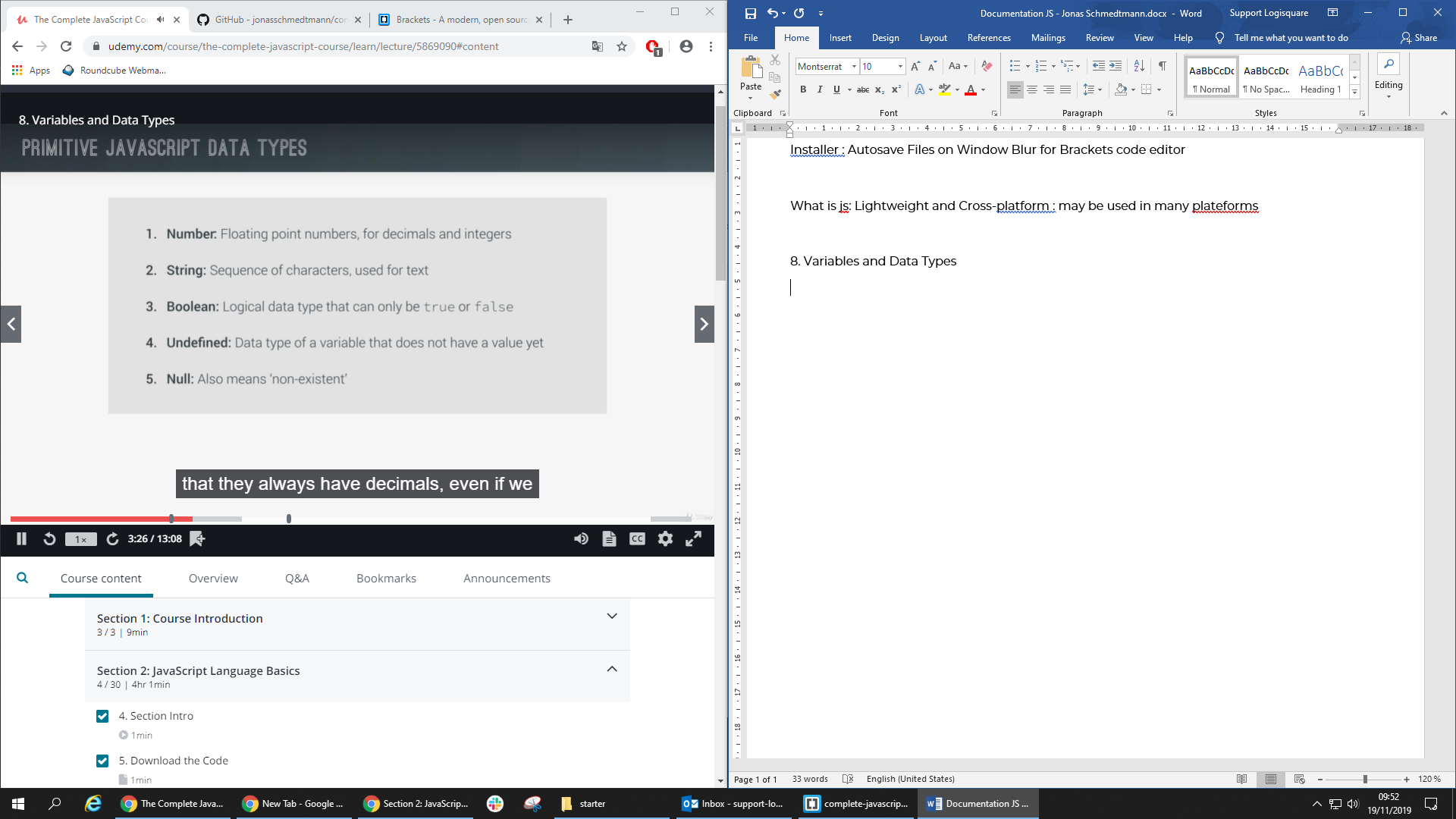
Section 1: Course Introduction

Installer : Autosave Files on Window Blur for Brackets code editor

What is js: Lightweight and Cross-platform : may be used in many plateforms

8. Variables and Data Types



All numbers are floating number even if they are integer number: 5 = 5.0

Javascript has dynamic typing : data types are automatically assigned to variables

