Good morning (good evening)… ☺

A QUICK SUMMARY ON KEY CONCEPTS

What are the die-die must know things?

Boolean logic

Named after George Boole

Boolean logic (aka Boolean algebra) is the branch of algebra in which the values of the variables are the truth values **true** and **false**, usually denoted **1** and **0** respectively

When 1 and 0, we only need 1 bit to represent this!

Boolean logic

We can explore logical relationships via

Conjunction (AND)

Disjunction (OR)

Negation (NOT)

- Just as you can do things with numbers in numerical operations (plus, minus, divide).
- You can also do things with Boolean variables via the operators (AND, OR, NOT)

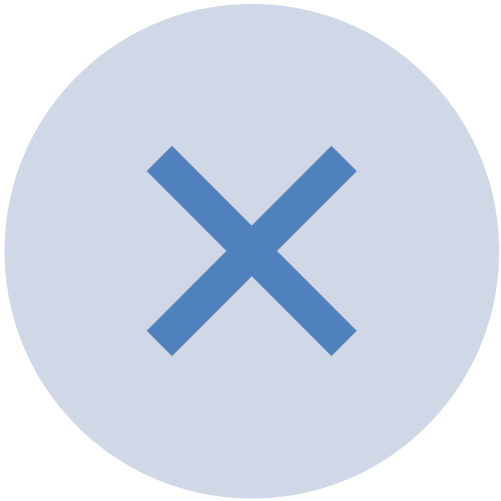
Boolean logic (evaluation)

1 **IS EQUALS** TO 1
EVALUATES TO **TRUE**

1 **IS EQUALS** TO 0
EVALUATES TO **FALSE**

Boolean logic (negation)

1 **IS NOT EQUALS** TO 0
EVALUATES TO **TRUE**



1 **IS NOT EQUALS** TO 1
EVALUATES TO **FALSE**

Boolean logic (conjunction)

1 **IS EQUALS** TO 1 **AND** 0 **IS EQUALS** TO 0 EVALUATES TO **TRUE**

TRUE AND TRUE is **TRUE**

0 **IS EQUALS** TO 0 **AND** 1 **IS EQUALS** TO 1 EVALUATES TO **TRUE**

TRUE AND TRUE is **TRUE**

Boolean logic (conjunction)

1 IS EQUALS TO 0 AND 0 IS EQUALS TO 0 EVALUATES TO FALSE

FALSE AND TRUE is FALSE

1 IS EQUALS TO 0 AND 0 IS EQUALS TO 1 EVALUATES TO FALSE

FALSE AND FALSE is FALSE

Boolean logic (disjunction)

1 IS EQUALS TO 1 OR 0 IS EQUALS TO 0 EVALUATES TO TRUE

TRUE OR TRUE is TRUE

1 IS EQUALS TO 0 OR 0 IS EQUALS TO 0 EVALUATES TO TRUE

FALSE OR TRUE is TRUE

Boolean logic (disjunction)

1 IS EQUALS TO 1 OR 1 IS EQUALS TO 0 EVALUATES TO TRUE

TRUE OR FALSE is **TRUE**

1 IS EQUALS TO 0 OR 0 IS EQUALS TO 1 EVALUATES TO FALSE

FALSE OR FALSE is **FALSE**

Boolean logic (condensation and precedence rules)

i. 1 is equals to 0 and 0 is not equals to 1 or 1 is equals to 0
ii. FALSE and TRUE or FALSE
iii. (FALSE and TRUE) or FALSE
iv. FALSE or FALSE
v. FALSE

Boolean logic truth table

Negation

NOT

x	x'
0	1
1	0

Conjunction

AND

x	y	xy
0	0	0
0	1	0
1	0	0
1	1	1

Disjunction

OR

x	y	x+y
0	0	0
0	1	1
1	0	1
1	1	1

Attention!
Think of 1s as True
And 0s as False

You should notice that getting True outcome in conjunction is hard but easy in disjunction

Which is easier to happen?

- A: If it is sunny and windy, I will go out to play
- B: If it is sunny or windy, I will go out to play
- For A, you need both conditions to be fulfilled before you do something
- For B, you need only one of the two conditions to be fulfilled before you do something.

Flow control

When the “if statement evaluates to false”, another branch will be activated automatically

This is in fact, the “diamond” in your flowcharts

Boolean logic is the bedrock of computing

It is used for controlling how code works

This is encapsulated in the ”**if…else**” statements

If (statement evaluates to **TRUE**)… {do this}

Else … {do that}

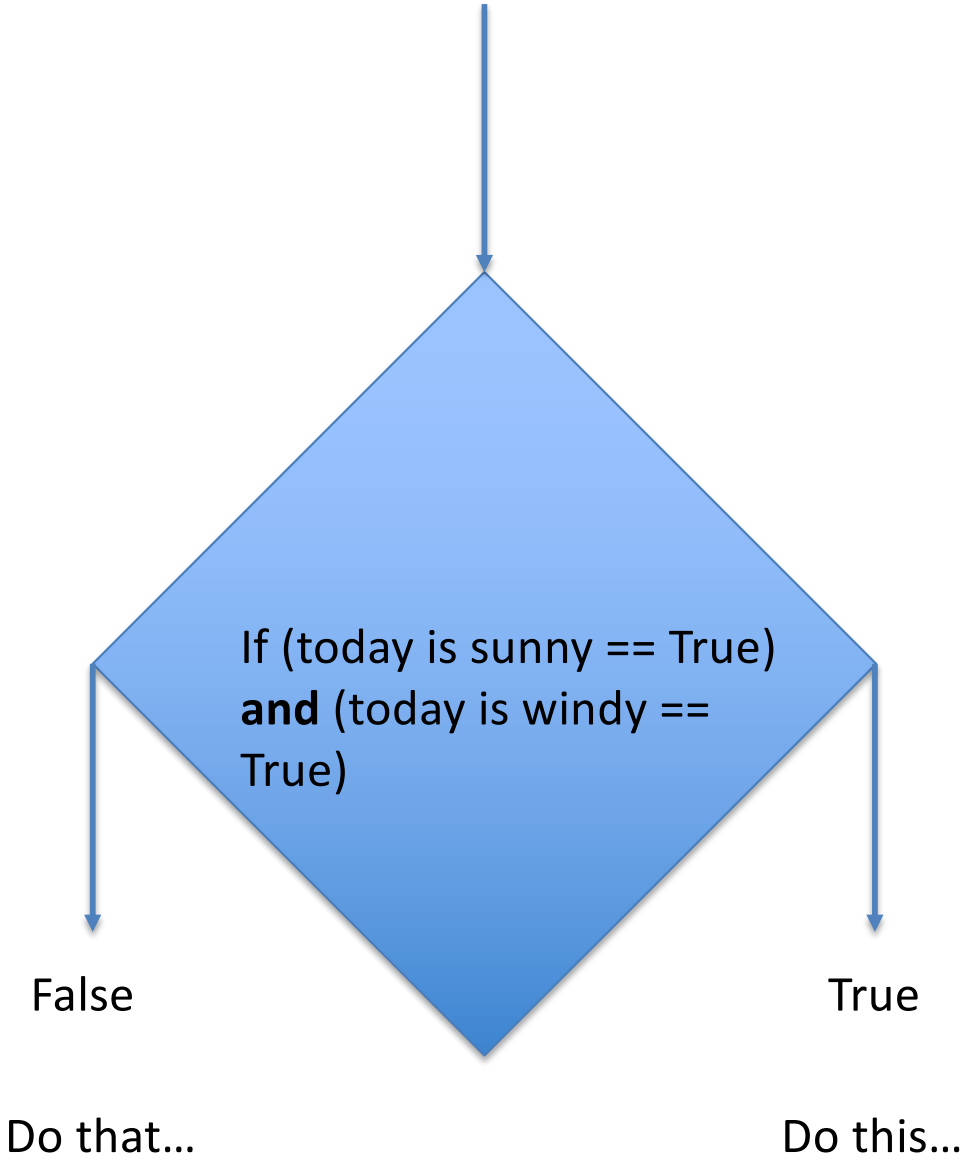
Flow control (example)

