

WELCOME BACK TO WEEK 2
OF INTRO TO PYTHON

REVIEW OF WEEK 1

- INPUT VS OUTPUT
- PRINT STATEMENTS – BASIC STRUCTURE
 - CONCATENATION (+) & REPEATED CONCATENATION (*)
 - MULTIPLE ARGUMENTS
- VARIABLES – TYPES (INT, FLOAT, STR, CHAR)
 - NAMING CONVENTIONS (CAMELCASE VS SNAKE_CASE)
 - INITIALIZING VALUES (HARDCODED VS INPUT) & UPDATING THEM
- ARITHMETIC OPERATORS – TYPES: (), **, *, /, //, %, +, -

CODING & DEBUGGING

- SYNTAX VS LOGICAL ERROR
- INCREMENTAL DEVELOPMENT:

CODE SMALL PORTIONS OF A PROGRAM AT A TIME AND TEST THE CODE.

IF IT WORKS PROPERLY, MOVE ON TO THE NEXT PORTION. IF IT DOESN'T, IDENTIFY THE PROBLEM CODE & MAKE APPROPRIATE EDITS.

REPEAT UNTIL YOUR CODE IS COMPLETED.

TEST THE CODE IN ITS ENTIRETY TO ENSURE IT WORKS PROPERLY.

As a frequent world traveler, Arup Guha has to exchange money to get various different currencies. However, he needs some help to determine exactly how much money he will have left after one of his trips. You will write a program to help him with his estimate. In particular, you will prompt the user for the following information:

- 1) How much US currency he/she has.
- 2) The exchange rate from US currency to the foreign currency.
- 3) How much of the foreign currency he/she spent.

The user exchanges all of his/her money for their trip. Usually, however, the user will have some money leftover when his/her trip is done. At this point, the user exchanges the rest of this currency for US dollars. Your goal will be to determine how much money the user has left in US dollars after his/her trip.

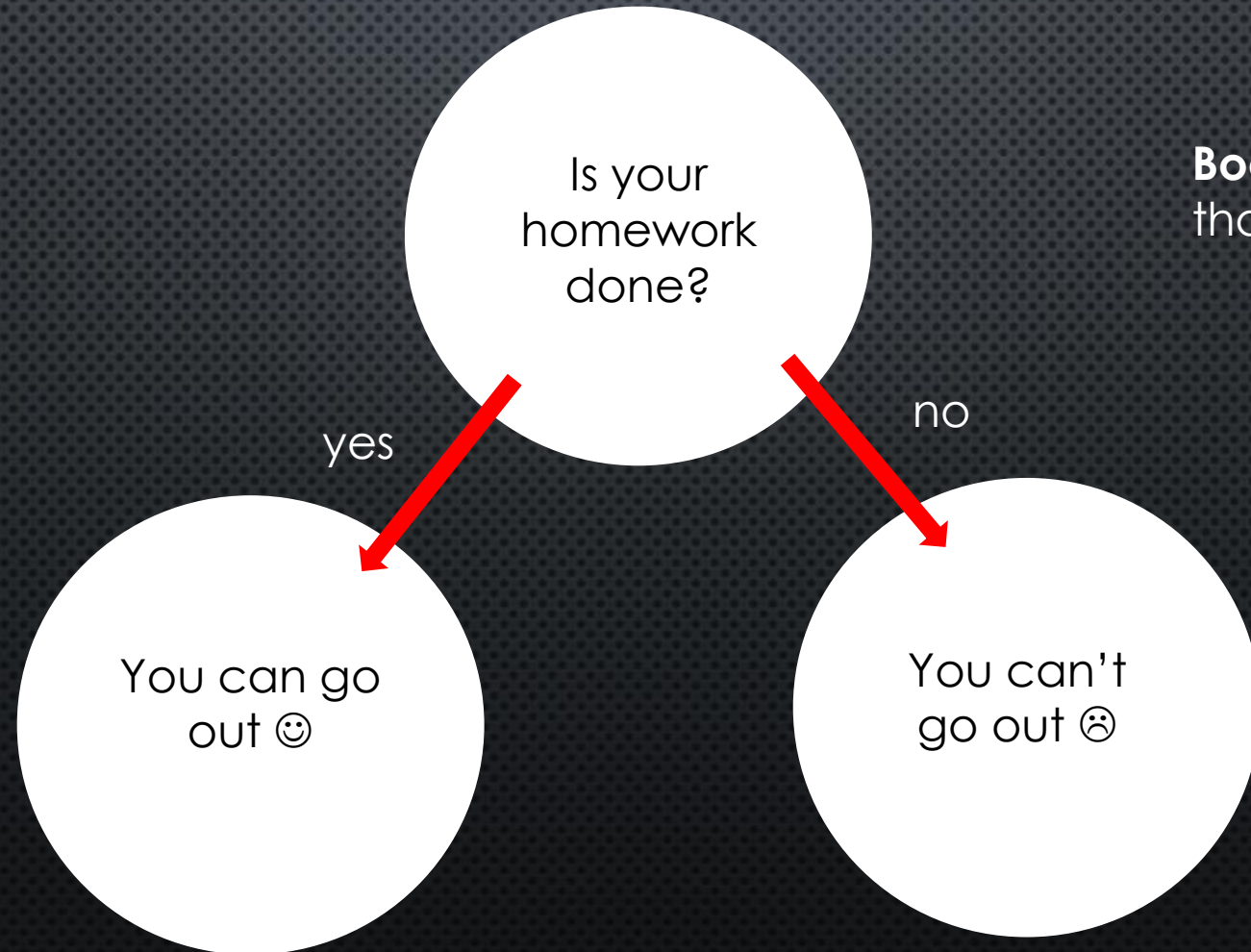
The charge for an exchange of currency is \$2 US. Thus, if you are exchanging 1002 US dollars for Canadian dollars and the Canadian exchange rate is 1.5, then you will receive exactly 1500 Canadian dollars. If you spend 597 Canadian dollars, then you have 903 Canadian dollars left to exchange back to US dollars. This evaluates to \$602 US. However, since the exchange fee is \$2 US, the user would actually have exactly \$600 US.

