APS106

UNIVERSITY OF TORONTO

String Comparisons and More “if” Statements.

Week 2 | Lecture 1 (2.1.2)

if nothing else, write #cleancode

Today's Content

- **Lecture 2.1.1**
  - Booleans, Logic, & Conditional if Statements
- **Lecture 2.1.2**
  - **String Comparisons and More on if Statements**

RECAP: Relational Operators

- Relational (or comparison) operators take two values (examples: int, float, str) and produce a bool value (True or False)

Description	Operator	Example	Result
Less than	<	3<4	True
Greater than	>	3>4	False
Equal to	==	3==4	False
Less than or equal to	<=	3<=4	True
Greater than or equal to	>=	3>=4	False
Not equal to	!=	3!=4	True

Boolean Expressions

Boolean Values

Python uses == for equality, because = is used for assignment

String Comparisons

- Boolean comparisons can also be applied to strings, whether single characters or sets of characters
- Compare two strings by their **dictionary order**, comparing letter by letter

Description	Operator	Example	Result of example
equality	==	'cat' == 'cat'	True
inequality	!=	'cat' != 'Cat'	True
less than	<	'A' < 'a'	True
greater than	>	'a' > 'A'	True
less than or equal	<=	'a' <= 'a'	True
greater than or equal	>=	'a' >= 'A'	True

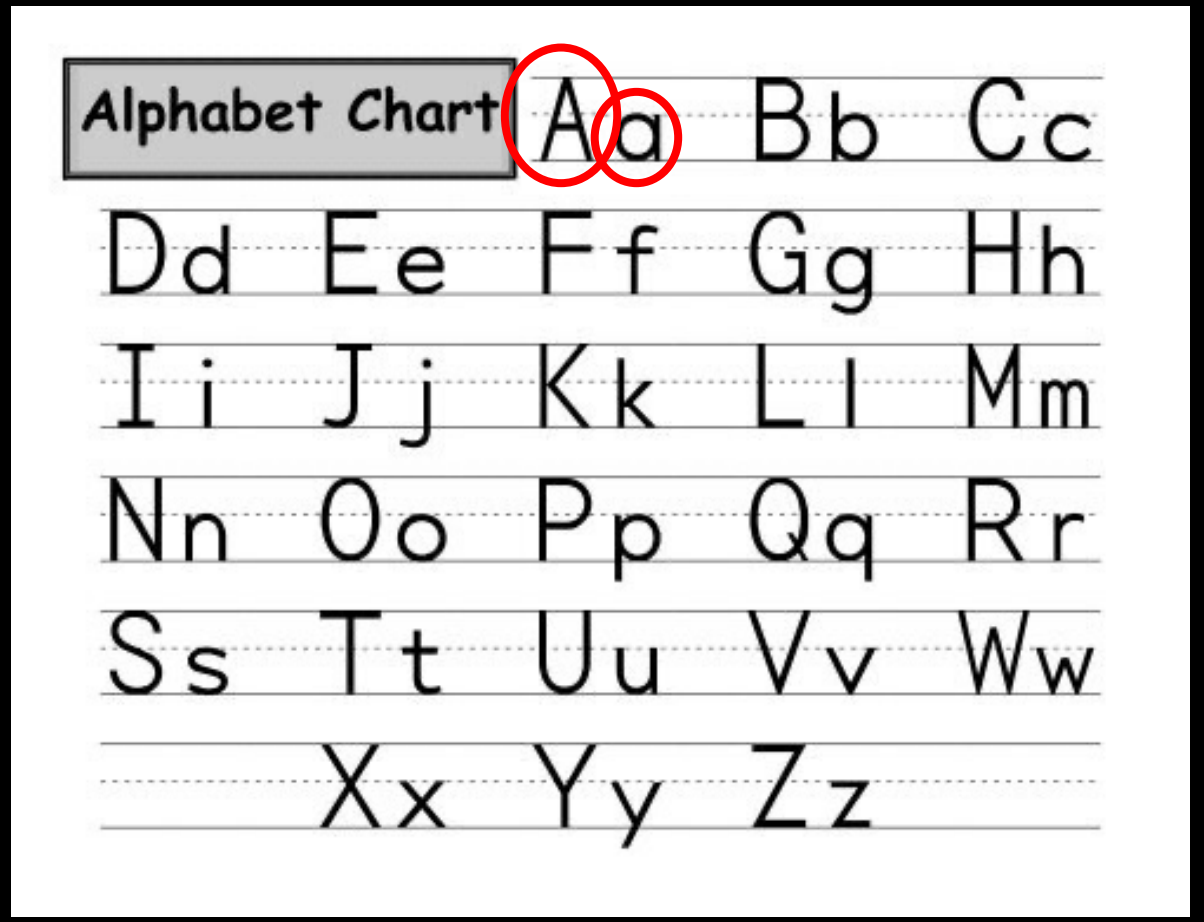
Strings as Integers (ASCII Encoding)