- grade = 5
- If (grade > 5) {“you got an A”} else {“you did not get an A”}
- If (grade > 5) {“you got an A”} evaluates to FALSE
- And so, the code outputs {“you did not get an A”}

Chaining

- You can use conjunctions* to make more complex expressions in a single “diamond”
- If (today is sunny == True) and (today is windy == True) then go out to play

*or a combination of conjunctions and disjunctions)



Nesting

- Multiple “if” and/or “if-else” statements can be embedded within each other
- This creates a logical chain where an inner statement is dependent on the logical statement of the outer statement

Nesting (using if)

```
If (today is sunny)
    if (today is not cloudy)
        if (today is a holiday)
            {go out to play}
```

```
If (TRUE)
    if (TRUE)
        if (TRUE)
            {go out to play}
```

'today is not cloudy' is nested under 'today is sunny'

Suppose if 'today is not sunny', then {go out to play} will not execute
Suppose if 'today is sunny' AND 'today is cloudy', {go out to play} will also not execute (you do not need to consider if 'today is a holiday')

Nesting (using if)

Notice that

```
If (TRUE)
    if (TRUE)
        if (TRUE)
            {go out to play}
```

Nesting

Can be condensed to

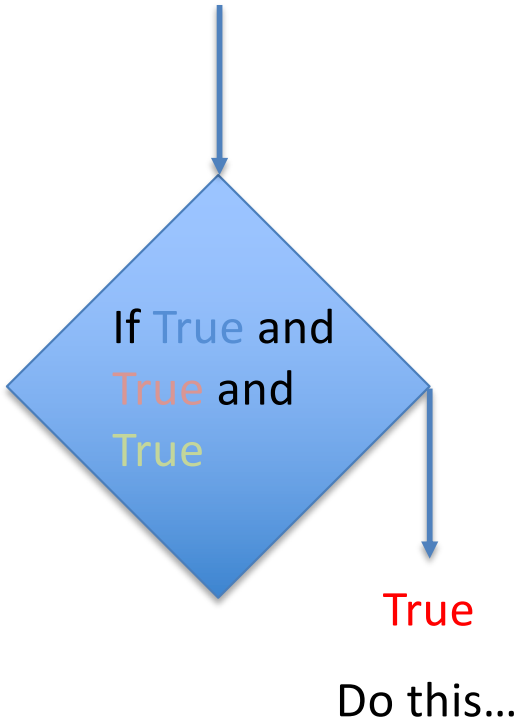
```
If (TRUE) AND (TRUE)
AND (TRUE)
        {go out to play}
```

Chaining

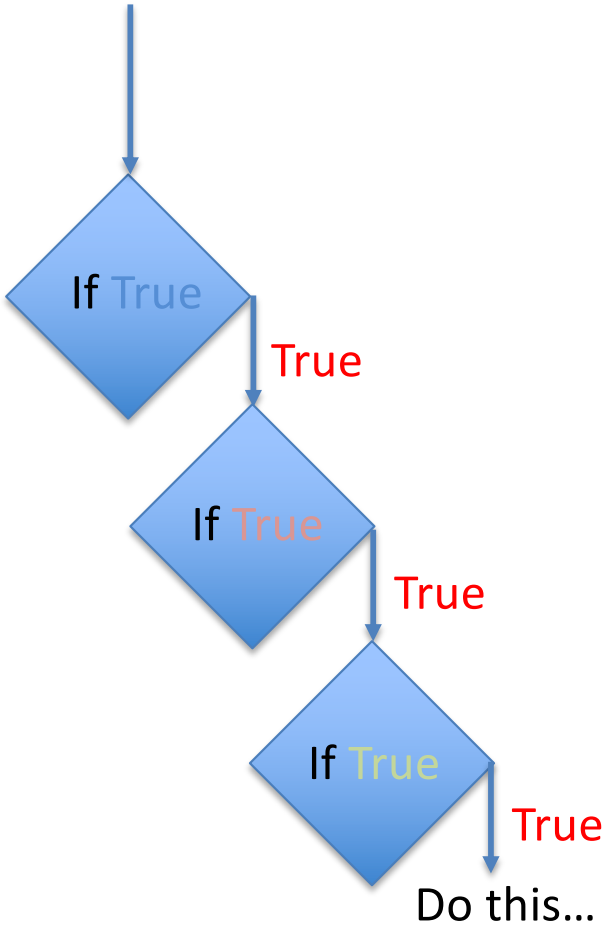
But note that this is not nesting as all three statements are evaluated simultaneously.

Nesting and chaining (in action)

Chaining



- Nesting



Nesting (using if-else)

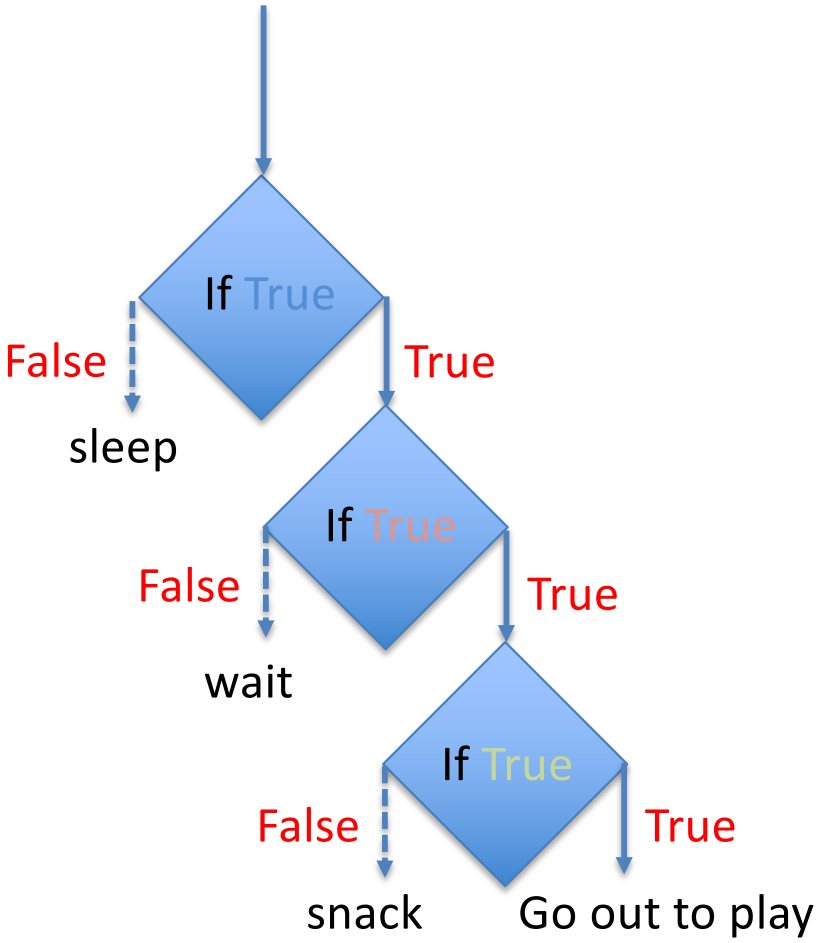
```
If (TRUE)
    if (TRUE)
        if (TRUE)
            {go out to play}
        else
            {snack}
    else
        {wait}
else
    {sleep}
```

