Programming languages		Python	Notes(Python)
Operands	"+"	int + int = int, if float, it turns into float	**Addition** if there is a float in there, float will always trump, meaning 2 + 2.0 = 4.2
	"-"	int - int = int, if float, it turns into float	**Subtraction** if there is a float in there, float will always trump, meaning 2 + 2.0 = 4.1
	X	int * int = int, if float, it turns into float	**Multiplication** If there is a float in there, float will always trump, meaning 2 + 2.0 = 4.0
	/	ALWAYS FLOAT	**Division** it will automatically display the decimal Do not divide by zero
	Floor Division	Removes decimal (//)	**Floor division** 2.0//2.0 != 2.0 but it equals 2. Floor division does not round, it cuts off decimal
	Exponents	**	**Exponents** Use exponents **
	Modulus	% (7%2) = 1	**Modulus** Its return the remainder of a division. Use it for comvertions, divide by 16, remainder etc etc
	Square Root		
Basic Anotations	Ending a line	N/A (no ending annotation needed	**Annotations (end of line)** No annotation needed
	Spaces	tabs and spaces are important	**While spaces** play a very important role. They will change and work with Python

	Beginning a file	Nothing, start typing	**Starting a file** Annotations (file beginning)
Importing	basic scheme		**Imports**
		import *name*	Basic import of a whole program
	specific scheme	from *name* import *class*	Import specific classes.
	nickname scheme	import *name* as *nickname*	Give is a shorter nickname for ease of use.
Importing Specific Classes	Random Class		
		import random	not built in, random must be imported
		name = random.ranint(st,end)	IN BOTH PYTHON AND JAVA THE END IN NONINCLUSIVE
	Input		
		BUILTIN	BUILTIN use input as such. X = input("questions'") - always yields a string
	formatting ways		

	File Import		
		import os	
		os.path.abspath(path) os.relpath(path)	create a relative or absolute path from to where to import the file
Primitive Types	int	*name* = 2 (auto selects type	
	float	*name* = 2.0 (auto selects type	
	string	*name* = "string" (auto selects type	Python will automatically select the type of variable that the variable is whether it is float, int, string or bool
	char		
		name = idk (auto selects type	
	char/ASCII Number		

Wrapper Classes	Boolean	*name_snake_case* = True (auto selects type	Use keyword True capitalized to show the inequality or False
	CONSTANTS	N/A	constants are only indicated in python, such as NAME = "bob". It can still be changed but it should no be
	Integer / Double		
	comparison of wrapper class		
Input	How the user inputs things to the program	*name* = input("input plz")	In python, every single time there is input, it is changed to a **string.** You can change it later on using **type**(*name)*
	Strings as inputs		
Turning Variables to different Types	Change to string to integer	mi = "2" mi = int(mi)	you can use type(*name)* to turn into float, int, string, etc
	Parse integers		

See type variable is	Type that each variable is	print(type(*name*))	use it to determine what type of varible you're working with
Print Options (Strings)	Print Single Line	print(*name*)	Prints a single variable or string use either ' or "" to surround a literal string
	Print Single Line with Break at End	print(name + "\n")	Prints a single variable or string with modifiers
	Print Line with "f" string	print(f"{name} string with variable")	f string used to combine literal strings and variables
	Print Line using .format (outdated)		

Print Options (floats and ints)	Format Floats: f strings	print(f"{*name* **:.2f**}")	format to two decimal places
	Format Floats: (alternative outdated)	print("text {a:.2f} textext" .format(name))	outdated method used for formating a string
	Special Formating (.format)		
Splitiing and Selecting	Variables		
	Strings		
	Char (Characters)		

Conditionals/ Comparisons (Integers)	Operators	"==" equal "==" equal != not equal > greater, >= greater or equal <less than, <= less than or equal	operators when using conditionals in python
	if statements/ Shorthand if	if **condition:** **->tab** statement	if statements in python are run by using the keyword if with a condition, followed by the : when writing within the loop, must be tabbed in. **tabs are important in python**
	else statements	else:	it does not take any statements because it is just a cover for everything is not covered by the if statement
	else if statements	elif **condition:** **-> tab** statement	elif needs a condition
	Bool Conditionals		

	Switch	will expect a statement to guide it between a choice of paths otherwise go to default	
	Comparing doubles and floats		
Logic gates		and, or, not	and or and not are used literal, no symbol
String Operations	Change case (upper or lower) or get a true if comparing using is and to is to transform	x = tOm x = x.lower() .upper(), .lower	Used to change the case of letters
	Joining Strings		
	Replacing Strings	ind	

String Comparisons (with if)