- Each character in a string is actually represented by integers following the ASCII encoding
  - American Standard Code for Information Interchange
- All uppercase letters come before all lowercase letters
  - Uppercase "Z" is less than lowercase "a"

Code	Char	Code	Char	Code	Char	Code	Char	Code	Char	Code	Char
32	[space]	48	0	64	@	80	P	96	`	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(	56	8	72	H	88	X	104	h	120	x
41	)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[	107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	|
45	-	61	=	77	M	93	]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	[backspace]

Strings as Integers (ASCII Encoding)

0-9
48-57



A-Z
65-90

a-z
97-122

Strings as Integers (ASCII Encoding)

- When you compare two strings, what you are really doing is comparing their numerical representations
- For example in ASCII the characters ‘a’ and ‘w’ are encoded as 97 and 119, respectively
  - The comparison ‘a’ > ‘w’ would translate to 97 > 119, evaluating to False

0-9	A-Z	a-z
48-57	65-90	97-122

ASCII Encoding

- To obtain the ASCII (integer) representation, we use the built-in function ord

```
>>> ord('a')
97
```

- To convert from ASCII integer representation back to a string, we use the built-in function chr
  - Think "character" (i.e. what is this integer's character)

```
>>> chr(97)
'a'
```

Strings and Logic Operators

- We can use string comparisons in Boolean/logical expressions:

```
>>> 50 > 5 and 't' > 'a'
True
>>> 'ABC' > 'abc' #remember ord('A') < ord('a')
False
>>> 'abcdefghijklmnopqrstuvwxyz' > 'z' #compares one at a time
False
>>> 'ab' > 'a' #if first part is equal, longer is greater
True
```

Testing for Substrings

- The in operator provides another way to check whether a string appears inside another string
  - Results in a Boolean (True or False)

```
>>> 'c' in 'aeiou'
False
>>> 'cad' in 'abracadabra'
True
>>> 'zoo' in 'ooze'
False
```