- Here, given various scenarios, there is a different outcome:
- If TRUE, TRUE, FALSE, {snack}
- If FALSE, {sleep}
- If TRUE, FALSE, {wait}

Note that in nesting, should the outermost loop evaluate to **FALSE**, it goes directly to the else condition that is paired with it.
In this case, to {sleep}.
All the other statements associated with TRUE will not run

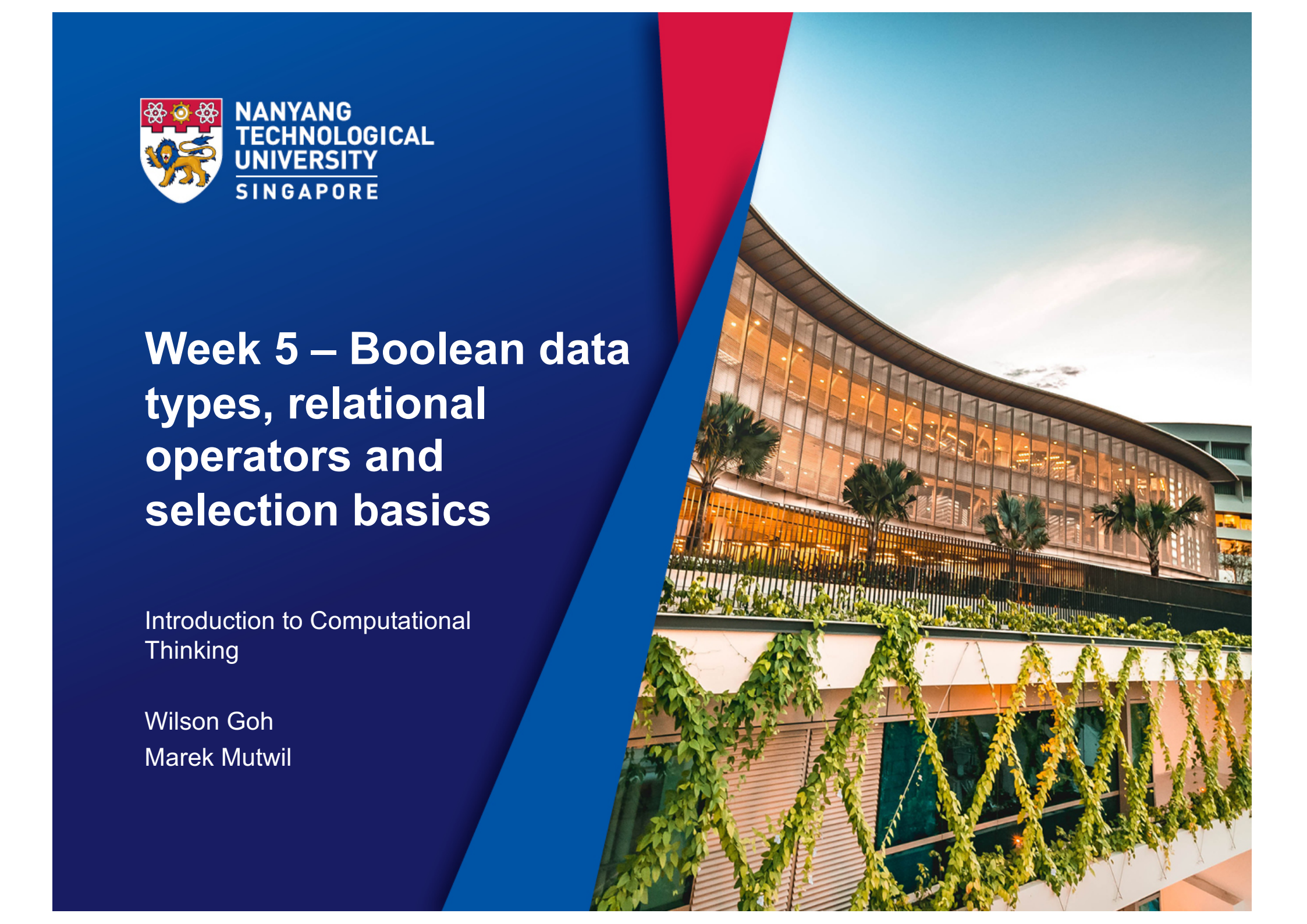
Nesting (using if-else, in action)

```
If (TRUE)
    if (TRUE)
        if (TRUE)
            {go out to play}
        else
            {snack}
    else
        {wait}
else
    {sleep}
```



READY?

