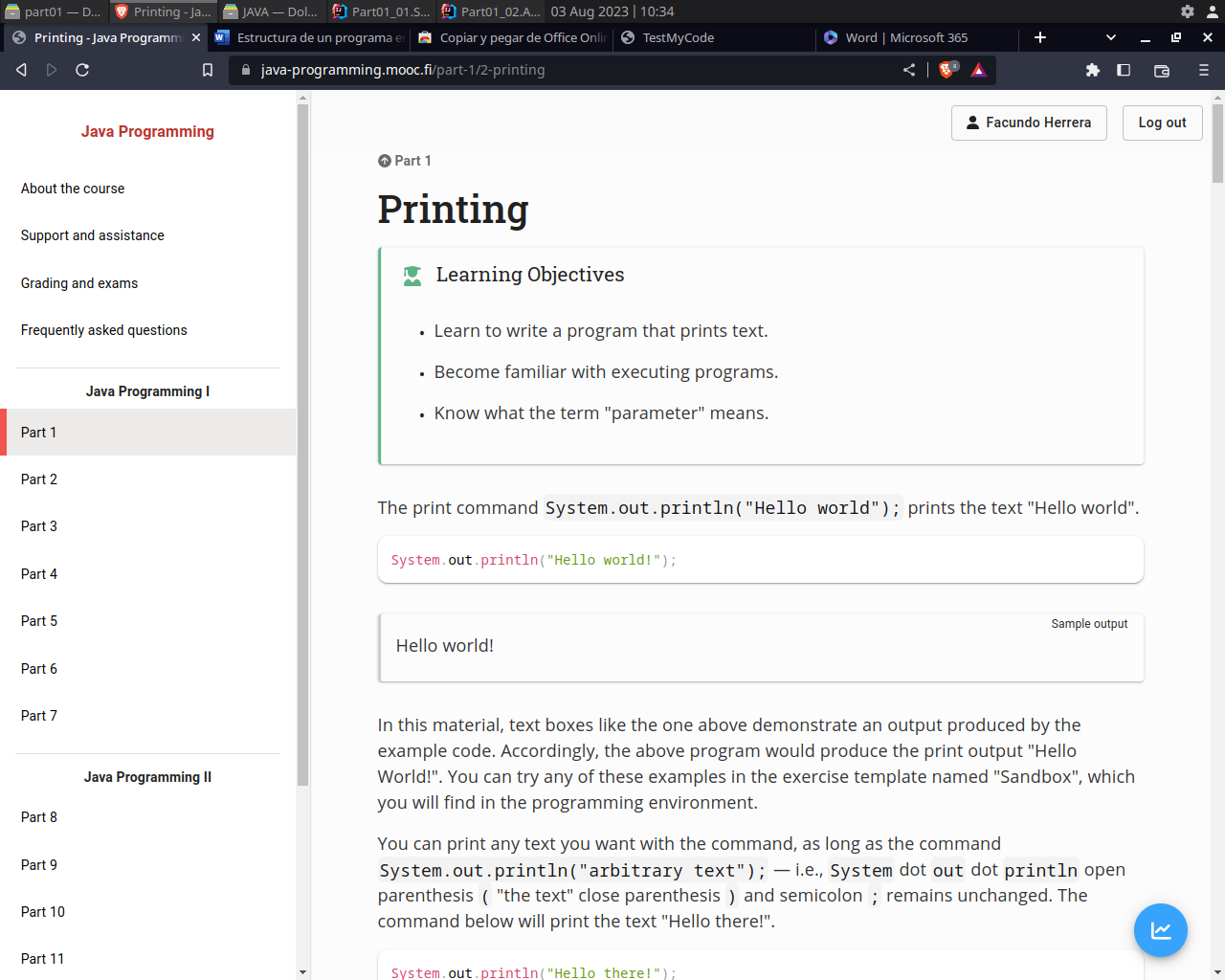
Estructura de un programa en JAVA:

public class Sandbox {  
 public static void main(String[] args) {  
 // Write your program here  
  
 }  
}

El comando System.out.println("Hello world"); printea el texto "Hello world"



La ejecución de un programa arranca desde public static void main(string[] args) { hasta el último }