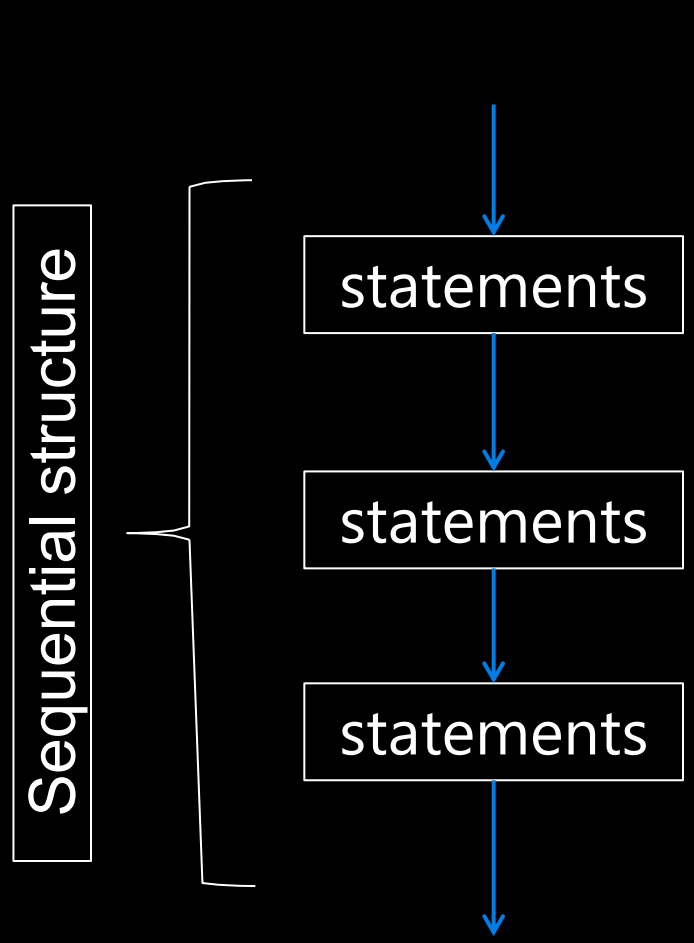
Let's Code!

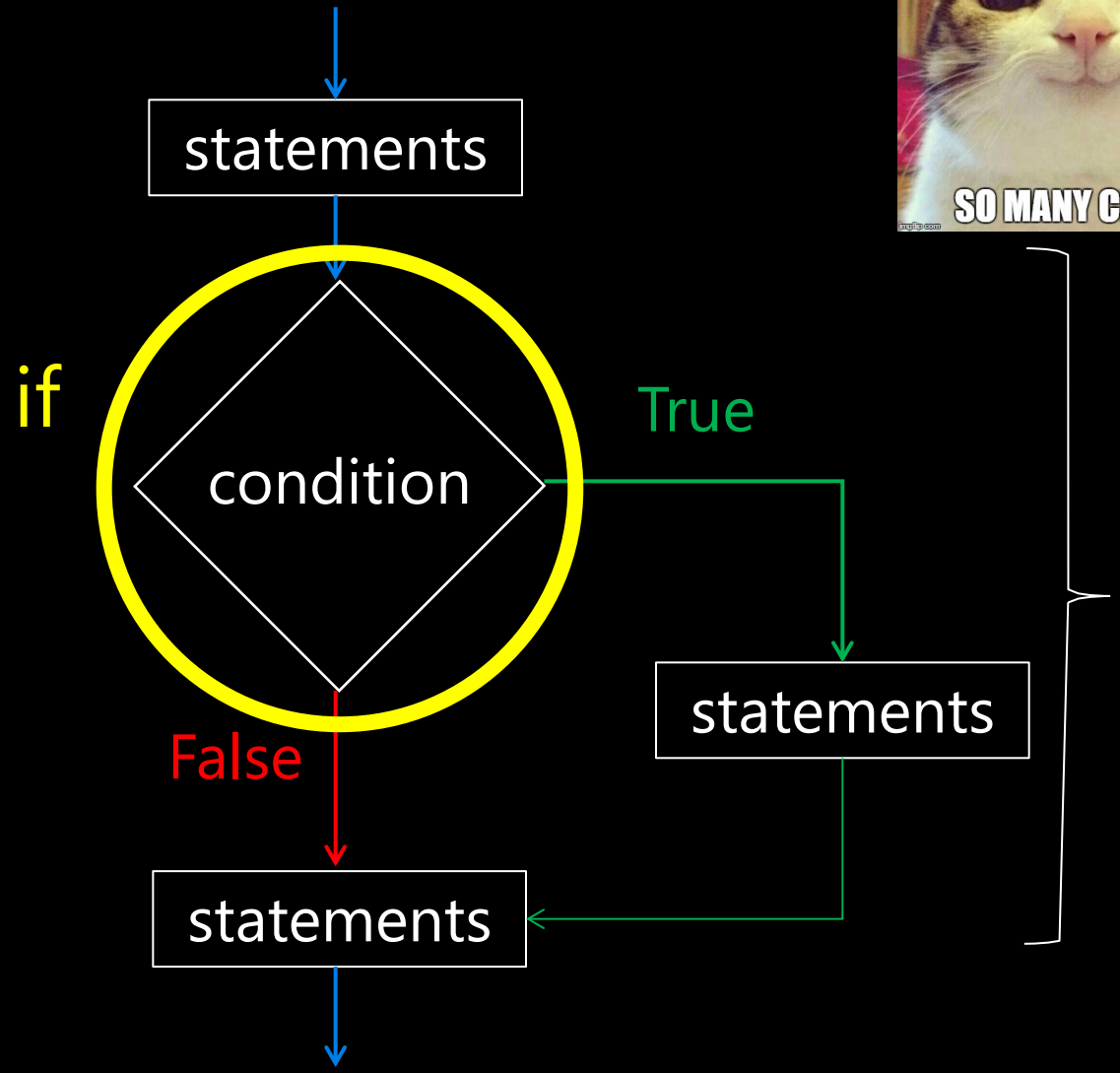
- Let's take a look at how this works in Python!
  - String comparisons