Now let’s start the tutorial proper

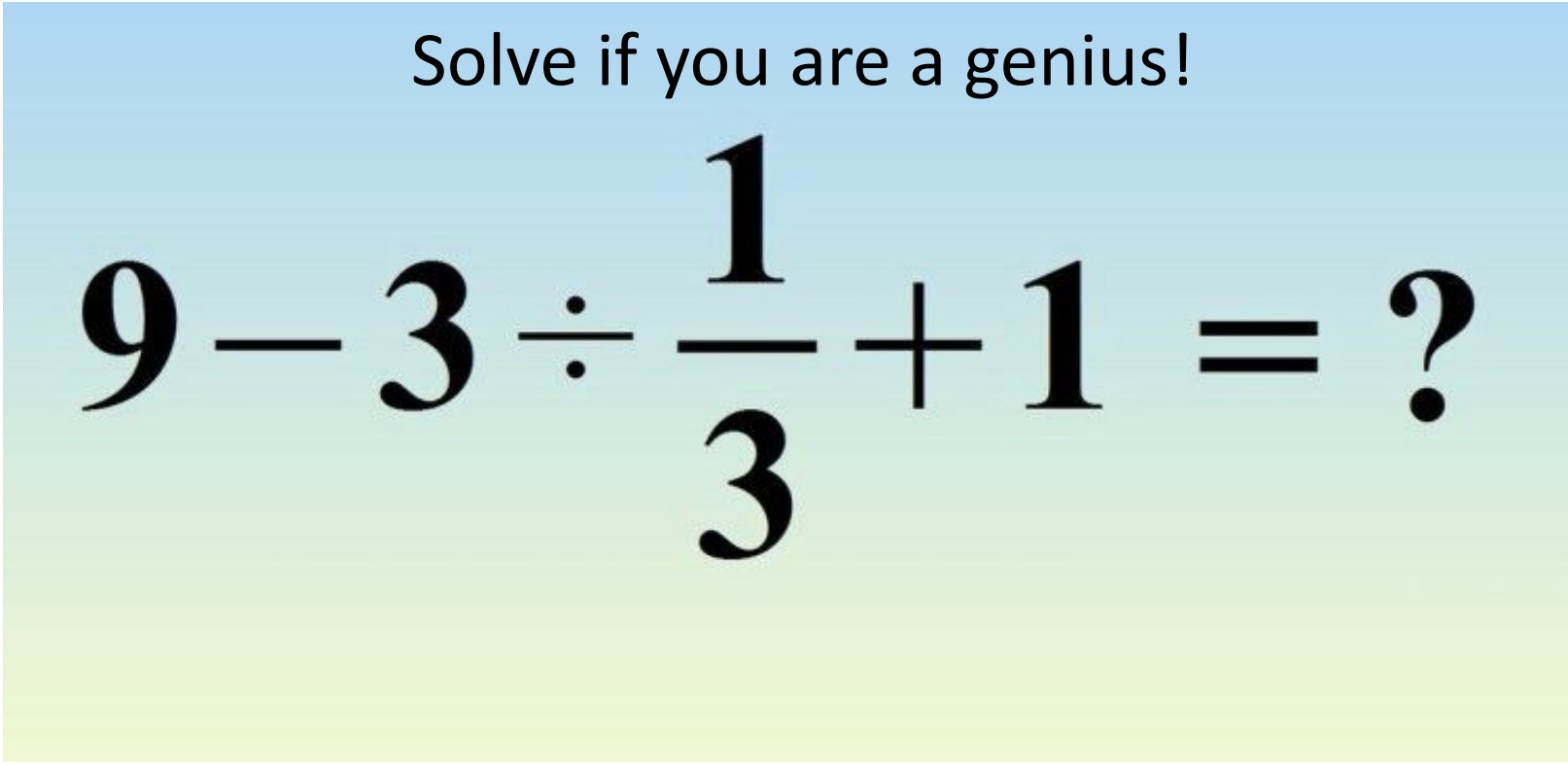


Week 5 – Boolean data types, relational operators and selection basics

Introduction to Computational Thinking

Wilson Goh
Marek Mutwil

1. Write a Python script functions.py to compute the following expressions, assuming that the angles are in radian. By importing the math package, you have access to many math-related methods or attributes, such as math.sqrt() for square root, math.pi for pi value, math.e for e value, math.log() for logarithm, math.cos(), math.sin(), math.tan() for trigonometric functions, etc.)




```
In [4]: 9-3/1/3+1
Out[4]: 9.0

In [5]: 9-3/(1/3)+1
Out[5]: 1.0
```

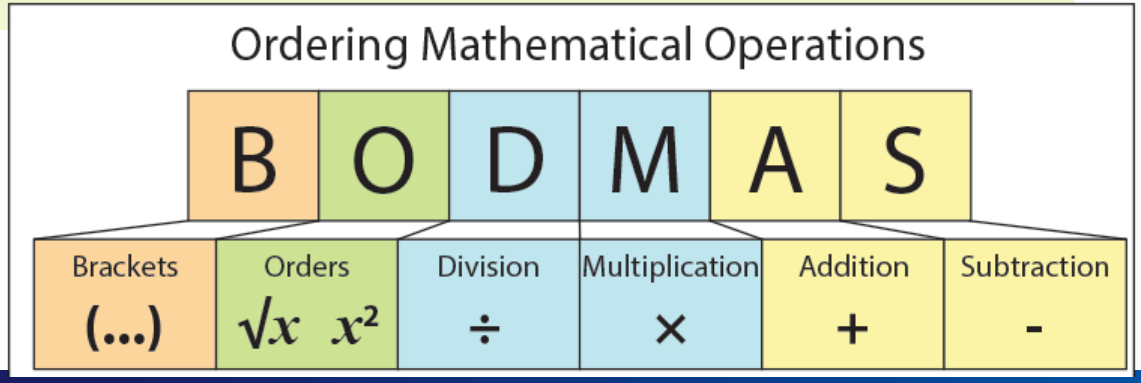
1. Write a Python script functions.py to compute the following expressions, assuming that the angles are in radian. By importing the math package, you have access to many math-related methods or attributes, such as math.sqrt() for square root, math.pi for pi value, math.e for e value, math.log() for logarithm, math.cos(), math.sin(), math.tan() for trigonometric functions, etc.)

Solve if you are a genius!

$$9-3\div\frac{1}{3}+1=?$$

```
In [4]: 9-3/1/3+1
Out[4]: 9.0

In [5]: 9-3/(1/3)+1
Out[5]: 1.0
```



1. Write a Python script functions.py to compute the following expressions, assuming that the angles are in radian. By importing the math package, you have access to many math-related methods or attributes, such as math.sqrt() for square root, math.pi for pi value, math.e for e value, math.log() for logarithm, math.cos(), math.sin(), math.tan() for trigonometric functions, etc.)

a) $$\frac{(e^{1.4} + \ln(465^2))}{\sqrt{2} + 14} + \frac{12}{\sqrt{e + 4}}$$

b) $$(-2.6)^{0.2} + \frac{e^{-\sqrt{43.3}}}{\tan(276)} + 17^{-1/7}$$

c) $$\frac{\pi^3 - 5.6^2 + 1}{1.2^{\pi/2} - \sin(43)} + \left(\frac{14.8}{5}\right)^{\pi - 1.8}$$

1. Write a Python script functions.py to compute the following expressions, assuming that the angles are in radian. By importing the math package, you have access to many math-related methods or attributes, such as math.sqrt() for square root, math.pi for pi value, math.e for e value, math.log() for logarithm, math.cos(), math.sin(), math.tan() for trigonometric functions, etc.)

a) $$\frac{(e^{1.4} + \ln(465^2))}{\sqrt{2} + 14} + \frac{12}{\sqrt{e + 4}}$$

b) $$(-2.6)^{0.2} + \frac{e^{-\sqrt{43.3}}}{\tan(276)} + 17^{-1/7}$$

c) $$\frac{\pi^3 - 5.6^2 + 1}{1.2^{\pi/2} - \sin(43)} + \left(\frac{14.8}{5}\right)^{\pi - 1.8}$$

1. Write a Python script functions.py to compute the following expressions, assuming that the angles are in radian. By importing the math package, you have access to many math-related methods or attributes, such as math.sqrt() for square root, math.pi for pi value, math.e for e value, math.log() for logarithm, math.cos(), math.sin(), math.tan() for trigonometric functions, etc.)

a) $$\frac{(e^{1.4} + \ln(465^2))}{\sqrt{2} + 14} + \frac{12}{\sqrt{e+4}}$$

(math.e**1.4+ math.log(465**2)) / (math.sqrt(2)+14) + 12/math.sqrt(math.e+4) = 5.689705917…

b) $$(-2.6)^{0.2} + \frac{e^{-\sqrt{43.3}}}{\tan(276)} + 17^{-1/7}$$

c) $$\frac{\pi^3 - 5.6^2 + 1}{1.2^{\pi/2} - \sin(43)} + \left(\frac{14.8}{5}\right)^{\pi - 1.8}$$

1. Write a Python script functions.py to compute the following expressions, assuming that the angles are in radian. By importing the math package, you have access to many math-related methods or attributes, such as math.sqrt() for square root, math.pi for pi value, math.e for e value, math.log() for logarithm, math.cos(), math.sin(), math.tan() for trigonometric functions, etc.)