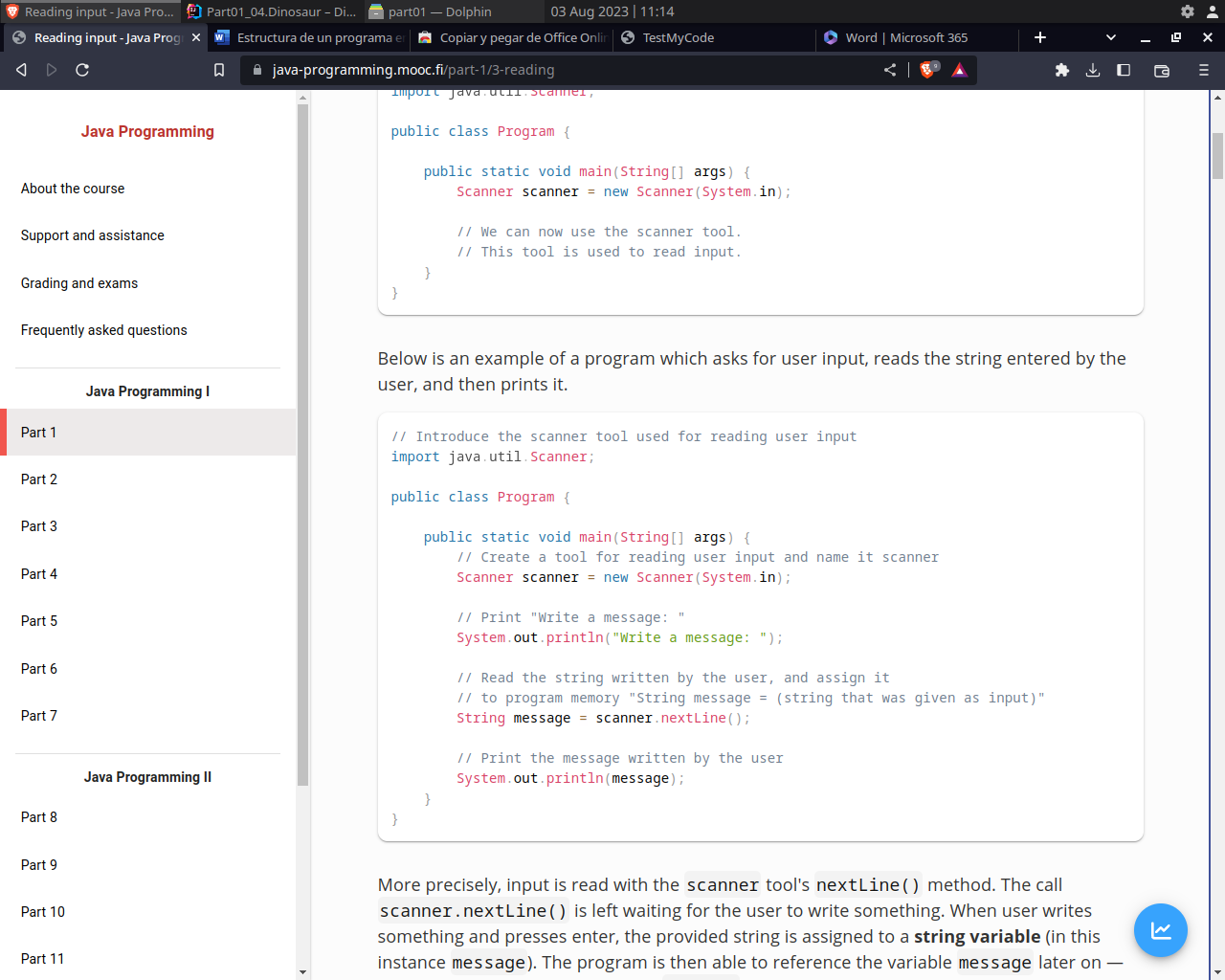
Para leer usamos Scanner

Esto lo hacemos primero importando desde import java.util.Scanner;

Y ya una vez en el main del programa “creamos el objeto” para usar el Scanner con Scanner scanner = new Scanner(System.in);

(donde dice scanner en minuscula podemos poner cualquier nombre que deseemos)

Acá un ejemplo:



Parseo para lectura:

String valueAsString = “42”;