Open your notebook

Click Link:
1. String Comparisons

RECAP: Making Choices

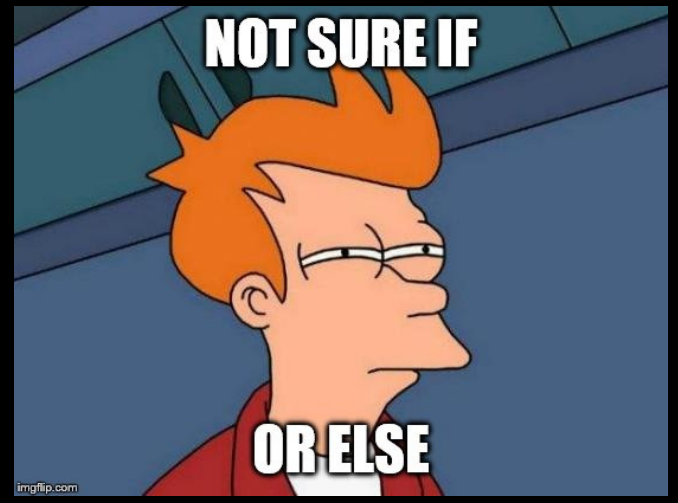




RECAP: Adding the else statement

- A more general form of the if conditional statement is:

```
if expression:
    body1
else:
    body2
```



- ONLY 1 of body1 or body2 will be executed.
  - if statement is True, executes body1
  - if statement is False, executes body2

