# Javascript Language

**1. Getting Started with Programming (1/28)**

1)"Serhiy"

2)"Serhiy".length; // 6

3) 3+5 // 8 //only 1 operation can be done

5-1

4) 23\*2 //only 1 operation can be done

46/2

5)eggplant //Error: eggplant is not defined

6)// comment

7) confirm //('This is an example of using JS to create some interaction on a website. Click OK to continue!'); подтверждать ( OK) (Отмена) (true)

8) confirm("I feel awesome!"); //User admit something.

confirm("I am ready to go.");

confirm("KO Punch!");

9) prompt("What is your name?"); //Ask input with prompt, запроc “Serhiy”

prompt("What is Ubuntu?");

10) //numbers 123445…

Strings “1” “Serhiy” …

11)// Boolens are true or false. It can be > <.

"I?m coding like a champ".length >10; // true

12)//to print smth. Out.

console.log(2\*5) //10

console.log("Hello") // Hello

13) List of comparison operators:

> Greater than

< Less than

<= Less than or equal to

>= Greater than or equal to

=== Equal to

!== Not equal to

console.log("Xiao Hui".length <= 122); //True

console.log("Goody Donaldson".length !== 8); //True

console.log(8\*2 === 16); //True

14)-16) //If statement

if( "myName".length >= 7 ) {

console.log("You have a long name!");

}

if ("Serhiy".length >=7) {

console.log("You have a long name!"); //True: You have a long name!

}

else {

console.log("You have a short name!"); //False: You have a short name!

}

17) //”debugging” finding mistakes.

19)//modulo is % between numbers.

console.log(14 % 3); //2

console.log(99 % 8); //3

console.log(11 % 3); //2

22)//substring подстрока( 0…n)

"Batman".substring(0,3); // ”Bat”

"laptop".substring(3,6); // ”top”

"wonderful day".substring(3,7); // ”derf”

23)

console.log("January".substring(0,3)); //Jan

console.log("Melbourne is great".substring(0,12)); // Melbourne is

console.log("Hamburgers".substring(3,10)); //burgers

24)//Variable( var varName = data;)

a. var myName = "Serhiy";

b. var myAge = 21;

c. var isOdd = true;

var myAge = 21;

console.log(myAge); //21

var myCountry = "Ukraine";

console.log(myCountry.length); //7

console.log(myCountry.substring(0,3)); //Ukra

**2. Choose Your Own Advanture! (1/7)**

// Check if the user is ready to play!

confirm("I am ready to play!");

var age = prompt("What´s your age ?");

if (age < 13)

{

console.log("You are allowed to play, but administration take no responsibility for the consequences!");

}

else

{

console.log("Start the game!");

}

console.log("You are at a Justin Bieber concert, and you hear this lyric 'Lace my shoes off, start racing.'");

console.log("Suddenly, Bieber stops and says, 'Who wants to race me?'");

var userAnswer = prompt("Do you want to race Bieber on stage?");

console.log("Do you want to race Bieber on stage ?");

if (userAnswer === "yes")

{

console.log("You and Bieber start racing. It's neck and neck! You win by a shoelace!");

}

else

{

console.log("Oh no! Bieber shakes his head and sings 'I set a pace, so I can race without pacing.'");

}

var feedback = prompt("Rate the Game out of 10");

if (feedback > "8")

{

console.log("Thank you! We should race at the next concert!");

}

else

{

console.log("I'll keep practicing coding and racing.");

}

**3. Introduction to Functions in JS (1/13)**

2) // This is what a function looks like:

var divideByThree = function (number) {

var val = number / 3;

console.log(val); //2

};

// On line 12, we call the function by name

// Here, it is called 'dividebythree'

// We tell the computer what the number input is (i.e. 6)

// The computer then runs the code inside the function!

divideByThree(6);

3) //joining strings with +

var greeting = function (name) {

console.log(“Great to see you,” “ + “ “ +name); //Great to see you, Serhiy

};

greeting(“Serhiy”)

6)

var orangeCost = function (number) {

var price = number \* 5;

console.log(price);

};

orangeCost(5);

6)// return function

7) var timesTwo = function(number) {

return number \* 2;

};

// Call timesTwo here!

var newNumber = timesTwo(8);

console.log(newNumber);

8) // Define quarter here.

var quarter = function(number) {

return number / 4;

};

if (quarter(12) % 3 === 0 ) {

console.log("The statement is true");

} else {

console.log("The statement is false");

}

9) var perimeterBox = function(length, width) {

return 2 \* length + 2 \* width; //4

};

var rectangle = perimeterBox(1,1);

console.log(rectangle);

10) var my\_number = 7; //this has global scope

var timesTwo = function(number) {

var my\_number = number \* 2; //local scope

console.log("Inside the function my\_number is: "); // Inside the function my\_number is:

console.log(my\_number); //14

};

timesTwo(7);

console.log("Outside the function my\_number is: ") // Outside the function my\_number is:

console.log(my\_number); //7

11) var nameString = function (name) {

return "Hi, I am" + " " + name; //Hi, I am Susie

console.log(nameString);

};

nameString("Susie");

12) var sleepCheck = function (numHours){

if (numHours >= "8") {

return "You're getting plenty of sleep! Maybe even too much!"

}

else {

return "Get some more shut eye!"

}

};

sleepCheck(10);

sleepCheck(5);

sleepCheck(8); //"You're getting plenty of sleep! Maybe even too much!"

