of the .then statement

inquirer.prompt([

{

name: "nameAnswer",

message: "What is your name?"

}, {

name: "positionAnswer",

message: "What is your current position?"

}, {

name: "ageAnswer",

message: "How old are you?"

}, {

name: "languageAnswer",

message: "What is your favorite programming language?"

}

]).then(function(answers) {

// initializes the variable newGuy to be a programmer object which will take

// in all of the user's answers to the questions above

var newGuy = new Programmer(

answers.nameAnswer,

answers.positionAnswer,

answers.ageAnswer,

answers.languageAnswer

);

// printInfo method is run to show that the newguy object was successfully created and filled

newGuy.printInfo();

});