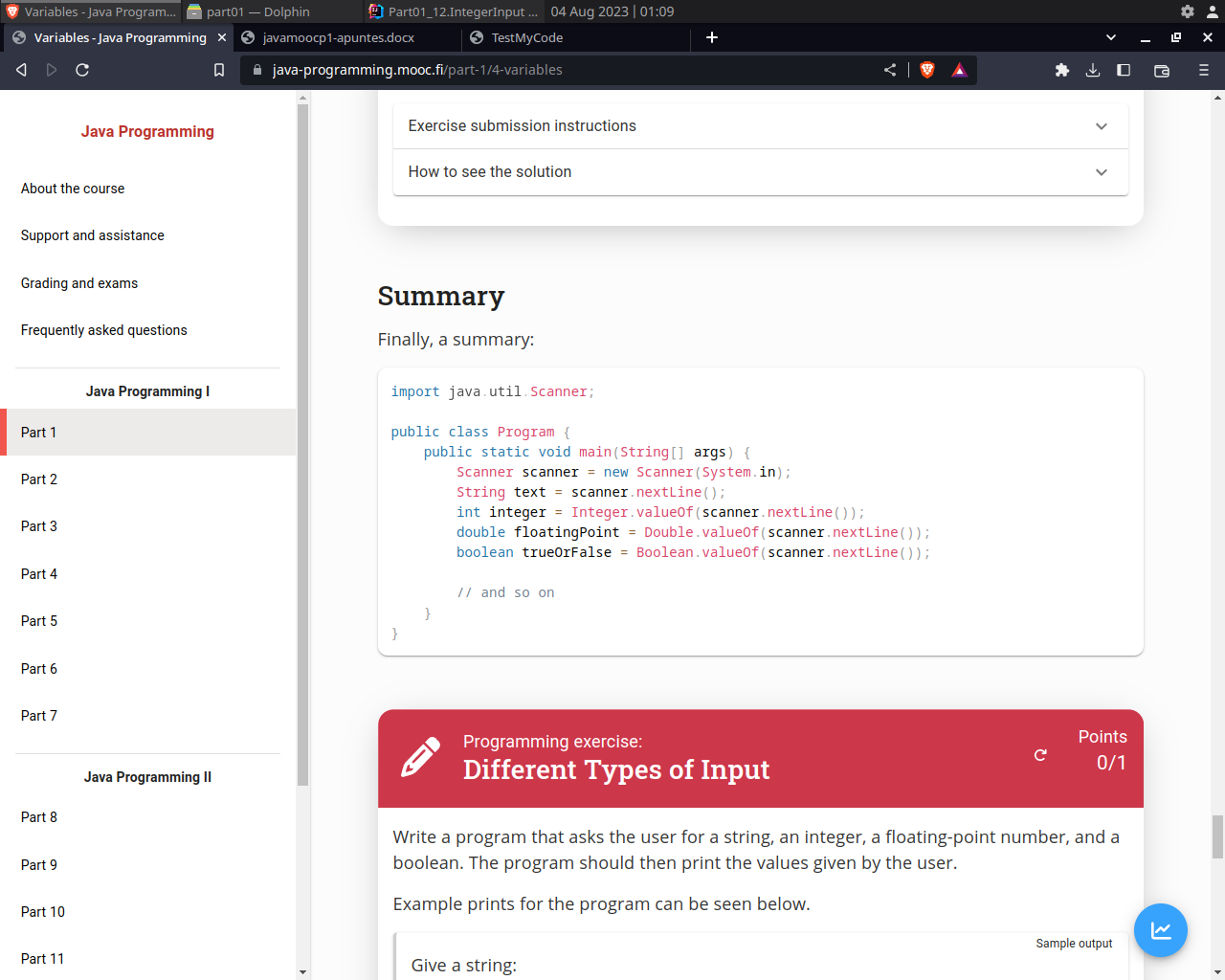
Es un string que retornaría 42

Ahora para convertirlo en int usamos Integer.valueOf(nombrevaliable)