**4. Build “Rock, Paper, Scissors” (1/9)**

1) – 3) var userChoice = prompt("Do you choose rock, paper or scissors?");

var computerChoice = Math.random();

console.log(computerChoice); //0.648766566

4) var userChoice = prompt("Do you choose rock, paper or scissors?");

var computerChoice = Math.random();

console.log(computerChoice);

if(computerChoice <= 0.33) { //. Not , !!!

computerChoice = "rock";

}

else if(computerChoice > 0.33 && computerChoice <= 0.66) {

computerChoice = "paper";

}

else {

computerChoice = "scissors";

}

5)-11) var userChoice = prompt("Do you choose rock, paper or scissors?");

var computerChoice = Math.random();

if (computerChoice < 0.34) {

computerChoice = "rock";

} else if(computerChoice <= 0.67) {

computerChoice = "paper";

} else {

computerChoice = "scissors";

} console.log("Computer: " + computerChoice);

var compare = function (choice1, choice2) {

if (choice1 === choice2) {

return "The result is a tie!";

}

else if (choice1 === "rock") {

if (choice2 === "scissors") {

return "rock wins";

}

else {

return "paper wins";}

}

else if (choice1 === "paper") {

if (choice2 === "rock") {

return "paper wins";

}

else {

return "scissors wins";}

}

else if (choice1 === "scissors") {

if (choice2 === "paper") {

return "scissors wins";

}

else {

return "rock wins";}

}

};

compare(userChoice, computerChoice);

**5. Introduction to ‘For’ Loops in JS (1/13)**

1) for (var counter = 1; counter < 6; counter++) { //1 2 3 4 5

console.log(counter);

}

2) for (var i = 1; i < 11; i = i + 1){

console.log(i);

}

5)//i += x (i +=3) –counts up by 3s // 3 6 9 …

for (var i = 5; i < 51; i +=5) {

console.log(i); //5 10 15 … 50

}

8) for (var i=100; i>0; i -=5) { // 100 … 5

console.log(i);

}

9)// var arrayName = [34, “candy”, “blue”, 11];

var junk = ["Hallo, world", "Javascript", 2, 21];

console.log(junk); //[ ‘Hallo, world’, ‘Javascript’, 2, 21 ]

10) var junkData = ["Eddie Murphy", 49, "peanuts", 31]; //1-st element in array junkData[0] ()

console.log(junkData[3]); //31 position 4 (start from 0)

12) var names = ["Serhiy", "Kirito", "Goku", "Julius", "Christian"];

for (var i = 0; i < names.length; i++) {

console.log("I know someone called " +names[i]);

}

**6. Search Text for Your Name (1/7)**

2)// \ used for avoiding long lines.

var text = "Since text could be quite long Serhiy, you can use a backslash '\' at the end of each line to make Serhiy string 'wrap' to the next line, like this: Serhiy.";

var myName = "Serhiy";

var hits = [];

7) var text = "Since text could be quite long Serhiy, you can use a backslash at the end of each line to make Serhiy string 'wrap' to the next line, like this: Serhiy.";

var myName = "Serhiy";

var hits = [];

for (var i = 0; i < text.length; i++) {

if (text[i] === "S"){

for (var j = i; j < (+myName.length + i); j++) {

hits.push(text[j]);

}

}

}

if (hits.length === 0) {

console.log("Your name wasn't found!");

} else {

console.log(hits);

}

**7. Introduction to ‘While’ Loops in JS (1/11)**

1)// use “while” loop when you don’t know when to stop

var coinFace = Math.floor(Math.random() \* 2);

while(coinFace === 0){

console.log("Heads! Flipping again...");

var coinFace = Math.floor(Math.random() \* 2);

}

console.log("Tails! Done flipping.");

2) // 1 is true 0 is false

var understand = true;

while(understand == 1){ //or just (understand)

console.log("I'm learning while loops!");

understand = false;

}

4) var bool = true; //1-st version

while(bool){

//Do something //bool = false;

}

var bool = true; // 2-d version

while (bool === true){

// Do smth.

}

var bool = 1; //3-d version

while(bool){

//Do smth.

}

5) var count = 0;

var loop = function(){

while(count < 3){

//Your code goes here!

console.log("I'm looping!"); //3x I’m looping!

count++;

}

};

loop();

9)// Do / while to run your loop at least one time.

var loopCondition = false;

do {

console.log("I'm gonna stop looping 'cause my condition is " + loopCondition + "!");

} while (loopCondition);

10) var getToDaChoppa = function(){

// Write your do/while loop here!

do {

console.log("I'll finish my education!");

}while (getToDaChoppa === false);

};

getToDaChoppa();

**8. Dragon Slayer! (1/6)**

1) var slaying = true;

var youHit = Math.floor(Math.random() \* 2); // rounds numbers; random (0 or 1)

var damageThisRound = Math.floor(Math.random() \* 5 +1); //from 0 up till 1; +1(1-5)

var totalDamage = 0;

6) var slaying = true;

var youHit = Math.floor(Math.random() \* 2);

//console.log(youHit);

var damageThisRound = Math.floor(Math.random() \* 5 + 1);

//console.log(damageThisRound);

var totalDamage = 0;

while (slaying) {

if (youHit === 1){

console.log("You hit the dragon and did " + damageThisRound + " damage!");

totalDamage = totalDamage + damageThisRound;