Cutting Strings

Split method

Getting index of String

String Builder

While/for	Will repeat what is in the loop until the statement is true	While *statement:* *tab->do something*	While takes a statement and puts it on loop(uppercase indented
	For loop	for items in list:	for *items* is the variable used for indiviuals, it will cycle through the list once and then stop
Array	A special value having one name but storing lots **index = position in a location of an array** **Vector/array element = individual parts of an array.**	x = ["a","b",]	
Array Properties	Updating lists adding to the end of a list	x[index] = "c"	will add or replace
		fruits.append("d")	will add to the end of the list
	Length	fruits.length()	get the length of a string as an int
	Copy Array		

	Fill Array
	Add Element to array
	Remov e Element form arrray
External Document	Reading from an external document
	Create a buffer for reading
	Writing to an external document
	Wrtiting to an external document

Close File

Exists/hasNextInt();

Reading and making a document

StringWriter
PrintSWriter

StreamReader
StreamWriter

Exceptions (on main)

Random (sort later)

Exceptions

Exceptions

ArrayLists	Arraylists (functions)
	Iterator<strings> it = nameList.iterator();
	enhansed for loops
	putting primitive types in arrayStrings
	get keys and values fom an arraylist

Hashmaps

HashMaps

For:each (Enhansed Loop)

For:each (Enhansed Loop) (simple)

hashmaps
commands to acesss

Differences Between Different Programing Languages

Java	Notes (Java)	JavaScript
	Addition	int + int = int
int + int = int if float it turns into a float	if float present it turns into float	int + float = int
	Subtraction	int - int = int
int - int = int if float it turn into a float	if float present it turns into float	int - float = int
		int * int = int
	if there is a float in there, float will always trump,	int * float = int (internally
int*int = int. If float, it turns into float	meaning 2 + 2.0 = 4.0	a float)
	If you want it to correctly display a decimal, use at	
returns integer. IF SMALLER THAN 0 IT WILL	least one float	
RETURN 0. 2/3 = 0	2/3.0 = .666	int / int = float
	Floor division (integer division)	
	regular division of two integers will return a floor	
N/A	division, 5/2 = 4 not 4.5	
	Exponents	
Math.pow (base, exponent)		
	Modulus	
	12345 find the 4	
System.out.print(10%3) = 1	would be (12345 / 10)%10	
	Use it for convertions	
	Square Root	
Math.sqrt(number) = double	**~~~~~~Needs import**	
	Annotations (end of line)	
System.out.println("hi");	Lines end with ";"	
	White Space	
	White space does not play an important role. It will	
Tabs and spaces are NOT important	not change the code	

```
public class Prog1 {
  public static void main(String[] args){
```

Starting a file
Type everything within those {} after args

```
import java.util.*
```

Imports
Imports java utilities

```
import java.util.* (import everything) or
import java.util.Random;

Random name =  new Random();
int name =  namerandom. nextInt(qttofvalues);
namerandom.nextInt(qttofvalues) + 10;
namerandom.nextint(max-min + 1) + min
namerandom. setSeed(specificSeed);
```

must be assigned first.
Scanner *name* = new Scanner(System.in)
not built in must be imported
(randNum % 9) possible numbers from 0 - 9
(randNum % 9) + variable will make it to a range so between 10-20 put 11
4 billion numbers made negative to positive (2E31)
Set low limit using the plus or minus
Think of the seed like minecraft it dictates behaviour

```
import java.util.* (import everything) or
import java.util.Scanner

Scanner name =  new Scanner(System.in);
int name  = scannername. next/Line()
java.text.*;
decimalFormat name  = new
decimalFormat(Pattern); (###,###.00)

name. format(integer)
```

Allows user to input using the keyboard. First import it , then name it.
decimalFormat
This is a special import way, it uses import to put a pattern.
Add nextline after nextint to avoid the error in which it reads \n

import java.io.*;	**Import io/ External Documents**
import java.io.FileInputStream;	The first way is by the book and the other one is by dr lang. Use the second one its easier
import java.io.IOException;	In the first way we import file Input Stream, add it to a new object and then put it into a scanner as input instead of keyboard
java.io.IOException;	
\\\\\\\alternative way\\\\\\\\\\\\\\\	
Scanner inFile = new Scanner(new File("yes.txt" or could be file name)	For doing absoluta path we must do the whole file name with path folder \folder\text.txt
\\\\\Finding a document outside main folder\\\	
import java.io.*	
File filetoUse = new File("path");	for going one folder deep at least we must create a file object
	Someitimes you might need to start a value to setit to equals zero to initialize
int *name* = 2;	
float *name* = 1.2;	
double *name* = 1.2;	
String *name* = "this"	String is **always capitaized** and the difference is string literal vs string
String *name* = boy	
char *name* = +	A single letter or character that can be used
char *name* = '+'	Character literal is surrounded by **single quotes**
	Char are stored as a number where 'a' might be = 97 in memory
int *name* = "yes"	
char *name* = *name*. charAt(0)	To get one character you would use .charAt(0);
	Character are stores as ascii number, each character has its own
	CHARACTER LITERALLS ONLY (WITH '') ARE STORES
	"" + will always yield a string and not the storage number of char
a = asciii number 65	