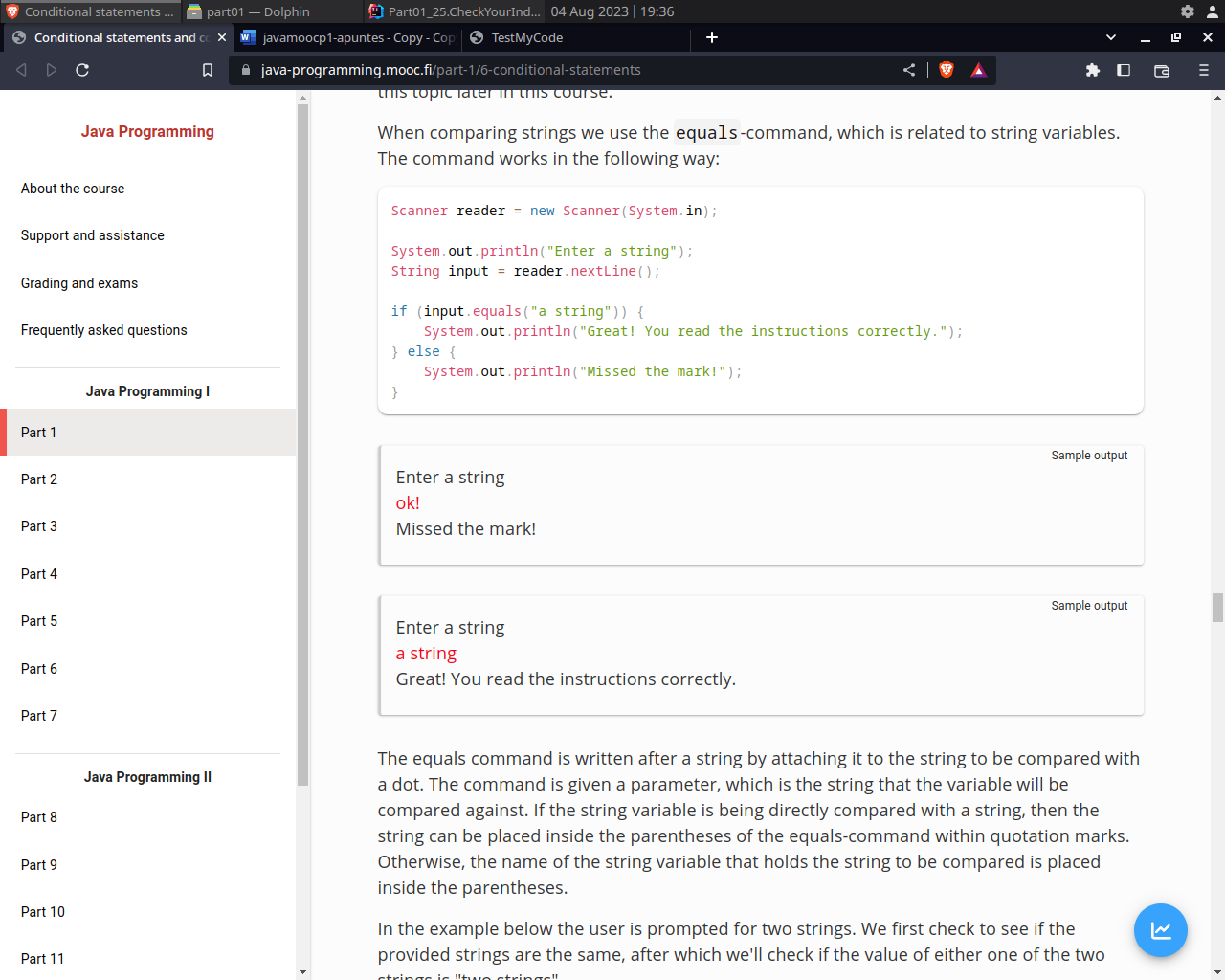
Ej: int value = Integer.valueOf(valueAsString) que ahora sería una variable de nombre value con tipo int

Y así con el tipo de dato al que queramos convertir TipoDato.valueOf(variable);