//totalDamage += damageThisRound;

if (totalDamage >= 4) {

console.log("You killed the dragon!");

slaying = false;

}else {

youHit = Math.floor(Math.random() \* 2);

}

}else {

console.log("The dragon defeated you");

slaying = false;

}

}

**9. More on Control Flow in JS (1/14)**

1) var isEven = function(number) {

if (number%2 === 0) {

return true;

}else {

return false;

}

};

isEven (3) //false(if 4 than will be true)

2) var isEven = function(number) {

// Your code goes here!

if (number%2 === 0) {

return true;

}else if(isNaN(number)) {

return "User input isn't a number";

} else {

return false;

}

};

isEven("ssqsd") //User input isn’t a number

4) var lunch = prompt("What do you want for lunch?","Type your lunch choice here");

switch(lunch){

case 'sandwich':

console.log("Sure thing! One sandwich, coming up.");

break;

case 'soup':

console.log("Got it! Tomato's my favorite.");

break;

case 'salad':

console.log("Sounds good! How about a caesar salad?");

break;

case 'pie':

console.log("Pie's not a meal!");

break;

default:

console.log("Huh! I'm not sure what " + lunch + " is. How does a sandwich sound?");

}

10) var iLoveJavaScript = true;

var iLoveLearning = true;

if(iLoveJavaScript && iLoveLearning) {

// if iLoveJavaScript AND iLoveLearning:

console.log("Awesome! Let's keep learning!"); // Awesome! Let's keep learning!

} else if(!(iLoveJavaScript || iLoveLearning)) {

// if NOT iLoveJavaScript OR iLoveLearning:

console.log("Let's see if we can change your mind.");

} else {

console.log("You only like one but not the other? We'll work on it.");

}

11) true && true //true

true && false //false

false && true //false

false && false //false

var hungry = true;

var foodHere = true;

var eat = function() {

// Add your if/else statement here!

if(hungry && foodHere === true) {

return true; // nothing

}else {

return false;

}

};

12) true || true //true

true || false //true

false || true //true

false || false //false

var tired = true;

var bored = false;

var nap = function() {

// Add your if/else statement here!

if(tired || bored) {

return true; // nothing

}else {

return false;

}

};

13) !true //false

!false //true

var programming = false;

var happy = function() {

// Add your if/else statement here!

if (!programming) {

return true;

}else {

return false;

}

};

**10. Choose your own advanture2! (1/6)**

3) .toUpperCase() //checks for all capital letters

.toLowerCase() //for all lower case letters

5) var answer = prompt("Are you ready to fight against Machamp.(yes, no, maybe) ?").toLowerCase();

switch(answer) {

case "yes":

console.log("Your fight begins in 3-2-1. Go!");

var psychic = prompt("Should you use psychik ?");

var confusion = prompt("Should you use confusion ?");

if (psychic === "yes" && confusion === "yes") {

console.log("You have made two times double damage, couse psychik and confusion makes x2 against fight type!");

}else{

console.log("Your attack was poor and Machamp ran away.");

}

break;

case "no":

console.log("You began running and got lost in the wood.");

break;

case "maybe":

console.log("As you were thinking the enemy began to atteck you.");

break;

default :

console.log("The enemy has waited to long. You were punched in the face.")

};

6) var answer = prompt("Are you ready to fight against Machamp.(yes, no, maybe) ?").toLowerCase();

switch(answer) {

case "yes":

console.log("Your fight begins in 3-2-1. Go!");

var psychic = prompt("Should you use psychik ?");

var confusion = prompt("Should you use confusion ?");

if (psychic === "yes" && confusion === "yes") {

console.log("You have made two times double damage, couse psychik and confusion makes x2 against fight type!");

}else{

console.log("Your attack was poor and Machamp ran away.");

}

break;

case "no":

console.log("You began running and got lost in the wood.");

break;

case "maybe":

console.log("As you were thinking the enemy began to atteck you.");

var des1 = prompt("Are you fighting ?");

var des2 = prompt("Are you running ?");

if (des1 === "Yes" || des2 === "No") {

console.log("Come back soon!");

} else {

console.log("Great! Let's do it!");

}

break;

default :

console.log("The enemy has waited to long. You were punched in the face.")

};

**11. Arrays and Objects in JS (1/17)**

1) var list = [14, "Hallo", 2];

2) var languages = ["HTML", "CSS", "JavaScript", "Python", "Ruby"];

console.log(languages[2]);

3) var languages = ["HTML", "CSS", "JavaScript", "Python", "Ruby"];

// 0 1 2 3 4

console.log(languages.length);

4) var languages = ["HTML", "CSS", "JavaScript", "Python", "Ruby"];

for (i = 0; i < languages.length; i++) {

console.log(languages[i]);

}

5) var myArray = [12, true, "Schiff"]; //heterogeneous array

6)// var twoDimensional = [[1,1], [1,1]];

var newArray = [[0,0,0] , [0,1,0] , [1,0,0]];

7) var jagged = [[1,0], [5], [1,0,5]]; //jagged array

8) var phonebookEntry = {}; //Object

phonebookEntry.name = 'Oxnard Montalvo'; //Keys

phonebookEntry.number = '(555) 555-5555';

phonebookEntry.phone = function() {

console.log('Calling ' + this.name + ' at ' + this.number + '...');