boolean isLargeParty = true
boolean isLargeParty = false

Do not capitalize it just leave it without quotes or any markup

is Teenager = (kidAge > 2 && kidAge <10)
if (Teenager){}

Use this as conditional extra way to do something and declare it as true or not

final NAME = 2;

This will prevent anyone and anything from changing this value. Use it only if sure object will not change

Integer integer.parseInt(String);
Double Double.parseDouble(String);

Wrapper classes

used for jumping between GUI and internal

All start with Character.iswhatever

Character.iswhatever

isLetter("C") - true if c or C
isDigit (c) - true if digit
isWhitespace© - true if ""\n
isUpperCase / isLowerCase
Character.toUpperCase/ toLowerCase©

Several Methods of working with characters
something.isLetter(3) - the object finds whatever is in parenthesis, at a certain spot, it returns True, else False
Reasign if changing case. Self assign

~~~~~~~~~~~~~~

Scanner *name* = new.Scanner(System.in);
string *name* = *scannername* .nextString()

import the scanner first, declare it and use it in next object notation as shown
next() - only the first word,skips rest,SKIPS WHITE SPACE
nextLine() - whole line as string even white space
nextInt() - turns input into int
nextDouble() - turn input into a double

priming an input. Make it equals zero to start a value

stringName = "bobby mike"
Scanner inSS = new Scanner(*stringName*);
firstName = inSS.next();
lastName = inss.nextLine();

Input from Strings

This gets inputs from strings and uses them to run a program using nextInt next etc

p = 2
m = (double)(p) **= 2.0**

Type casting

specify the type of integer the variable is
Turn both into double before dividing

String bob = "11";
int number = Integer.parseInt(*bob*);

Turns into a string an integer .INTEGER CAPITALIZED!!!!

	Prints a single variable or string use only "" to surround a literal string
System.out.println(*name*) or System.out.print(*name* + ″\n")	Prints a single variable or string with modifiers
System.out.printf("%s thank you", variable name) %c character %d decimal %e exponential %f floating-point number %d integer %o octal number (base 8) %s String %u unsigned decimal (integer) number %x number in hexadecimal (base 16) %t formats date/time %% print a percent sign \% print a percent sign	printf is like an f string, must specify what type it is: %s = string, %f = float or double. %d = int %e = scientifiic float = 6 decimal points format by saying %.3f use %1.0f to specify width "-" this left alligns the output given a width pads with spaces (string ___)
printf("%-4")	"+" prints plus for postivie, NEGATIVE ARE ALWAYS NEGATIVE
printf("%+3s")	08s - **prints zeres if not enough characters(including symbols) to the left**
printf("%08s")	**Whole numbers count symbols etc decimals only decimals** .2 - Cuts string at 2 decimal places. if too short, it will fill with zeroes
printf("%.2s")	"e" indicated the scientific form print
printf(10e2)	IF IT IS LONGER IT IS NOT TRUCADED
frintf("%-8s123") = string 123	- if not long enough it will add spaces to the right
print("text {a} textext" .format(name)	outdated method used for formating a string

printf("%.2f")	See more in f strings.
	Import java.text.* ~~~~~~~~~~~~`
	To use this, you must declare it, create a pattern and import it.
import java.text.*;	DecimalFormat *name* = new
DecimalFormat name = new	DecimalFormat("#,###,###.00");
DecimalFormat("#,###,###.00");	using the *name* we create a pattern where it will fill
name. format(***integer***);	up with the pattern and the zeroes if none avaliable
	Character Selection
string tom = "hello"	the **zero** specifies which character position is
string initialName = tom.charAt(0)	shown, giving "h"
	This specifies that if it is not longer than 8
printf("%08s")	characters, it will fill with zeroes. IF left blank it will fill with space from **the left.**
	Decimal cuts, it will only do the first two characters.
	Will not add!!! it will add zeroes if too short
printf("%.2s")	**Whole number adds to the left. Decimal adds if not enough**
frintf("%-8s123") = string 123	"-" will add blank space to the right to fill a size
int name = "yes"	
char name = name.charAt(0)	get a single character and turn in into a char

"==" equal
!= not equal
> greater, >= greater or equal
<less than, <= less than or equal

operators when using conditionals in JavaScript
DO NOT COMPARE FLOATS OR STRINGS USING OPERATORS IN JAVA IT YIELDS UNEXPECTED RESULTS. (X-Y) < 0.001 IS CLOSE ENOUGH

if (condition){
statement}

if statements in java are made by using () and putting the condition within, followed by {} wherein the statement must be made

y = (condition)?true:false;

shorhand if just declare within the variable that you need the condition in parenthesis followed by ? True and then : false stmt, must both have the same variable assign (if and else same = some)

else{
}

it does not take any statements because it is just a cover for everything is not covered by the if statement

else if(**condition){**
statement

two words followed by a parentheis statement and backets

is Teenager = (kidAge > 2 && kidAge <10)
if (Teenager){}

Use this as conditional extra way to do something and declare it as true or not

switch(expression){
case 0: // case can be a string or char 'G' '
do something
break
case 1
default:

A Switch is just an if statement, limited to string char or int. it can be only check for ==. Case looks for the char or int to == it and then excecuses
Within we use : to express something

Use break otherwise it will continue checkin throught all of them

Math.abs(x-y) == 0.001

Do not compare double or strings using == but use math abs, we want close enough that's all

&&, | |, !

equivalents to and or and not

All start with Character.iswhatever

isLetter("C") - true if c or C
isDigit (c) - true if digit
isWhitespace© - true if ""\n
isUpperCase / isLowerCase
Character.toUpperCase/ toLowerCase©

Several Methods of working with characters
something.isLetter(3) - the object finds whatever is in parenthesis, at a certain spot, it returns True, else False
Reasign if changing case. Self assign

string.concat("!")

Concat will add whatever you pun in parantheis to the indicated area! USE DOUBLE QUOTES

String = hi string = string + "n"

Will add indicated value to the lines literay adding it

Replace Strings

userText = Hello
userText.replace(h,j);

You can edit by using char or literal strings to edit the strings

This will return a boolean expresson either true or false when set to a variable
do not use == to compare strings, it returns addreesses

String1. equals("apples")
compareTo compares string lexicographically using the unicode value of each string.

string1.compareTo("thing")
string1.compareToIgnoreCase("string")
ignore case, also works with equals. **compare to returns a postivie number** if true or negative if false

String Tokenizer