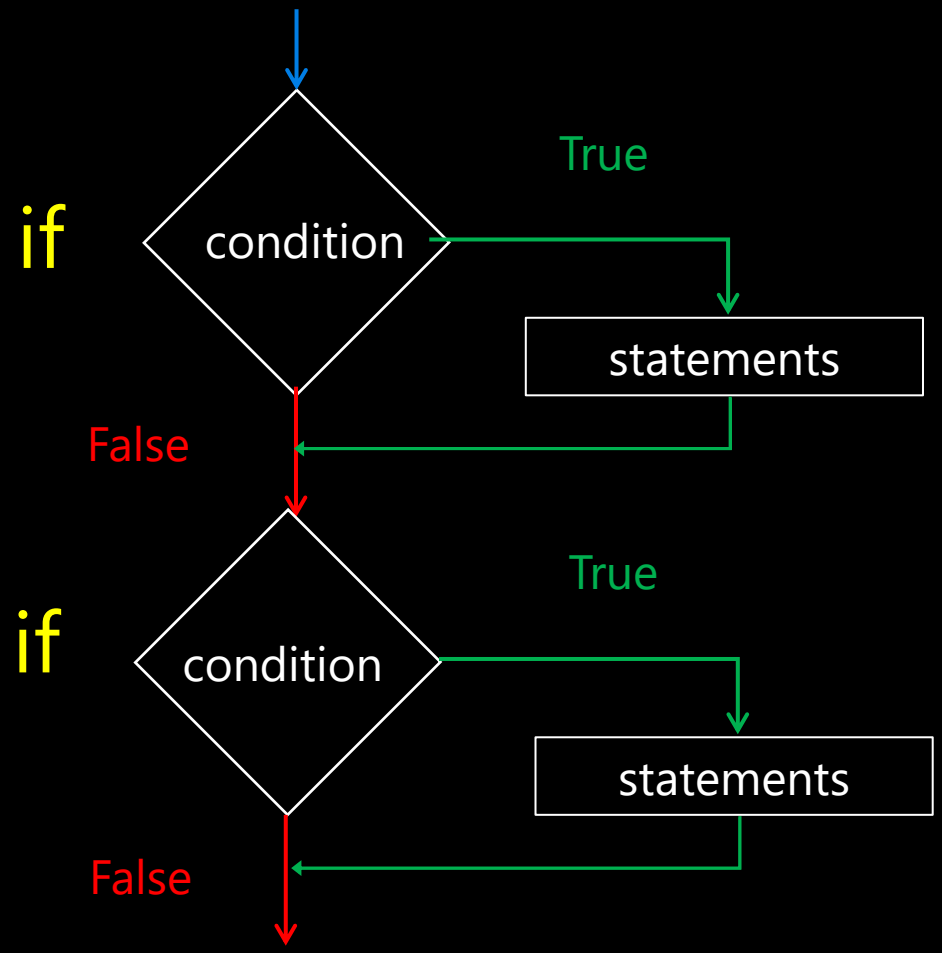
Adding the elif (else if) statement

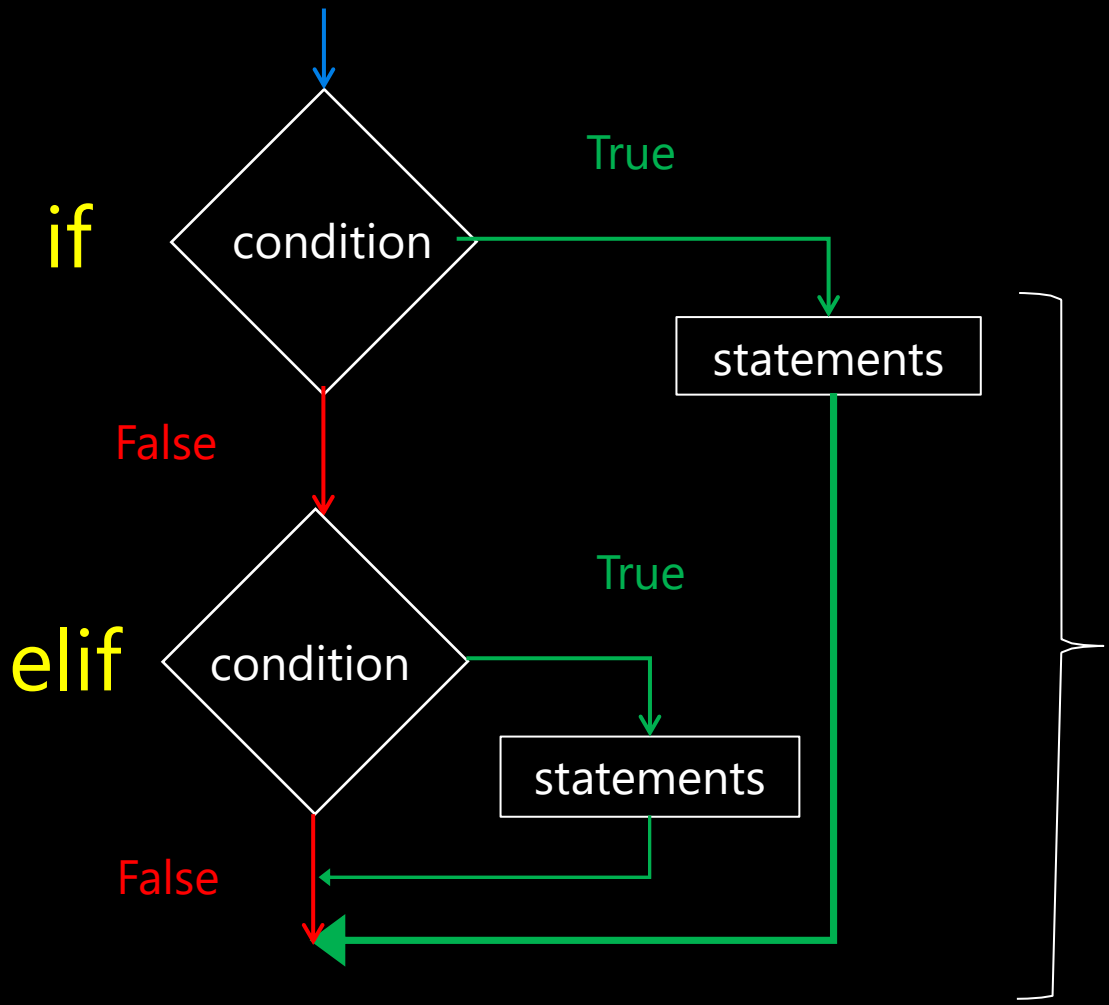
- The most general form of the if conditional statement is:

```
if condition1:
    body1
elif condition2:
    body2
elif conditionN:
    bodyN
else:
    other_body
```