REFRESHER EXERCISE

CONDITIONAL STATEMENTS

- SEGMENTS OF CODE THAT RUN DEPENDING ON WHETHER OR NOT AN ARGUMENT IS TRUE.
- IN REAL LIFE, WE CARRY OUT DIRECTIONS, WE DON'T EXECUTE THE SAME STEPS EVERY TIME.

CAN YOU GO OUT TONIGHT ?



Boolean : a statement that is ONLY true OR false

IF STATEMENT

```
if <Boolean Expression>:
```

```
    stmt1
```

```
    stmt2
```

```
    ...
```

```
    stmtn
```

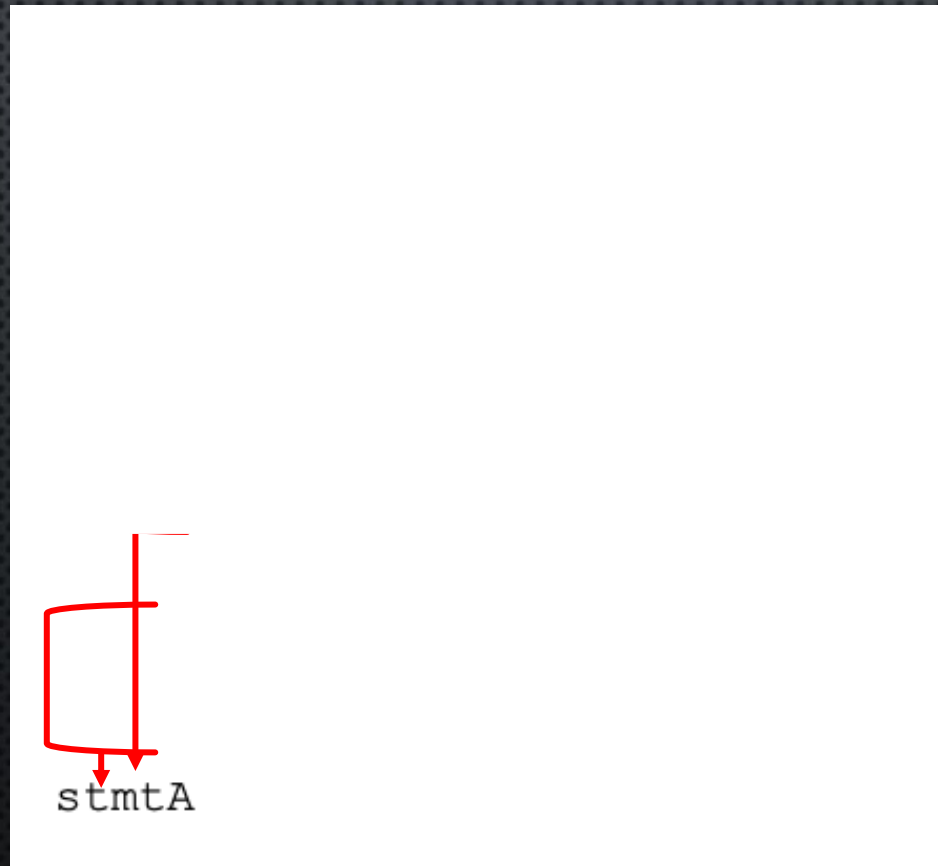
```
stmtA
```

- IN PYTHON, CODING BLOCKS ARE MARKED BY INDENTATION !!
- IF A BOOLEAN EXPRESSION IS TRUE THEN THE CODE WILL BE EXECUTED
- IF THE EXPRESSION ISN'T TRUE THEN THE CODE WILL BE SKIPPED OVER

MORE ON THE IF STATEMENT

- TO CHECK IF MORE THAN ONE OPTION IS TRUE...

1. Check the first Boolean expression
2. If not true, check the second Boolean expression
3. If not true run the indented block of code under else



- 2 options:
if<Boolean Expression #1>

else<Boolean Expression #2>
- 3 or more options:
if<Boolean Expression #1>

elif<Boolean expression #2>

elif<Boolean expression #3>

else<Boolean Expression #4>

NOTE: THE ELSE STATEMENT WILL
ALWAYS RUN IF IT EXISTS AND
ALL PRECEDING CONDITIONAL
STATEMENTS ARE FALSE

Relational Operator	Meaning
==	Equal to
!=	Not equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to

COMPLEX BOOLEAN EXPRESSIONS

Purpose: check multiple conditions within one expression - "and" / "or"

Operand #1	Operand #2	Result (and)
False	False	False
False	True	False
True	False	False
True	True	True

AND: BOTH CONDITIONS MUST BE TRUE

OR: ONLY ONE CONDITION MUST BE TRUE

Operand #1	Operand #2	Result (or)
False	False	False
False	True	True
True	False	True
True	True	True