};

phonebookEntry.phone(); //Calling Oxnard Montalvo at (555) 555-5555…

10) var me = {}; //object constructor

me.name = "Serhiy";

me.age = 21

me.biographie = function() {

console.log("name: " + this.name +" and age: " + this.age);

};

me.biographie();

//OR

var me = { //object literal notation

name: "Serhiy",

age: 21

};

11) var me = new Object();

me.name = "Serhiy"; //me["name"] = "Serhiy";

me.age = 21 //me["age"] = 21

14) var myArray = [1 , false , "Hallo" , myObj , "Yeah!" , true];

var myObj = {}; //var myObj = new Object();

15) var newArray = [[1,0],[2,0,8,me]];

var me = Object();

16) var myObject = {

name: 'Eduardo',

type: 'Most excellent',

// Add your code here!

interests: ['Sports' , 'Reading']

};

myObject.biography = function() {

console.log("Information: " + this.name + " " + this.type + " " + this.interests);

};

myObject.biography();

17) var myOwnObject = Object();

myOwnObject.name = "Game";

**12. Contact List (1/8)**

8) var friends = {};

friends.martin = {

firstName: "Martin",

lastName: "Smith",

number: "097 9474316",

address: ['Cambridge Avenu','Chicago','WA','98052']

};

friends.steve = {

firstName: "Steve",

lastName: "Jobs",

number: "095 2345533",

address: ['Apple','NY','WA','98052']

};

friends.bill = {

firstName: "Bill",

lastName: "Gates",

number: "(206) 555-5555",

address: ['One Microsoft Way','Redmond','WA','98052']

};

friends.mark = {

firstName: "Mark",

lastName: "Tuskler",

number: "(311) 333-3333",

address: ['Bomsch','Los Angeles','WA','98052']

};

var list = function (friends) {

for (var prop in friends) {

console.log(prop);

}

}

var search = function(name) {

for(var prop in friends) {

if(friends[prop].firstName === name) {

console.log(friends[prop]);

return friends[prop];

}

}

};

list(friends);

search ("Steve");

**13. Introduction to Objects I (1/33)**

1) (((3 \* 90) === 270) || !(false && (!false)) || "bex".toUpperCase() === "BEX");

var answer = true; //true

2) var multiplesOfEight = [8,16,24,32,40,58];

var answer = multiplesOfEight[5] % 8 !== 0;

3) for (i=1; i<=20; i++) {

if (i%3 === 0 && i%5 === 0) {

console.log("FizzBuzz");

}

else if (i%3 === 0) {

console.log("Fizz");

}

else if (i%5 === 0) {

console.log("Buzz");

}

else {

console.log(i);

}

}

4) var getReview = function (movie) {

switch(movie) {

case "Toy Story 2":

return ("Great story. Mean prospector.");

case "Finding Nemo":

return ("Cool animation, and funny turtles.");

case "The Lion King":

return ("Great songs.");

default:

return ("I don't know!");

}

};

getReview("Finding Nemo");

5) var bob = {

name: "Bob Smith",

age: 30

};

var susan = {

name: "Susan Jordan",

age: 25

};

// here we save Bob's information

var name1 = bob.name;

var age1 = bob.age;

// finish this code by saving Susan's information

var name2 = susan.name;

var age2 = susan.age;

8) var bob = {

name: "Bob Smith",

age: 30

};

var susan = {

name: "Susan Jordan",

age: 25

};

// here we save Bob's information

var name1 = bob.name;

var age1 = bob.age;

// finish this code by saving Susan's information

var name2 = susan.name;

var age2 = susan.age;

9) // Take a look at our next example object, a dog

var dog = {

species: "greyhound",

weight: 60,

age: 4

};

var species = dog["species"];

// fill in the code to save the weight and age using bracket notation

var weight = dog["weight"];

var age = dog["age"];

11) // help us make snoopy using literal notation

// Remember snoopy is a "beagle" and is 10 years old.

var snoopy = {

species: "beagle",

age: 10

};

// help make buddy using constructor notation

// buddy is a "golden retriever" and is 5 years old

var buddy = new Object();

buddy.species = "golden retriever";

buddy.age = 5;

12) // Accepts a number x as input and returns its square

var square = function (x) {

return x \* x;

};

// It should take two parameters and return the product

var multiply = function (x,y) {

return x \* y;

};

multiply(2,3);

14) // here is bob again, with his usual properties

var bob = new Object();

bob.name = "Bob Smith";

bob.age = 30;

// this time we have added a method, setAge

bob.setAge = function (newAge){

bob.age = newAge;

};

// here we set bob's age to 40

bob.setAge(40);

// bob's feeling old. Use our method to set bob's age to 20

bob.setAge(20);

15) var bob = new Object();

bob.age = 17;

// this time we have added a method, setAge

bob.setAge = function (newAge){

bob.age = newAge;

};

bob.getYearOfBirth = function () {

return 2014 - bob.age;

};

console.log(bob.getYearOfBirth());

16) // here we define our method using "this", before we even introduce bob

var setAge = function (newAge) {

this.age = newAge;

};

// now we make bob

var bob = new Object();

bob.age = 30;

// and down here we just use the method we already made

bob.setAge = setAge;

// change bob's age to 50 here

bob.setAge(50);

17) // here we define our method using "this", before we even introduce bob

var setAge = function (newAge) {

this.age = newAge;

};

// now we make bob

var bob = new Object();

bob.age = 30;

bob.setAge = setAge;