```
StringTokenizer str  = new StringToKenizer (names,",");
for(int I  = 0; i<count; i++)
{String name = str.nextToken();
System.out.println(name);
```

Using this to user the " " to undemine and cut each then using a loop you cycle through it and cut displaying or assigning each
user (true) to include demimeter

SubStrings

```
string = google.com
string.substring(7,10)
```

It will start from a certain string and go to the end of the string or you can give ending index
(noninclusive)

Split ()

```
string names = "bob,joe,jane"
string [] pieces = names.split(","); <- store names split by comas in a new array
for (int I = 0; i< pieces.length; i++)
```

Use it to separate list using (, | \\.) (coma or period)
You create a list and store it
make a new list to store the parts and access them later.

Get index of an object

```
indexOf(thing(space etc );
indexOf(object, starting index );
lastIndexOf(object );
```

returns index on a certain string , first occurance or last occurance if asked. Will return -1 to indicated it failed

NO IMPORT NEEDED BUT MUST CREATE A NEW OBJECT

String Buildier (threadsafe)

```
StringBuilder sb = new StringBuilder();
```

An object that will build the object more efficiently using the memory.

while (userNum>0){}	While takes a statement and put it on loop, lowercase in brackets (**indefinite loop)**
do{} while()	exit conditional loop. Does something at least once and the while after determinedthe tmies it will do it **(indefinite loop)**
for(count = 1; count <= 5;count ++){ do thing	**For loop (counter controlled Loop)** 3 parts initializing action, test condition, update action. Has to be controlled will not automatically stop
for (String name:playerlist) -- enhansed list	**Enhanced Loops** very robust and inflexible used for displaying values most of the time
int [ ] itemcounts = new int[3]//must provide length of array int[] itemcounts = {10,10,10] int [] array = {1,3,4,5}; -- only used in declaratoin x[]intex] = "c"	an array names itemCounts is declared 3 ints with each the value of 0 it can be declares separeately too if needed **Int[] array holding ints, integer[] array holding references to integer objects** you can give default values using the {}
fruits.length() -- for strings fruits.length - for characters or ints	gets the length of a list, no parenthesis
System.arraycopy (array, 0, array2, 0 , array.length)	**Array Copy** old method of copying a list

Array Fill

int[] cookies = new int[3];
fill(cookies, 5);

fills the array. Takes in the array and the object to put in it and then fills it all the way

Add Element

addElement(*listName, to add);*
fruits.add();

Add element to array

Remove Element

fruits.remove();

Remove an element from an array.

FileInputStream//Java.io

Allows program to read the contents of an external file

FileInputStream *name* = new FileInputStream(document.txt);

Use a scanner to set up a input stream and set equal to a variable using nextlint or line

FILEINPUT + scanner

Scanner inTxt = new Scanner(nameofinput);
new.File(filename)

new File(filename)

Creates a buffer that keeps the file inside and allows it to be read

.isFile()

isfile checks if the object is a valid file

fileOutputStream outFile = new FileOutputStream(document.txt);

\\
\\

printWrite outST = new PrintWriter (*name of fileoutputstream);*
outST.println("hello world")
outST.close();

	fileInput / output close
	check to close borth file input and file output streams. if it does not get closed, we do not get
outSt.close() (name of the printwriter)	anything in the file because it is still in the buffer