a) $$\frac{(e^{1.4} + \ln(465^2))}{\sqrt{2} + 14} + \frac{12}{\sqrt{e+4}}$$

(math.e**1.4+ math.log(465**2)) / (math.sqrt(2)+14) + 12/math.sqrt(math.e+4) = 5.689705917…

b) $$(-2.6)^{0.2} + \frac{e^{-\sqrt{43.3}}}{\tan(276)} + 17^{-1/7}$$

a) 1.64373+0.7115629957j
b) 256.9957j
c) -15.342+0.65j

c) $$\frac{\pi^3 - 5.6^2 + 1}{1.2^{\pi/2} - \sin(43)} + \left(\frac{14.8}{5}\right)^{\pi - 1.8}$$

1. Write a Python script functions.py to compute the following expressions, assuming that the angles are in radian. By importing the math package, you have access to many math-related methods or attributes, such as math.sqrt() for square root, math.pi for pi value, math.e for e value, math.log() for logarithm, math.cos(), math.sin(), math.tan() for trigonometric functions, etc.)

a) $$\frac{(e^{1.4} + \ln(465^2))}{\sqrt{2} + 14} + \frac{12}{\sqrt{e+4}}$$

(math.e**1.4+ math.log(465**2)) / (math.sqrt(2)+14) + 12/math.sqrt(math.e+4) = 5.689705917…

b) $$(-2.6)^{0.2} + \frac{e^{-\sqrt{43.3}}}{\tan(276)} + 17^{-1/7}$$

(-2.6)**0.2 + (math.e**-math.sqrt(43.3) /math.tan(276)) + 17**(-1/7) = 1.64373+0.7115629957j

c) $$\frac{\pi^3 - 5.6^2 + 1}{1.2^{\pi/2} - \sin(43)} + \left(\frac{14.8}{5}\right)^{\pi - 1.8}$$

a) 156.112392j
b) 0.112824965
c) 4.586976425131735

1. Write a Python script functions.py to compute the following expressions, assuming that the angles are in radian. By importing the math package, you have access to many math-related methods or attributes, such as math.sqrt() for square root, math.pi for pi value, math.e for e value, math.log() for logarithm, math.cos(), math.sin(), math.tan() for trigonometric functions, etc.)

a) $$\frac{(e^{1.4} + \ln(465^2))}{\sqrt{2} + 14} + \frac{12}{\sqrt{e+4}}$$

(math.e**1.4+ math.log(465**2)) / (math.sqrt(2)+14) + 12/math.sqrt(math.e+4) = 5.689705917…

b) $$(-2.6)^{0.2} + \frac{e^{-\sqrt{43.3}}}{\tan(276)} + 17^{-1/7}$$

(-2.6)**0.2 + (math.e**-math.sqrt(43.3) /math.tan(276)) + 17**(-1/7) = 1.64373+0.7115629957j

c) $$\frac{\pi^3 - 5.6^2 + 1}{1.2^{\pi/2} - \sin(43)} + \left(\frac{14.8}{5}\right)^{\pi - 1.8}$$

(math.pi**3 - 5.6**2 + 1) /(1.2**(math.pi/2) - math.sin(43)) + (14.8/5)**(math.pi-1.8) = 4.58699

2. What would be the result of the following expressions?

(a) 3 == 2 + 1

(b) (3 == 2) + 1

(c) 5 < 6 or 8 > 4

(d) 5 < 10 > 4

(e) False and True and True or False or True and False

```
In [11]: False + False
Out[11]: 0

In [12]: False + True
Out[12]: 1

In [13]: True + True
Out[13]: 2
```

Increase in priority ↑

Operators	Description
()	Parentheses/ round brackets (grouping)
**	Exponentiation
+x, –x	Positive, negative
*, /, %	Multiplication, division, remainder
+, –	Addition, subtraction
<, <=, >, >=. !=, ==	Comparisons
not x	Boolean NOT
and	Boolean AND
or	Boolean OR

A	B	not A	A and B	A or B
False	False	True	False	False
False	True	True	False	True
True	False	False	False	True
True	True	False	True	True

2. What would be the result of the following expressions?

(a) 3 == 2 + 1

3 == 3

True

(b) (3 == 2) + 1

(c) 5 < 6 or 8 > 4

(d) 5 < 10 > 4

(e) False and True and True or False or True and False

```
In [11]: False + False
Out[11]: 0

In [12]: False + True
Out[12]: 1

In [13]: True + True
Out[13]: 2
```

Increase in priority ↑

Operators	Description
()	Parentheses/ round brackets (grouping)
**	Exponentiation
+x, –x	Positive, negative
*, /, %	Multiplication, division, remainder
+, –	Addition, subtraction
<, <=, >, >=. !=, ==	Comparisons
not x	Boolean NOT
and	Boolean AND
or	Boolean OR

A	B	not A	A and B	A or B
False	False	True	False	False
False	True	True	False	True
True	False	False	False	True
True	True	False	True	True

2. What would be the result of the following expressions?

(a) 3 == 2 + 1

3 == 3

True

(b) (3 == 2) + 1

False + 1

True

(c) 5 < 6 or 8 > 4

(d) 5 < 10 > 4

(e) False and True and True or False or True and False

```
In [11]: False + False
Out[11]: 0

In [12]: False + True
Out[12]: 1

In [13]: True + True
Out[13]: 2
```

Increase in priority ↑

Operators	Description
()	Parentheses/ round brackets (grouping)
**	Exponentiation
+x, –x	Positive, negative
*, /, %	Multiplication, division, remainder
+, –	Addition, subtraction
<, <=, >, >=. !=, ==	Comparisons
not x	Boolean NOT
and	Boolean AND
or	Boolean OR

A	B	not A	A and B	A or B
False	False	True	False	False
False	True	True	False	True
True	False	False	False	True
True	True	False	True	True

2. What would be the result of the following expressions?

(a) 3 == 2 + 1

3 == 3

True

(b) (3 == 2) + 1

False + 1

True

(c) 5 < 6 or 8 > 4

True or True

True

(d) 5 < 10 > 4

(e) False and True and True or False or True and False

```
In [11]: False + False
Out[11]: 0

In [12]: False + True
Out[12]: 1

In [13]: True + True
Out[13]: 2
```

Increase in priority ↑

Operators	Description
()	Parentheses/ round brackets (grouping)
**	Exponentiation
+x, –x	Positive, negative
*, /, %	Multiplication, division, remainder
+, –	Addition, subtraction
<, <=, >, >=. !=, ==	Comparisons
not x	Boolean NOT
and	Boolean AND
or	Boolean OR

A	B	not A	A and B	A or B
False	False	True	False	False
False	True	True	False	True
True	False	False	False	True
True	True	False	True	True

2. What would be the result of the following expressions?

(a) 3 == 2 + 1

3 == 3

True

(b) (3 == 2) + 1

False + 1

True

(c) 5 < 6 or 8 > 4

True or True

True

(d) 5 < 10 > 4

(5 < 10) and (10 > 4)

True and True

True

(e) False and True and True or False or True and False

```
In [11]: False + False
Out[11]: 0

In [12]: False + True
Out[12]: 1

In [13]: True + True
Out[13]: 2
```

Increase in priority ↑

Operators	Description
()	Parentheses/ round brackets (grouping)
**	Exponentiation
+x, –x	Positive, negative
*, /, %	Multiplication, division, remainder
+, –	Addition, subtraction
<, <=, >, >=. !=, ==	Comparisons
not x	Boolean NOT
and	Boolean AND
or	Boolean OR

A	B	not A	A and B	A or B
False	False	True	False	False
False	True	True	False	True
True	False	False	False	True
True	True	False	True	True