// make susan here, and first give her an age of 25

var susan = new Object();

susan.age = 25;

susan.setAge = setAge;

// here, update Susan's age to 35 using the method

susan.setAge(35);

18) var rectangle = new Object();

rectangle.height = 3;

rectangle.width = 4;

// here is our method to set the height

rectangle.setHeight = function (newHeight) {

this.height = newHeight;

};

// help by finishing this method

rectangle.setWidth = function (newWidth) {

this.width = newWidth;

};

// here change the width to 8 and height to 6 using our new methods

rectangle.setHeight(6);

rectangle.setWidth(8);

19) var square = new Object();

square.sideLength = 6;

square.calcPerimeter = function() {

return this.sideLength \* 4;

};

// help us define an area method here

square.calcArea = function() {

return this.sideLength \* this.sideLength

};

var p = square.calcPerimeter();

var a = square.calcArea();

console.log(p); //24

console.log(a); //36

20) function Person(name,age) {

this.name = name;

this.age = age;

}

// Let's make bob and susan again, using our constructor

var bob = new Person("Bob Smith", 30);

var susan = new Person("Susan Jordan", 25);

// help us make george, whose name is "George Washington" and age is 275

var george = new Person("George Washington", 275);

22) function Cat(age, color) {

this.age = age;

this.color = color;

}

// make a Dog constructor here

function Dog(age, name, breed) {

this.age = age;

this.name = name;

this.breed = breed;

}

var Chappie = new Dog(22, "Chappie", "Pitbull");

console.log(Chappie);

23) function Person(name,age) {

this.name = name;

this.age = age;

this.species = "Homo Sapiens";

}

var sally = new Person("Sally Bowles", 39);

var holden = new Person("Holden Caulfield", 16);

console.log(sally.name + " is pritty nice girl."+" Sally's species is " + sally.species + " and she is " + sally.age + "." );

console.log(holden.name + " is a cool guy. " + "Holden's species is " + holden.species + " and he is " + holden.age + ".");

24) function Rectangle(height, width) {

this.height = height;

this.width = width;

this.calcArea = function() {

return this.height \* this.width;

};

// put our perimeter function here!

this.calcPerimeter = function() {

return 2 \* this.height + 2 \* this.width

};

}

var rex = new Rectangle(7,3);

var area = rex.calcArea();

var perimeter = rex.calcPerimeter();

25) function Rabbit(adjective) {

this.adjective = adjective;

this.describeMyself = function() {

console.log("I am a " + this.adjective + " rabbit");

};

}

// now we can easily make all of our rabbits

var rabbit1 = new Rabbit("fluffy");

var rabbit2 = new Rabbit("happy");

var rabbit3 = new Rabbit("sleepy");

rabbit1.describeMyself();

rabbit2.describeMyself();

rabbit3.describeMyself();

26) // Our person constructor

function Person (name, age) {

this.name = name;

this.age = age;

}

// Now we can make an array of people

var family = new Array();

family[0] = new Person("alice", 40);

family[1] = new Person("bob", 42);

family[2] = new Person("michelle", 8);

// add the last family member, "timmy", who is 6 years old

family[3] = new Person("timmy", 6);

27) // Our Person constructor

function Person(name, age) {

this.name = name;

this.age = age;

}

// Now we can make an array of people

var family = [];

family[0] = new Person("alice", 40);

family[1] = new Person("bob", 42);

family[2] = new Person("michelle", 8);

family[3] = new Person("timmy", 6);

// loop through our new array

for (i=0; i<family.length; i++) {

console.log(family[i].name);

};

28) // Our person constructor

function Person (name, age) {

this.name = name;

this.age = age;

}

// We can make a function which takes persons as arguments

// This one computes the difference in ages between two people

var ageDifference = function(person1, person2) {

return person1.age - person2.age;

}

var alice = new Person("Alice", 30);

var billy = new Person("Billy", 25);

// get the difference in age between alice and billy using our function

var diff = ageDifference(alice, billy);

console.log(diff);

29) // Our person constructor

function Person (name, age) {

this.name = name;

this.age = age;

}

// We can make a function which takes persons as arguments

// This one computes the difference in ages between two people

var ageDifference = function(person1, person2) {

return person1.age - person2.age;

};

// Make a new function, olderAge, to return the age of

// the older of two people

var olderAge = function(person1, person2){

if (person1.age > person2.age) {

return person1.age;

}else {

return person2.age;

}

};

// Let's bring back alice and billy to test our new function

var alice = new Person("Alice", 30);

var billy = new Person("Billy", 25);

console.log("The older person is " + olderAge(alice, billy));

30) var spencer = {

age: 22,

country: "United States"

};

// make spencer2 here with constructor notation

var spencer2 = new Object();

spencer2.age = 22;

spencer2.country = "United States";

31) var snoopy = new Object();

snoopy.species = "beagle";

snoopy.age = 10;

// save Snoopy's age and species into variables

// use dot notation for snoopy's species

var species = snoopy.species;

// use bracket notation for snoopy's age

var age = snoopy["age"];

32) // 3 lines required to make harry\_potter

var harry\_potter = new Object();

harry\_potter.pages = 350;

harry\_potter.author = "J.K. Rowling";

// A custom constructor for book

function Book (pages, author) {

this.pages = pages;

this.author = author;