Boolean are converted to string

* Console.log(false) 🡺 false

10. Basic Operators

var johnOlder = ageJohn < ageMark;

// typeof operator

console.log(typeof johnOlder); 🡺 boolean

16. The Ternary Operator

var firstName = 'John';

var age = 14;

// Ternary operator

age >= 18 ? console.log(firstName + ' drinks beer.') : console.log(firstName + ' drinks juice.');

17. Truthy and Falsy values and equality operators

// falsy values: undefined, null, 0, '', NaN

// truthy values: NOT falsy values

21. Function Statements and Expressions

var whatDoYouDo = function(job, firstName) {

switch(job) {

case 'teacher':

return firstName + ' teaches kids how to code';

case 'driver':

return firstName + ' drives a cab in Lisbon.'

case 'designer':

return firstName + ' designs beautiful websites';

default:

return firstName + ' does something else';

}

}

console.log(whatDoYouDo('teacher', 'John'));

22. Arrays (=== -1 : check if not exists)

var john = ['John', 'Smith', 1990, 'designer', false];

var isDesigner = john.indexOf('designer') **=== -1** ? 'John is NOT a designer' : 'John IS a designer';

console.log(isDesigner);

24. CODING CHALLENGE 3

number \* 20% === number \* . 2

25. Objects and Properties

Two way to initialize an object:

var mohamed = {

firstName: 'Mohamed',

lastName: 'CHIBANI',

birthYear: 1996,

family: ['Am', 'Za', 'Ma', 'Ba'],

job: 'developer',

isMarried : false

}

mohamed.job = 'enginner';

mohamed['isMarried'] = true;

console.log(mohamed);

var amine = new Object();

amine.firstName = 'Amine';

amine.birthYear = 2003;

amine['lastName'] = 'CHIBANI';

console.log(amine);

31-32 : Challenge

var john = {

fullName : 'John Smith',

bills : [124,48,268,180,42],

calcTip: function(){

this.tips = [];

this.finalValues = [];

var percentage;

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

bill = this.bills[i];

if(bill < 50){

percentage = .2;

}else if(bill >= 50 && bill <= 200){

percentage = .15;

}else if(bill > 200){

percentage = .1;

}

this.tips[i] = bill \* percentage;

this.finalValues[i] = bill + this.tips[i];

}

}

}

var mark = {

fullName : 'Mark Miller',

bills : [77,375,110,45],

calcTip: function(){

this.tips = [];

this.finalValues = [];

var percentage;

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

bill = this.bills[i];

if(bill < 100){

percentage = .2;

}else if(bill >= 100 && bill <= 300){

percentage = .1;

}else if(bill > 300){

percentage = .25;

}

this.tips[i] = bill \* percentage;

this.finalValues[i] = bill + this.tips[i];

}

}

}

john.calcTip();

mark.calcTip();

function calcAverage(tips){

var sum = 0;

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

sum = sum + tips[i];

}

return sum / tips.length;

}

john.average = calcAverage(john.tips);

mark.average = calcAverage(mark.tips);

console.log(john,mark);