2. What would be the result of the following expressions?

(a) 3 == 2 + 1

3 == 3

True

(b) (3 == 2) + 1

False + 1

True

(c) 5 < 6 or 8 > 4

True or True

True

(d) 5 < 10 > 4

(5 < 10) and (10 > 4)

True and True

True

(e) False and True and True or False or True and False

(False and True) and (True or False) or (True and False)

False and True or False

False or False

False

```
In [11]: False + False
Out[11]: 0

In [12]: False + True
Out[12]: 1

In [13]: True + True
Out[13]: 2
```

Increase in priority ↑

Operators	Description
()	Parentheses/ round brackets (grouping)
**	Exponentiation
+x, –x	Positive, negative
*, /, %	Multiplication, division, remainder
+, –	Addition, subtraction
<, <=, >, >=. !=, ==	Comparisons
not x	Boolean NOT
and	Boolean AND
or	Boolean OR

A	B	not A	A and B	A or B
False	False	True	False	False
False	True	True	False	True
True	False	False	False	True
True	True	False	True	True

3. Write a script that tests whether the user can follow instructions. It prompts the user to enter the character ‘x’. If the user enters anything other than an ‘x’, it prints an error message - otherwise, the script does nothing.

3. Write a script that tests whether the user can follow instructions. It prompts the user to enter the character ‘x’. If the user enters anything other than an ‘x’, it prints an error message - otherwise, the script does nothing.

- What do we know?
  - A command line prompt requires the use of an <input> function and an assignment of the results of the input to some variable
  - The requirement is to check for non-equality to the letter ‘x’. If true, print a message. Where we are looking for non-equality, we use the ‘!=‘ operator
  - The requirement says to do nothing if the input is equals to x. In other words, we do not have to link the ‘if’ condition with an ‘else’ statement

Try doing this in your groups. Otherwise discuss your solutions for the next 5 minutes

3. Write a script that tests whether the user can follow instructions. It prompts the user to enter the character ‘x’. If the user enters anything other than an ‘x’, it prints an error message - otherwise, the script does nothing.

```
# Can the user follow
instructions ??
inx = input('Enter an x: ')
if inx != 'x':
    print("That was no 'x'!")
```

4. Write a script to calculate the volume of a pyramid (square base), which is 1/3 * base * height, where the base is length * width. Prompt the user to enter values for the length, width, and height, and then calculate the volume of the pyramid. When the user enters each value, she will then also be prompted for either `i' for inches or `c' for centimeters (2.54 cm = 1 inch). The script should print the volume in cubic inches. As an example, the output format can be:

```
This program will calculate the
volume of a pyramid.
Enter the length of the base: 50
Is that i or c? i
Enter the width of the base: 6.3
Is that i or c? c
Enter the height: 4
Is that i or c? i
The volume of the pyramid is
165.3543307086614 cubic inches.
```

4. Write a script to calculate the volume of a pyramid (square base), which is 1/3 * base * height, where the base is length * width. Prompt the user to enter values for the length, width, and height, and then calculate the volume of the pyramid. When the user enters each value, she will then also be prompted for either `i' for inches or `c' for centimeters (2.54 cm = 1 inch). The script should print the volume in cubic inches. As an example, the output format can be:

- What do we know?
  - Use 3 sets of inputs (length, width and height)
  - For each input, check if inch or cm. Since we want the output in inches, we convert whenever we get cm. This has to be repeated 3 times.
  - We also do not need to store the results in cm, and can write over the original variable storing the results in cm.
  - There is no need for deep nesting or branching since the 3 inputs are treated independent of each other.

Try doing this in your groups. Otherwise discuss your solutions for the next 10 minutes

4. Write a script to calculate the volume of a pyramid (square base), which is 1/3 * base * height, where the base is length * width. Prompt the user to enter values for the length, width, and height, and then calculate the volume of the pyramid. When the user enters each value, she will then also be prompted for either `i' for inches or `c' for centimeters (2.54 cm = 1 inch). The script should print the volume in cubic inches. As an example, the output format can be:

```
print('This script calculates the volume of a pyramid.')
# input and potential conversion of the length
length = float(input('Enter the length of the base: '))
if input('Is that i or c?') == 'c':
      length = length/2.54
# input and potential conversion of the width
width = float(input('Enter the width of the base: '))
if input('Is that i or c?') == 'c':
      width = width/2.54
# input and potential conversion of the height
height = float(input('Enter the height: '))
if input('Is that i or c?') == 'c':
      height = height/2.54
# computation and printing of the volume
vol = 1/3 * length * width * height
print('\nThe volume of the pyramid is ',vol ,'cubic inches.')
```

5. In chemistry, the pH of an aqueous solution is a measure of its acidity. The pH scale ranges from 0 to 14, inclusive. A solution with a pH of 7 is said to be neutral, a solution with a pH greater than 7 is basic, and a solution with a pH less than 7 is acidic. Write a script that will prompt the user for the pH of a solution, and will print whether it is neutral, basic, or acidic. If the user enters an invalid pH, an error message will be printed.

5. In chemistry, the pH of an aqueous solution is a measure of its acidity. The pH scale ranges from 0 to 14, inclusive. A solution with a pH of 7 is said to be neutral, a solution with a pH greater than 7 is basic, and a solution with a pH less than 7 is acidic. Write a script that will prompt the user for the pH of a solution, and will print whether it is neutral, basic, or acidic. If the user enters an invalid pH, an error message will be printed.

- What do we know?
  - Only 1 input to be evaluated against some conditions
  - Since pH should be bound between 0 to 14 , any input beyond this range should give an error message --- you may use an “or” control operator e.g. X < 0 or X > 14. Conversely, you can also use “and” e.g. X >= 0 and X <= 14
  - An input may be non-numeric, would you want to check for this?
  - If the input is numeric and in correct range, there are 3 possible splits of the pH, acidic, basic and neutral. There is some flexibility here..

Try doing this in your groups. Otherwise discuss your solutions for the next 10 minutes

5. In chemistry, the pH of an aqueous solution is a measure of its acidity. The pH scale ranges from 0 to 14, inclusive. A solution with a pH of 7 is said to be neutral, a solution with a pH greater than 7 is basic, and a solution with a pH less than 7 is acidic. Write a script that will prompt the user for the pH of a solution, and will print whether it is neutral, basic, or acidic. If the user enters an invalid pH, an error message will be printed.

```
# Prompts the user for the pH of a solution and prints
# whether it is basic , acidic , or neutral
ph = float(input('Enter the pH of the solution : '))
if ph < 0 or ph > 14:
    print('Error in pH!')
elif ph < 7:
    print('It is acidic')
elif ph > 7:
    print('It is basic')
else:
    print('It is neutral') # no need to write " elif ph ==7"
```

The final situation is implicit hence no need to specify it explicitly as ph==7

6. Write a script that will first prompt the user for a distance in kilometers, with error check that the input distance can't be negative. Then, the script will print the cost of the travel by walking, by motorcycle, by car and by plane, knowing that:

- walking, motorcycle, car and plane travel cost 0, 0.22, 0.26 and 0.78 SGD per kilometer, respectively;
- walking can't be done for more than 20 kilometers, motorcycle travel can't be done for more than 200 kilometers, car travel can't be done for more than 800 kilometers, plane travel can't be done for less than 100 kilometers.