	Exists
	check if the variable exists usually done with the inported file.
if(file.exists)	**Has**
hasNextInt();	checks if file has next int continues on
StringWriter fullName = new StringWriter();	**StringWriter and PrintWriter//Java.io**
printWriter fullNameOSS = new	String Writer creates a string within the proram and
PrintWriter(fullname);	adds to it using printWriter while the other example
fullName = fullName.toString();	shows printWriter using it with a text file and editing it by itself using a buffer
printWriter outfile = new PrinterWriter("file.txt");	
String line	can be used for writing to a new document. As
line = s.nextline();	shown on the example below. The stringWriter must
outfile.close()	be followed by the printWriter and
public class *programName* {	
public static void main(String[] args) **throws**	
FileNotFound (error to ignore)	
{	handles certain known errors and skips it in main

file.close(); (you close the filewriter or printWriter)

check if file has nextint and closes file

Classes

constructors and functions

im tired boss and a bit depressed you know?

Wrapper classes - Integer, Character, Boolean,

integer.toString();

toString();

try catch

if(statements) throw new exeption("msg");

catch(Exception excpt) {excpt.getMessage());
excpt.printStackTrace();

catch Exception excpt{excpt.getMessage();

.add . size. Get , .set(index, replacement)

.add(to the end) , add(index, object);, delete(intex);

ArrayList namelist = new arrayList();

yo ucan indicate the type of data type it will hold by using <>

ArrayList<String> nameList = new ArrayList<or leave empty>();

(String) namelist.get()

Iterator<strings> it = nameList.iterator();

while(it.hasNext())

{String name = it.next();

System.out.println(name);

for(string name: nameList)

{system.out.println(name)

Autoboxing, automatic convertion othe java complier maekes tbetweek primitity types and their corresponing wrapper

ArrayList<Integer> Scores = new ArrayList<Interger>();

get: Uses a key to get the value

put: (thing,thing) which are linked

values: gets the values of a hashmap

.get() .put() .values()

Hashmap<String> name = new etcetc

we use the for loop on the name of the Hashmap, (hm) in this case, specifying for each entry and saying the type of things inside, then we use a variable name, var, in this case. Then we do it to map.entrySet() which gives the length
Then we use var.key or var.value depending on what we want to access

```
  for(hm.entry<st,st> var:hm.entrySet())
System.out.println(var.getentry(),var.getValue()
```

```
For(string Key: map.keySet()
{
String value = map.get(key);
```

keyset gives the whole list of key sets
get: Uses a key to get the value
put: (thing,thing) which are linked
values: gets the values of a hashmap
keyset: gives the whole keyset values
size: returns the size of the hashmap

.get() .put() .values() .keySet()
.size()

Notes(JavaScript)	C#	C# Notes

If there is a whole number it
will cut the decimals away

if there is a whole number it
will cut the decimals away

Multiplies etwo values using
an asterist
decimal internally
JAVASCRIPT DOES NOT
DIFFERNETIATE BETWEEN
FLOAT AND INTEGERS

Imports

```
using System;
```

Using the keywords using

BUILTIN
Console.WriteLine

bool flag = true;	Bool instead of boolean or just implying
int result = Int32.Parse(string);	Parses a string into n int

Character Selection

string name = "mike"
char = name[0];

The number will specify the position of the element

StreamReader/StreamWriter

Author: Eder Martinez

github:theEderMartinez