if(john.average > mark.average){

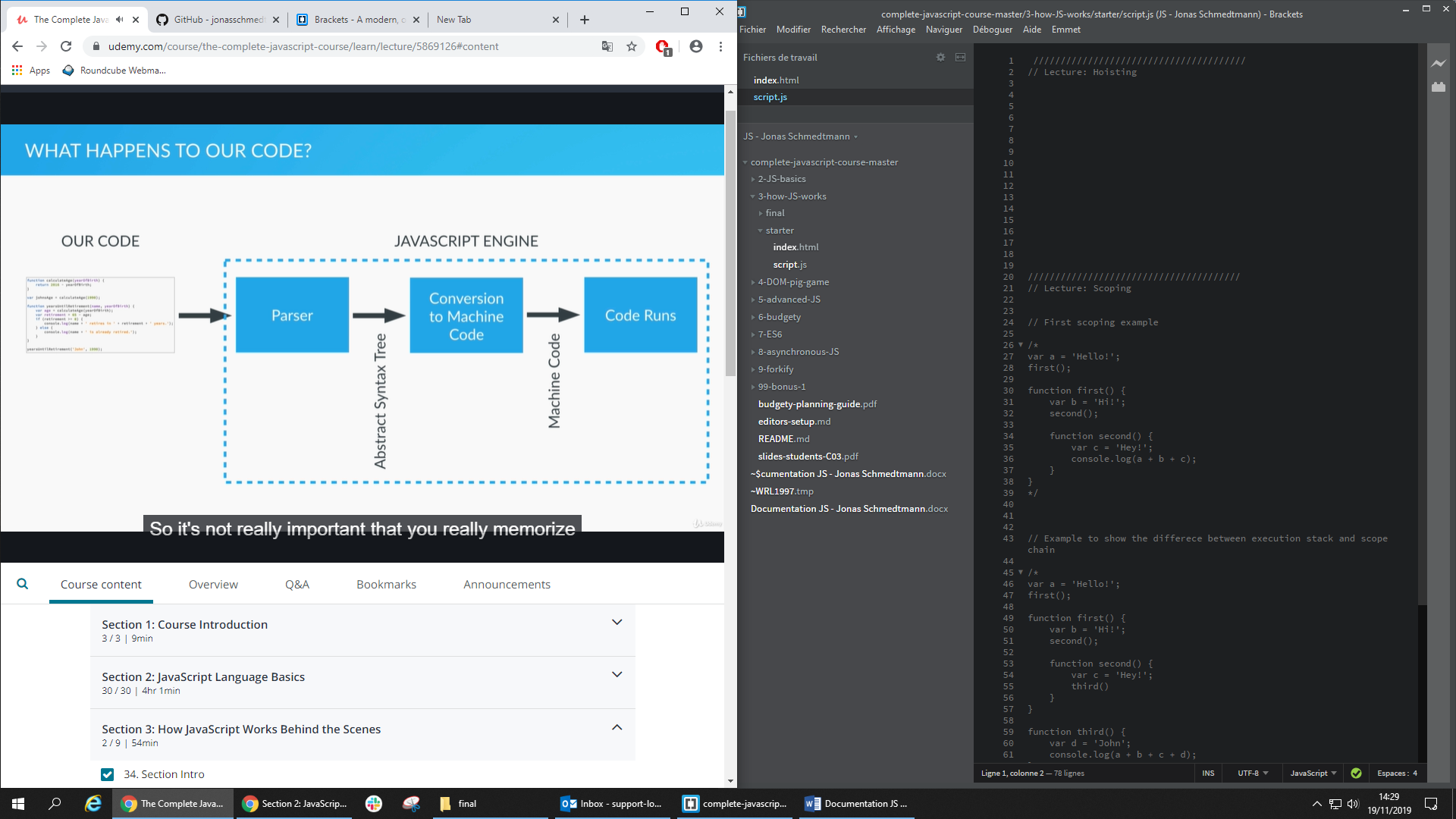
console.log(john.fullName+'s\' family paid the highest tips on average with '+ john.average);

}else{

console.log(mark.fullName+'s\' family paid the highest tips on average with '+ mark.average);

}

Section 3: How JavaScript Works Behind the Scenes



Hoasting only works with function declaration

Hoisting works only with function declarations:

calculateAge(1996);

function calculateAge(year){

console.log(2019 - year);

}

Hoisting doesn’t work with function expression:

retirement(1996);