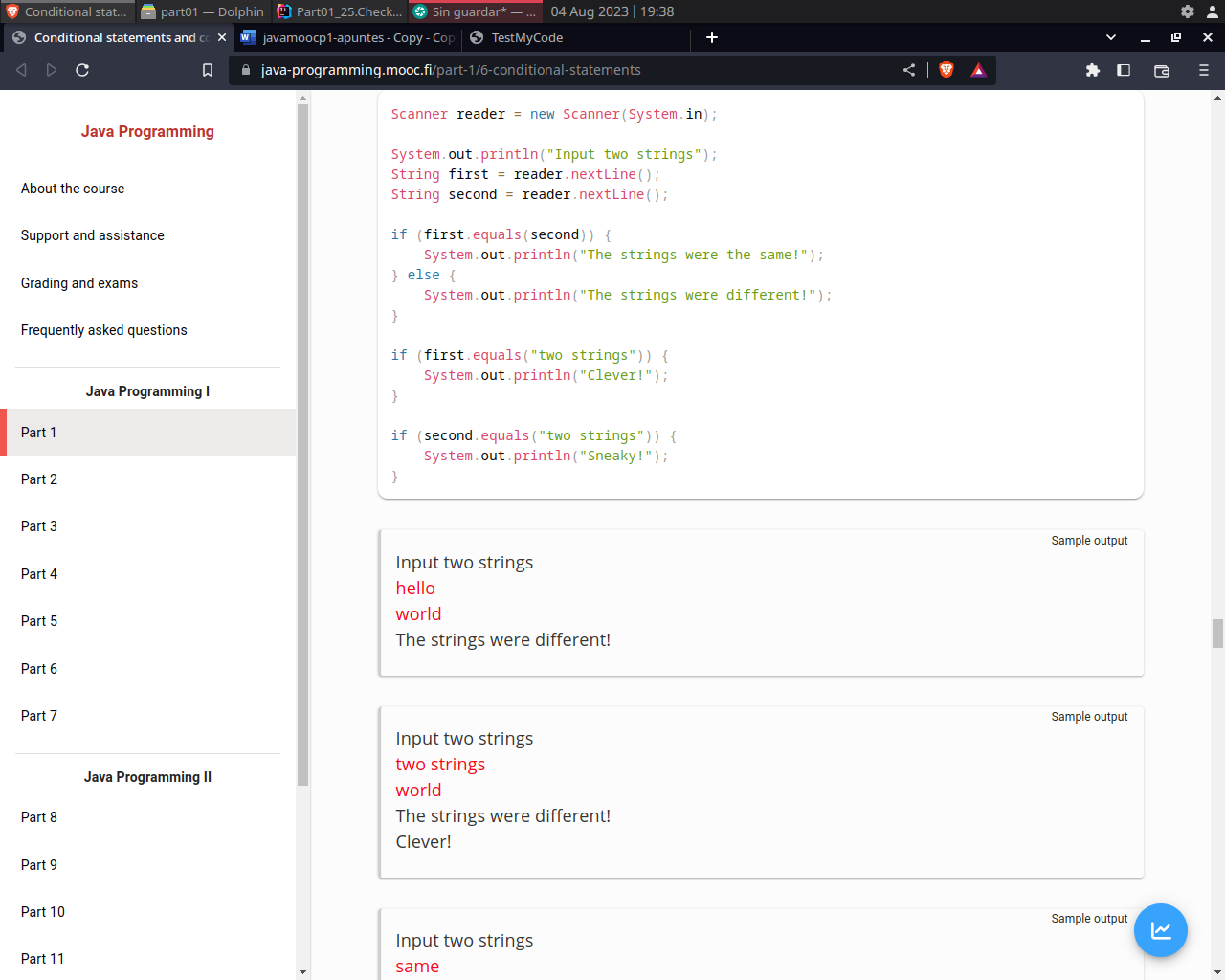
También se puede leer inputs con el tipo de dato que queramos de esta forma:



Para comparar Strings a diferencia de los int, bool, etc no usamos == si no que equals:



Y así para dos variables distintas:



**METHODS:**

Un método lo creamos fuera del main con la instrucción

Public static void methodName(){

// metodo

}

Así por ejemplo creamos qsy una función que calcule si un número es primo. Entonces cada vez que querramos saber si un número es primo simplemente llamamos al método esPrimo(int); y ya está

Para definir metodos con inputs lo creamos igual que el anterior pero así Public static void methodName(int variableName){ (por ejemplo int, podría ser otra cosa)

Para más de un parámetro es lo mismo, solo agregamos más opciones un una coma , -> methodName(int num1, int num2)

En estos ejemplos no especificamos el retorno:

Si no necesitamos que RETORNE un valor, por ejemplo puede imprimir en consola o simplemente cambiar el valor de la variable, ponemos void

Si lo que queremos es que retorne un valor, en lugar de usar void usamos directamente el tipo del dato y al final del método el ponemos return:

Ej:

public static int alwaysReturnsTen() {  
 return 10;  
}

Las variables definidas dentro de los métodos solo tienen validez dentro del método, no las podemos usar fuera.