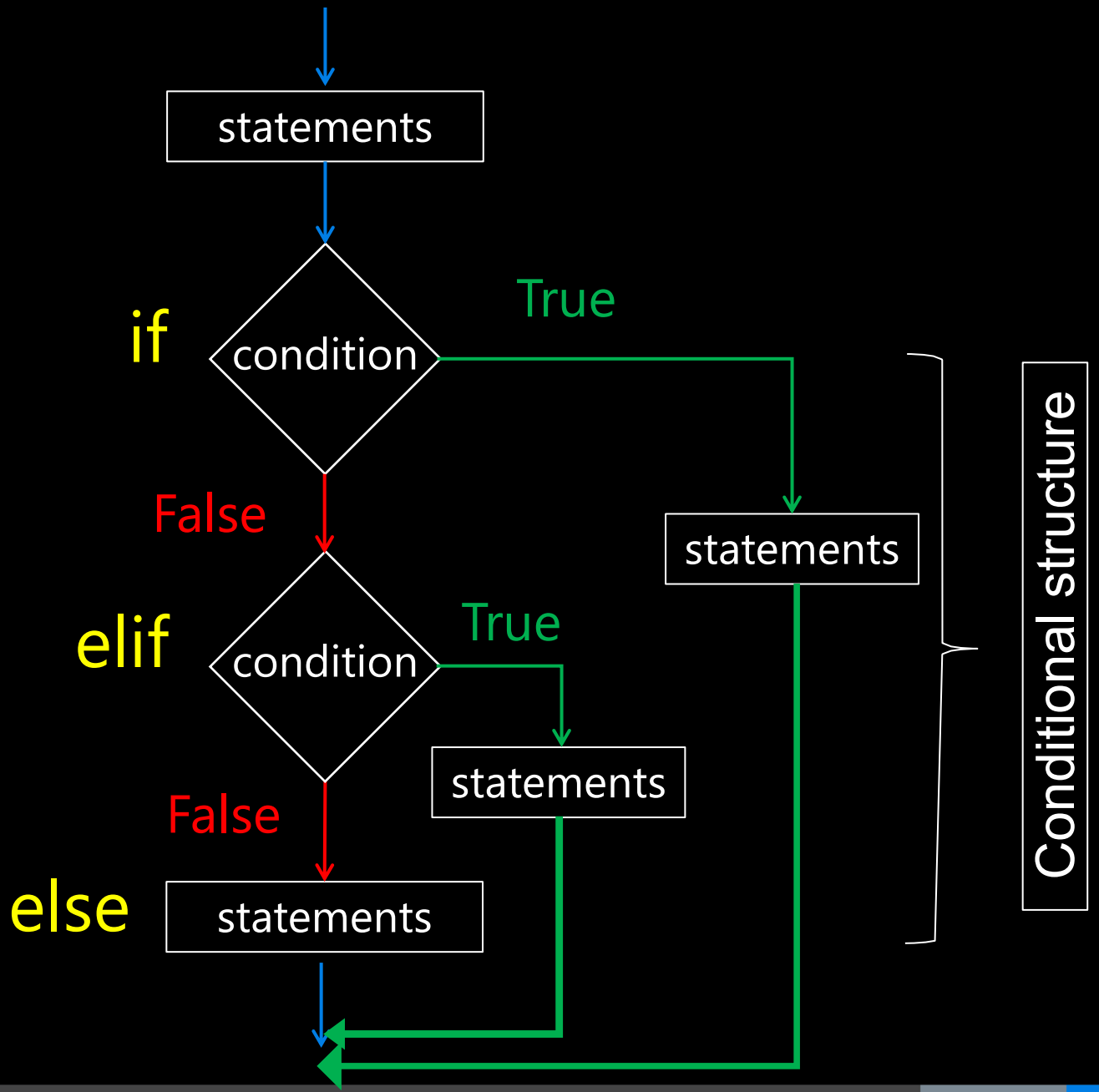
- Note the colons (:) and the indents!
- ONLY 1 body will be executed.
  - if statement is True, execute body1, exits if structure
  - if statement is False, continue to elif statement
  - elif statement is True, execute elif body, exits if structure
  - elif statement is False, continue to next elif statement
  - All if's and elif's are False, execute else statement