For example, the output will look like this:

```
Enter the distance in kilometer: 12
The cost by walking is free!
The cost by motorcycle is 2.6 SGD
The cost by car is 3.1 SGD
You travel is too short to be done by plane
```

6. Write a script that will first prompt the user for a distance in kilometers, with error check that the input distance can't be negative. Then, the script will print the cost of the travel by walking, by motorcycle, by car and by plane, knowing that:

- walking, motorcycle, car and plane travel cost 0, 0.22, 0.26 and 0.78 SGD per kilometer, respectively;
- walking can't be done for more than 20 kilometers, motorcycle travel can't be done for more than 200 kilometers, car travel can't be done for more than 800 kilometers, plane travel can't be done for less than 100 kilometers.

- What do we know?
  - To catch a negative value, we only need to check if the distance is less than 0, if true, we return an error, else we continue running the code.
  - Although it is possible to simply calculate the cost of any given distance for all 3 travel modes, we need to input a condition that determines the limits of each mode e.g. if X > 20, we do not consider walking a feasible option and thus return a message indicating this.
  - The price of each travel mode and its respective limitations can be determine independently of each other (no nesting and branching required).

Try doing this in your groups. Otherwise discuss your solutions for the next 10 minutes

6. Write a script that will first prompt the user for a distance in kilometers, with error check that the input distance can't be negative. Then, the script will print the cost of the travel by walking, by motorcycle, by car and by plane, knowing that:

- walking, motorcycle, car and plane travel cost 0, 0.22, 0.26 and 0.78 SGD per kilometer, respectively;
- walking can't be done for more than 20 kilometers, motorcycle travel can't be done for more than 200 kilometers, car travel can't be done for more than 800 kilometers, plane travel can't be done for less than 100 kilometers.

```
distance = float(input('Enter the distance in kilometers:'))

if distance <0:
    print('Error, your distance is lower than 0!')
else:
    if distance <= 20: # travel by walking
        print('The cost by walking is free!')
    else:
        print('Your travel is too long to be done by walking')
    if distance <= 200:      # travel by motorcycle
        cost = distance * 0.22
        print('The cost by motorcycle is ',cost ,'SGD')
    else:
        print('Your travel is too long for the motorcycle')
    if distance <= 800: # travel by car
        cost = distance * 0.26
        print('The cost by car is ',cost ,'SGD')
    else:
        print('Your travel is too long to be done by car')
    # travel by plane
    if distance >= 100:
        cost = distance * 0.78
        print('The cost by plane is ',cost ,'SGD')
    else:
        print('Your travel is too short to be done by plane')
```

7. Write a script that will prompt the user for an amount in Singapore Dollars, and then to which foreign currency he would like to convert this amount ('E' for Euros, 'U' for US Dollars, 'J' for Japanese Yen). The script will then print the original amount and the converted amount. The output can look like this:

```
Enter the amount in Singapore Dollars: 29.340
Do you want to convert this amount in Euros (E), US Dollars (U)
or Japanese Yen (J)? U
29.34 Singapore Dollars equals 21.603042 US Dollars
```

The program must check that the user didn't enter a negative amount. You can check online for the latest currency rates.

7. Write a script that will prompt the user for an amount in Singapore Dollars, and then to which foreign currency he would like to convert this amount ('E' for Euros, 'U' for US Dollars, 'J' for Japanese Yen). The script will then print the original amount and the converted amount. The output can look like this:

- What do we know?
  - To catch a negative value, we only need to check if the distance is less than 0, if true, we return an error, else continue running.
  - The input amount is always assumed to start in SGD, to be converted to something else.
  - The option to select E, U or J, will independently create 3 code blocks that will each do its own conversion and return an output.
  - Currency is a float variable (can have cents). Do not set input to integer.

Try doing this in your groups. Otherwise discuss your solutions for the next 10 minutes

7. Write a script that will prompt the user for an amount in Singapore Dollars, and then to which foreign currency he would like to convert this amount ('E' for Euros, 'U' for US Dollars, 'J' for Japanese Yen). The script will then print the original amount and the converted amount. The output can look like this:

```
Enter the amount in Singapore Dollars: 29.340
Do you want to convert this amount in Euros (E), US Dollars (U)
or Japanese Yen (J)? U
29.34 Singapore Dollars equals 21.603042 US Dollars
```

The program must check that the user didn't enter a negative amount. You can check online for the latest currency rates.

```
amount = float(input('Enter the amount in Singapore Dollars: '))

if amount <0: # error check
    print('Error, the amount is negative ')

else: # if no error when inputting the amount
    choice = input('Convert to Euros(E), US Dollars(U) or Japanese Yens(J)?')
    # conversion to Euros
    if choice == 'E':
        converted_amount = amount * 0.6716
        print(amount, 'Singapore Dollars equals ', converted_amount, ' Euros ')
    # conversion in US Dollars
```

8. Write a script that will print a list consisting of “cylinder”, “circle”, and “rectangle”. It prompts the user to choose one, prompts the user for the appropriate quantities (e.g., the radius of the circle) and then prints its area. If the user enters an invalid choice, the script simply prints an error message. By importing the math package, you have access to the value of pi via math.pi. The script should use a nested if-else statement to accomplish this. Here are two examples of running it (units are assumed to be inches).

```
Menu
1. Cylinder
2. Circle
3. Rectangle
Please choose one: 2
Enter the radius of the circle: 4.1
The area is 52.81017250684441

Menu
1. Cylinder
2. Circle
3. Rectangle
Please choose one: 3
Enter the length: 4
Enter the width: 6
The area is 24.0
```

8. Write a script that will print a list consisting of "cylinder", "circle", and "rectangle". It prompts the user to choose one, prompts the user for the appropriate quantities (e.g., the radius of the circle) and then prints its area. If the user enters an invalid choice, the script simply prints an error message. By importing the math package, you have access to the value of pi via math.pi. The script should use a nested if-else statement to accomplish this. Here are two examples of running it (units are assumed to be inches).

- What do we know?
  - If you call in a library using import, accessing its methods/attributes requires using the form <library name>.<method name> e.g. math.pi will access the value of pi stored in the math library.
  - Different shape have different quantities for calculating area
  - The choice of what type of shape can be evaluated via the if-elif-else control statements.
  - The ordering of the shapes do not matter.

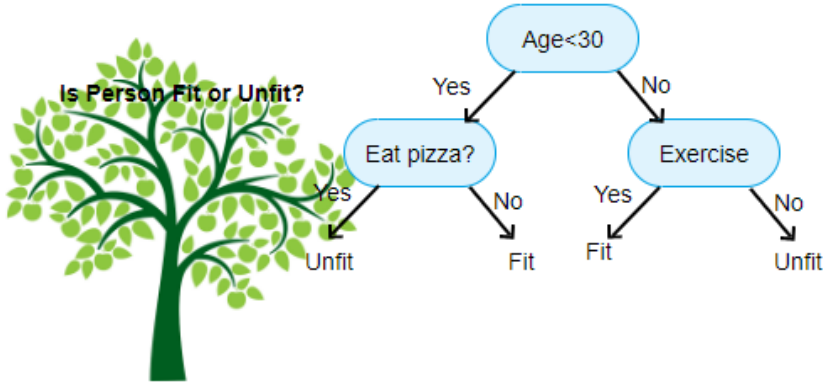
Try doing this in your groups. Otherwise discuss your solutions for the next 10 minutes