}

// Use our new constructor to make the\_hobbit in one line

var the\_hobbit = new Book(320, "J.R.R. Tolkien");

console.log(The\_hobbit);

33) function Circle (radius) {

this.radius = radius;

this.area = function () {

return Math.PI \* this.radius \* this.radius;

};

// define a perimeter method here

this.perimeter = function() {

return 2 \* Math.PI \* this.radius;

}

};

**14. Building an Address Book (1/6)**

6) var bob = {

firstName: "Bob",

lastName: "Jones",

phoneNumber: "(650) 777-7777",

email: "bob.jones@example.com"

};

var mary = {

firstName: "Mary",

lastName: "Johnson",

phoneNumber: "(650) 888-8888",

email: "mary.johnson@example.com"

};

var contacts = [bob, mary];

function printPerson(person) {

console.log(person.firstName + " " + person.lastName);

}

function list() {

var contactsLength = contacts.length;

for (var i = 0; i < contactsLength; i++) {

printPerson(contacts[i]);

}

}

/\*Create a search function

then call it passing "Jones"\*/

function search(lastName) {

for (var i = 0; i < contacts.length; i++) {

if (contacts[i].lastName === lastName) {

printPerson(contacts[i]);

}

}

};

search("Jones");

function add(firstName, lastName, email, phoneNumber){

contacts[contacts.length]={

firstName: firstName,

lastName: lastName,

phoneNumber: phoneNumber,

email: email

}

};

var mike = new add("Mike", "Brown", "mike.brown@example.com", "(650) 999-9999");

list();

**15. Introduction to Objects II (1/30)**

1) var james = {

// add properties to this object!

job: "programmer",

married: false

};

function Person(job, married) {

this.job = job;

this.married = married;

}

// create a "gabby" object using the Person constructor!

var gabby = new Person("student", true);

2) function Person(job, married) {

this.job = job;

this.married = married;

// add a "speak" method to Person!

this.speak = function() {

console.log("Hello!");

}

}

var user = new Person("Codecademy Student",false);

user.speak();

3) var james = {

job: "programmer",

married: false,

speak: function(mood) {

console.log("Hello, I am feeling " + mood);

}

};

james.speak("great");

james.speak("just okay");

4) var james = {

job: "programmer",

married: false,

sayJob: function() {

// complete this method

console.log("Hi, I work as a "+ this.job);

}

};

// james' first job

james.sayJob();

// change james' job to "super programmer" here

james.job = "super programmer";

// james' second job

james.sayJob();

5) var james = {

job: "programmer",

married: false

};

// set to the first property name of "james"

var aProperty = "job";

// print the value of the first property of "james"

// using the variable "aProperty"

console.log(james[aProperty]);

6) var anObj = { job: "I'm an object!" };

var aNumber = 42;

var aString = "I'm a string!";

console.log(typeof anObj); // should print "object"

console.log(typeof aNumber); // should print "number"

console.log(typeof aString); // should print "string"

7) var myObj = {

// finish myObj

name: ""

};

console.log( myObj.hasOwnProperty('name') ); // should print true

console.log( myObj.hasOwnProperty('nickname') ); // should print false

8) var suitcase = {

shirt: "Hawaiian"

};

if (suitcase.hasOwnProperty("shorts") === true) {

console.log(suitcase.shorts);

}else {

suitcase.shorts = "Venum"; //Venum

console.log(suitcase.shorts);

}

9) var nyc = {

fullName: "New York City",

mayor: "Bill de Blasio",

population: 8000000,

boroughs: 5

};

for (var list in nyc) {

console.log(list); //fullName mayor population boroughts

}

10) var nyc = {

fullName: "New York City",

mayor: "Bill de Blasio",

population: 8000000,

boroughs: 5

};

// write a for-in loop to print the value of nyc's properties

for (var x in nyc) {

console.log(nyc[x]); //New York City Bill de Blasio 80000000 5

}

11) function Person(name,age) { //class

this.name = name;

this.age = age;

}

// Let's make bob again, using our constructor

var bob = new Person("Bob Smith", 30);

var susan = new Person("Susan Jordan", 35);

// make your own class here

function Circle(radius) {

this.radius = radius;

}

var firstCircle = new Circle(3);

12) function Dog (breed) {

this.breed = breed;

}

// here we make buddy and teach him how to bark

var buddy = new Dog("Golden Retriever");

buddy.bark = function() {

console.log("Woof");

};

buddy.bark();

// here we make snoopy

var snoopy = new Dog("Beagle");

// we need you to teach snoopy how to bark here

snoopy.bark = function() {

console.log("bark");

};

// this causes an error, because snoopy doesn't know how to bark!

snoopy.bark();

13) function Person(name,age) {

this.name = name;

this.age = age;

}

// a function that prints the name of any given person

var printPersonName = function (p) {

console.log(p.name);

};

var bob = new Person("Bob Smith", 30);

printPersonName(bob);

// make a person called me with your name and age

// then use printPersonName to print your name

var me = new Person("Serhiy Bolkun", 21);

printPersonName(me);

14) function Dog (breed) {

this.breed = breed;

};

// here we make buddy and teach him how to bark

var buddy = new Dog("golden Retriever");

Dog.prototype.bark = function() { //helps functions to do for all objects

console.log("Woof");

};

buddy.bark();

// here we make snoopy

var snoopy = new Dog("Beagle");

/// this time it works!

snoopy.bark();

15) function Cat(name, breed) {

this.name = name;

this.breed = breed;