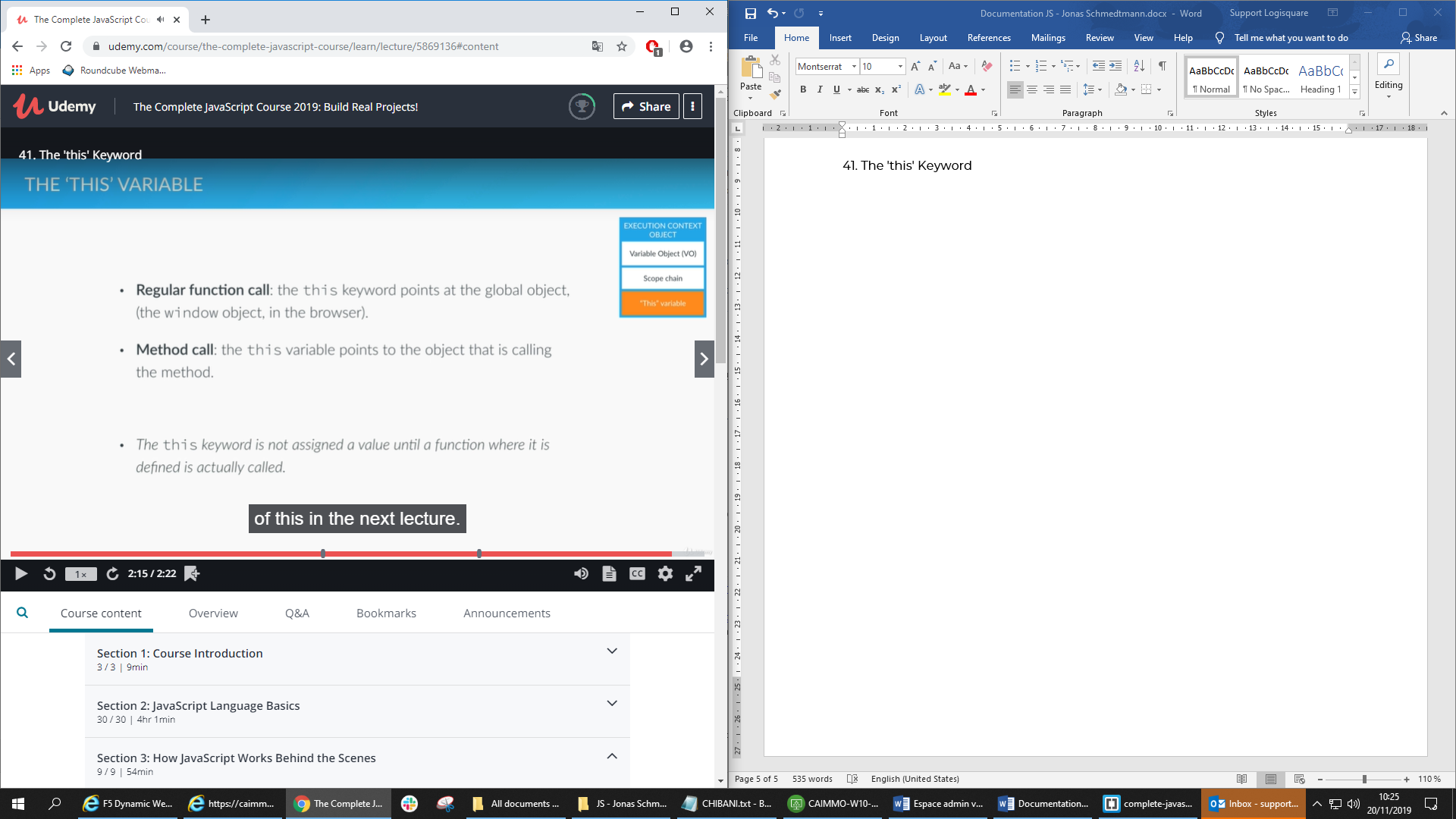
retirement(1996);

var retirement = function(year){

console.log(65-(2019-year));

}

41. The 'this' Keyword



Boths points on the global object window

1) points on the global object window

console.log(this);

2) points on the global object window because a regular function is attached to the window object :

calculateAge(1996);

function calculateAge(year){

console.log(2016-year);

console.log(this);

}

3) this points on the current object john

var john = {

name : 'John',

yearOfBirth : 1995,

calcAge: function(){

console.log(this);

console.log(2019 – this.yearOfBirth);

}

}

john.calcAge();

4) this points on the window object and not the john object, because innerFunction stills a regular function

var john = {

name : 'John',

yearOfBirth : 1995,

calcAge: function(){

console.log(this);

console.log(2019 - this.yearOfBirth);

function innerFunction(){

console.log(this);

}

innerFunction();

}

}

john.calcAge();

5) even if we create a new object, and use the method in other object it , the this keyword still point on the mike object and not the window object

var john = {

name : 'John',

yearOfBirth : 1995,

calcAge: function(){

console.log(this);

console.log(2019 - this.yearOfBirth);

function innerFunction(){

console.log(this);

}

innerFunction();