8. Write a script that will print a list consisting of "cylinder", "circle", and "rectangle". It prompts the user to choose one, prompts the user for the appropriate quantities (e.g., the radius of the circle) and then prints its area. If the user enters an invalid choice, the script simply prints an error message. By importing the math package, you have access to the value of pi via math.pi. The script should use a nested if-else statement to accomplish this. Here are two examples of running it (units are assumed to be inches).

```
import math
print('Menu')
print('1. Cylinder')
print('2. Circle')
print('3. Rectangle')
sh = input('Please choose one:')
if sh == '1':
      rad = float(input('Enter the radius of the cylinder : '))
      ht = float(input('Enter the height of the cylinder : '))
      print('The surface area is ', 2 * math.pi*rad*ht)
elif sh == '2':
      rad = float(input('Enter the radius of the circle: '))
      print('The area is ', math.pi*rad*rad)
elif sh == '3':
      len = float(input('Enter the length: '))
      wid = float(input('Enter the width: '))
      print('The area is ', len*wid)
else :
      print('Error! Not a valid choice.')
```

Wilson's challenge

- The decision tree is one of the most simple yet most useful (explainable) machine learning models.




```
If (Age < 30):
        If (eat pizza == Yes):
        <Unfit>
        else:
        <Fit>
else:
        if (exercise == Yes):
        <Fit>
        else:
        <Unfit>
```

Wilson's challenge

Outlook	Temperature	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes

- Looking at the outlook, temperate, humidity and windiness, derive a set of relationships that allows you to decide whether or not to play golf
- You will need to use nested "if-else" statements

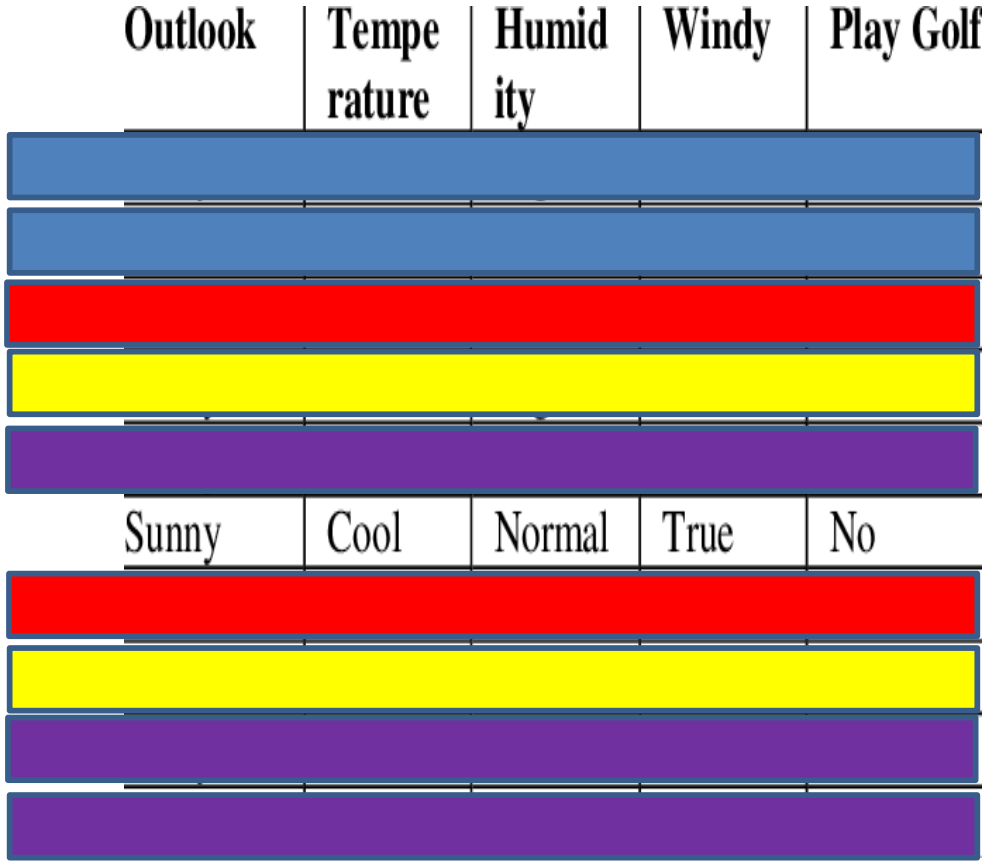
```
…..
If (outlook == ‘Rainy’ and temperature == ‘hot’
and humidity == ‘high’ and …):
        < no golf >
elif
…
else
…..
```

One way is to write it all manually

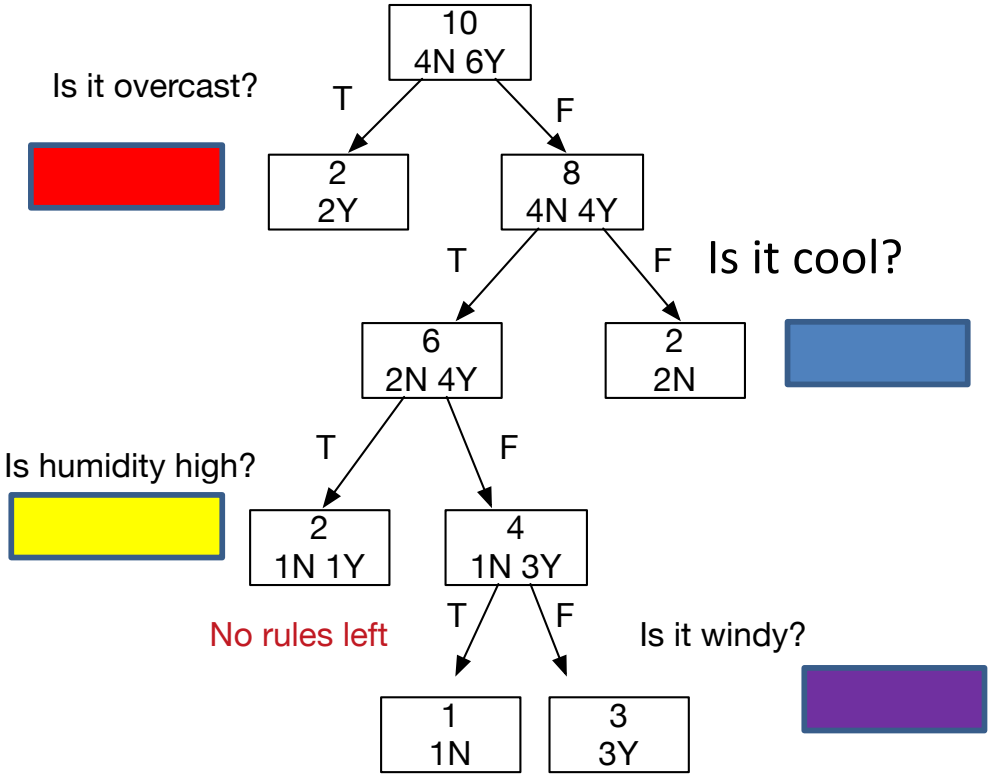
But this is too explicit. And does not factor in actual logic involved in the decision-making process.

It is also not efficient code.

Another way is to establish rules



A useful rule is one that leads towards



Using rules is not perfect but it is elegant

Motif search challenge (take a look at these --- implementations in R

- Jia Min

https://qrgo.page.link/wHxkx

- Priscilla

https://qrgo.page.link/7CLBU