}

// let's make some cats!

var cheshire = new Cat("Cheshire Cat", "British Shorthair");

var gary = new Cat("Gary", "Domestic Shorthair");

// add a method "meow" to the Cat class that will allow

// all cats to print "Meow!" to the console

Cat.prototype.meow = function() {

console.log("Meow!");

}

// add code here to make the cats meow!

cheshire.meow();

gary.meow();

16) // create your Animal class here

function Animal(name, numLegs){

this.name = name;

this.numLegs = numLegs;

}

// create the sayName method for Animal

Animal.prototype.sayName = function() {

console.log("Hi my name is " +this.name);

}

// provided code to test above constructor and method

var penguin = new Animal("Captain Cook", 2);

penguin.sayName();

17) function Animal(name, numLegs) {

this.name = name;

this.numLegs = numLegs;

}

Animal.prototype.sayName = function() {

console.log("Hi my name is " + this.name);

};

// create a Penguin constructor here

function Penguin(name, numLegs) {

this.name = name;

this.numLegs = numLegs;

}

// create a sayName method for Penguins here

Penguin.prototype.sayName = function() {

console.log("Hi my name is " + this.name);

}

// our test code

var theCaptain = new Penguin("Captain Cook", 2);

theCaptain.sayName();

18) // the original Animal class and sayName method

function Animal(name, numLegs) {

this.name = name;

this.numLegs = numLegs;

}

Animal.prototype.sayName = function() {

console.log("Hi my name is " + this.name);

};

// define a Penguin class

function Penguin(name, numLegs) {

this.name = name;

this.numLegs = 2;

}

// set its prototype to be a new instance of Animal

Penguin.prototype = new Animal();

19) // the original Animal class and sayName method

function Animal(name, numLegs) {

this.name = name;

this.numLegs = numLegs;

}

Animal.prototype.sayName = function() {

console.log("Hi my name is " + this.name);

};

// define a Penguin class

function Penguin(name, numLegs) {

this.name = name;

this.numLegs = 2;

}

// set its prototype to be a new instance of Animal

Penguin.prototype = new Animal();

var penguin = new Penguin("Charly");

penguin.sayName();

20) function Penguin(name) {

this.name = name;

this.numLegs = 2;

}

// create your Emperor class here and make it inherit from Penguin

function Emperor(name) {

this.name = name;

}

Emperor.prototype = new Penguin();

// create an "emperor" object and print the number of legs it has

var emperor = new Emperor("Sussy");

console.log(emperor.numLegs); //2

21) // original classes

function Animal(name, numLegs) {

this.name = name;

this.numLegs = numLegs;

this.isAlive = true;

}

function Penguin(name) {

this.name = name;

this.numLegs = 2;

}

function Emperor(name) {

this.name = name;

this.saying = "Waddle waddle";

}

// set up the prototype chain

Penguin.prototype = new Animal();

Emperor.prototype = new Penguin();

var myEmperor = new Emperor("Jules");

console.log( myEmperor.saying); // should print "Waddle waddle"

console.log( myEmperor.numLegs); // should print 2

console.log( myEmperor.isAlive); // should print true

22) function Person(first,last,age) {

this.firstName = first;

this.lastName = last;

this.age = age;

}

var john = new Person('John','Smith',30);

var myFirst = john.firstName;

var myLast = john.lastName;

//declare variable myAge set to the age of the john object.

var myAge = john.age;

23) function Person(first,last,age) {

this.firstname = first;

this.lastname = last;

this.age = age;

var bankBalance = 7500;

}

// create your Person

var john = new Person("John", "Armstrong", 32);

// try to print his bankBalance

console.log(john.bankBalance); //undefined

24) function Person(first,last,age) {

this.firstname = first;

this.lastname = last;

this.age = age;

var bankBalance = 7500;

this.getBalance = function() {

// your code should return the bankBalance

return bankBalance;

};

}

var john = new Person('John','Smith',30);

console.log(john.bankBalance); //undefined

// create a new variable myBalance that calls getBalance()

var myBalance = john.getBalance();

console.log(myBalance); //7500

25) function Person(first,last,age) {

this.firstname = first;

this.lastname = last;

this.age = age;

var bankBalance = 7500;

var returnBalance = function() {

return bankBalance;

};

// create the new function here

this.askTeller = function() {

return returnBalance;

}

}

var john = new Person('John','Smith',30);

console.log(john.returnBalance); //undefined

var myBalanceMethod = john.askTeller();

var myBalance = myBalanceMethod();

console.log(myBalance); //7500

26) function Person(first,last,age) {

this.firstname = first;

this.lastname = last;

this.age = age;

var bankBalance = 7500;

this.askTeller = function(pass) {

if (pass == 1234) return bankBalance;

else return "Wrong password.";

};

}

var john = new Person('John','Smith',30);

/\* the variable myBalance should access askTeller()

with a password as an argument \*/

var myBalance = john.askTeller(1234);

27) var languages = {

english: "Hello!",

french: "Bonjour!",

notALanguage: 4,

spanish: "Hola!"