}

}

john.calcAge();

var mike = {

name : 'John',

yearOfBirth : 1984

}

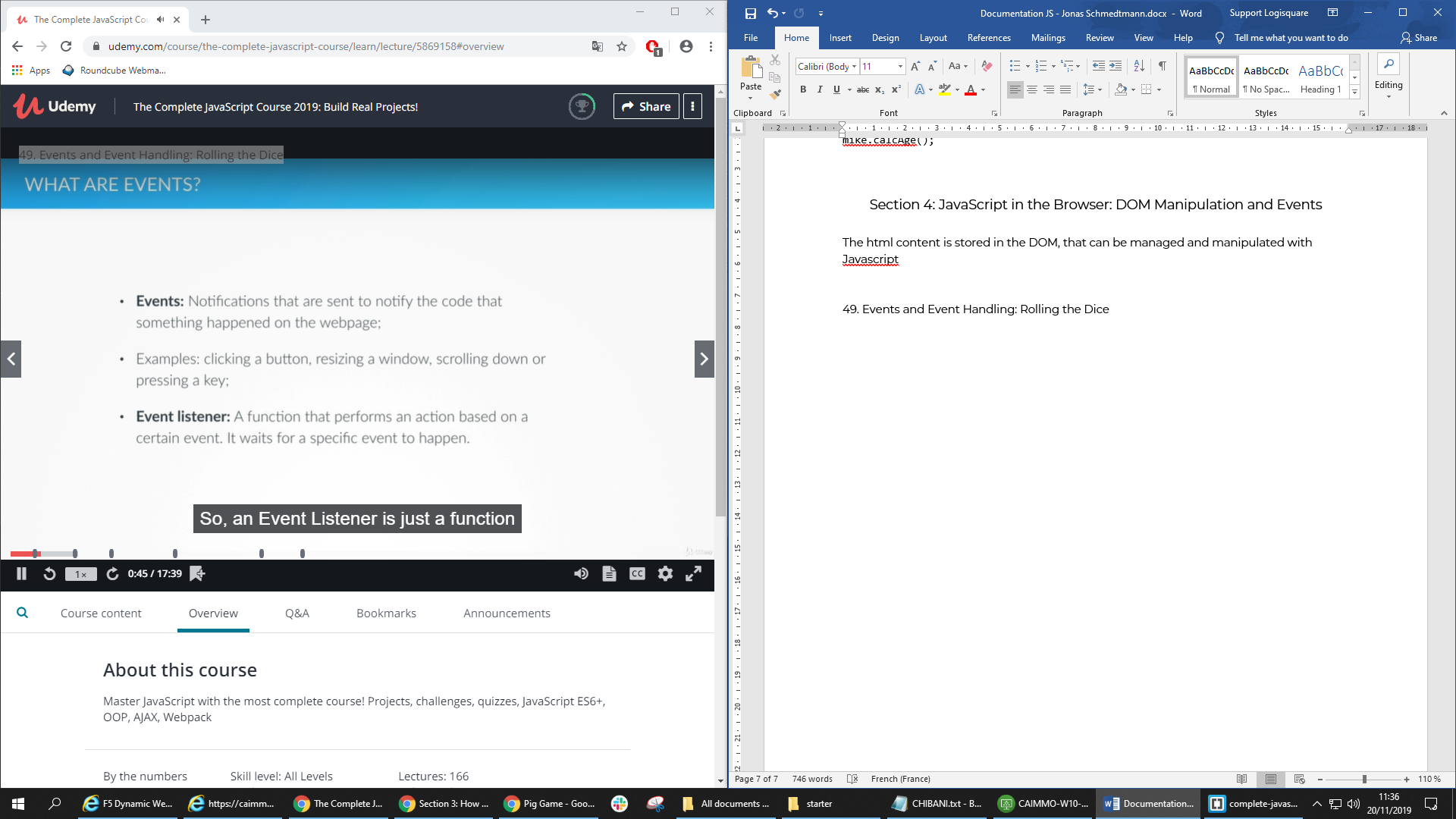
**mike.calcAge = john.calcAge;**

mike.calcAge();

Section 4: JavaScript in the Browser: DOM Manipulation and Events

The html content is stored in the DOM, that can be managed and manipulated with Javascript

49. Events and Event Handling: Rolling the Dice



Anonymous function: is a function that does not have a name, and can’t be reused

53. Finishing Touches: State Variables

State Variable: store the state of an variable on our system

22/11/2019