};

// print hello in the 3 different languages

for (var x in languages) {

if(typeof languages[x] === "string") {

console.log(languages[x]);

}

}

28) function Dog (breed) {

this.breed = breed;

};

// add the sayHello method to the Dog class

// so all dogs now can say hello

Dog.prototype.sayHello = function(){

console.log("Hello this is a " +this.breed +" dog");

}

var yourDog = new Dog("golden retriever");

yourDog.sayHello();

var myDog = new Dog("dachshund");

myDog.sayHello();

29) // what is this "Object.prototype" anyway...?

var prototypeType = typeof Object.prototype;

console.log(prototypeType);

// now let's examine it!

var hasOwn = Object.prototype.hasOwnProperty("hasOwnProperty");

console.log(hasOwn);

30) function StudentReport() {

var grade1 = 4;

var grade2 = 2;

var grade3 = 1;

this.getGPA = function() {

return (grade1 + grade2 + grade3) / 3;

};

}

var myStudentReport = new StudentReport();

for(var x in myStudentReport) {

if(typeof myStudentReport[x] !== "function") {

console.log("Muahaha! " + myStudentReport[x]);

}

}

console.log("Your overall GPA is " + myStudentReport.getGPA()); //Your overall GPA is 2,3333

**16. Building a Cash Register (1/7)**

2) var cashRegister = {

total:0,

add: function(itemCost){

this.total += itemCost;//this.total = this.total + itemCost;

}

};

//call the add method for our items

cashRegister.add(0.98);

cashRegister.add(1.23);

cashRegister.add(4.99);

cashRegister.add(0.45);

//Show the total bill

console.log('Your bill is '+cashRegister.total);

3) var cashRegister = {

total: 0,

//insert the add method here

add: function(itemCost) {

this.total += itemCost;

},

scan: function (item) {

switch (item) {

case "eggs":

this.add(0.98);

break;

case "milk":

this.add(1.23);

break;

case "magazine":

this.add(4.99);

break;

case "chocolate":

this.add(0.45);

break;

}

return true;

}

};

//Scan 2 eggs and 3 magazines

cashRegister.scan("eggs");

cashRegister.scan("eggs");

cashRegister.scan("magazine");

cashRegister.scan("magazine");

cashRegister.scan("magazine");

//Show the total bill

console.log('Your bill is '+cashRegister.total);

4) var cashRegister = {

total:0,

add: function(itemCost){

this.total += itemCost;

},

scan: function(item, quantity) {

switch (item) {

case "eggs": this.add(0.98 \* quantity); break;

case "milk": this.add(1.23 \* quantity); break;

case "magazine": this.add(4.99 \* quantity); break;

case "chocolate": this.add(0.45 \* quantity); break;

}

}

};

// scan each item 4 times

cashRegister.scan("eggs", 4);

cashRegister.scan("milk", 4);

cashRegister.scan("magazine", 4);

cashRegister.scan("chocolate", 4);

//Show the total bill

console.log('Your bill is '+cashRegister.total);

5) var cashRegister = {

total:0,

lastTransactionAmount: 0,

//Dont forget to add your property

add: function(itemCost) {

this.total += itemCost;

this.lastTransactionAmount = itemCost;

},

scan: function(item, quantity) {

switch (item) {

case "eggs": this.add(0.98 \* quantity); break;

case "milk": this.add(1.23 \* quantity); break;

case "magazine": this.add(4.99 \* quantity); break;

case "chocolate": this.add(0.45 \* quantity); break;

}

return true;

},

//Add the voidLastTransaction Method here

voidLastTransaction: function() {

this.total -= this.lastTransactionAmount;

}

};

cashRegister.scan('eggs',1);

cashRegister.scan('milk',1);

cashRegister.scan('magazine',1);

cashRegister.scan('chocolate',4);

//Void the last transaction and then add 3 instead

cashRegister.voidLastTransaction();

cashRegister.scan("chocolate",3);

//Show the total bill

console.log('Your bill is '+cashRegister.total);

6) // create a constructor for the StaffMember class

function StaffMember(name, discountPercent){

this.name = name;

this.discountPercent =discountPercent;

}

var sally = new StaffMember("Sally",5);

var bob = new StaffMember("Bob",10);

//Create a StaffMember for yourself called me

var me = new StaffMember("Serhiy", 20);

7) function StaffMember(name,discountPercent){

this.name = name;

this.discountPercent = discountPercent;

}

var sally = new StaffMember("Sally",5);

var bob = new StaffMember("Bob",10);

// Create yourself again as 'me' with a staff discount of 20%

var me = new StaffMember("Serhiy", 20);

var cashRegister = {

total:0,

lastTransactionAmount: 0,

add: function(itemCost){

this.total += (itemCost || 0);

this.lastTransactionAmount = itemCost;

},

scan: function(item,quantity){

switch (item){

case "eggs": this.add(0.98 \* quantity); break;

case "milk": this.add(1.23 \* quantity); break;

case "magazine": this.add(4.99 \* quantity); break;

case "chocolate": this.add(0.45 \* quantity); break;

}

return true;

},

voidLastTransaction : function(){

this.total -= this.lastTransactionAmount;

this.lastTransactionAmount = 0;

},

// Create a new method applyStaffDiscount here

applyStaffDiscount : function(employee){

this.total -= this.total \* (employee.discountPercent / 100);

}

};

cashRegister.scan('eggs',1);

cashRegister.scan('milk',1);

cashRegister.scan('magazine',3);

// Apply your staff discount by passing the 'me' object

// to applyStaffDiscount

cashRegister.applyStaffDiscount(this.me);

// Show the total bill

console.log('Your bill is '+cashRegister.total.toFixed(2));