Multiple if vs if-elif



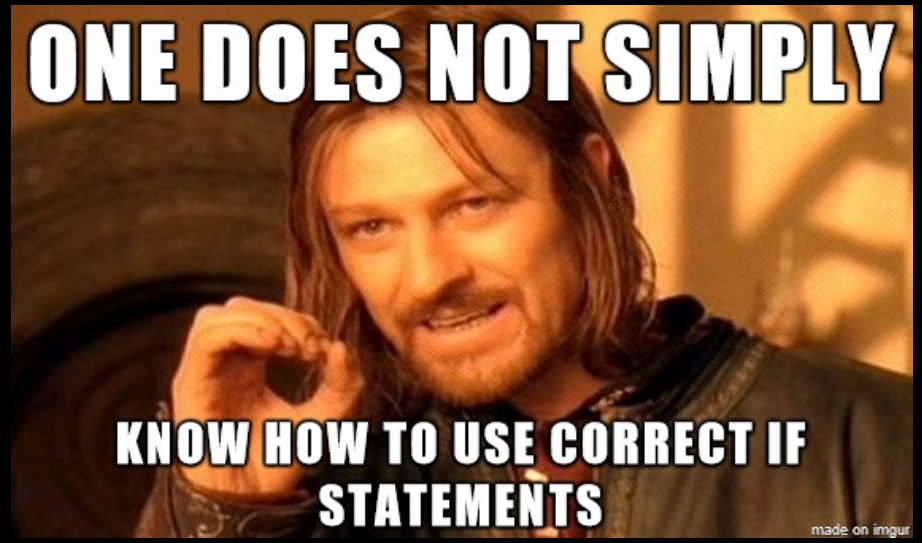


Making Choices



Let's Code!

- Let's take a look at how this works in Python!
  - if-elif-else statements



Open your notebook

Click Link:
2. More on if Statements

Nested if Statements

- It is possible to place an if statement within the body of another if statement

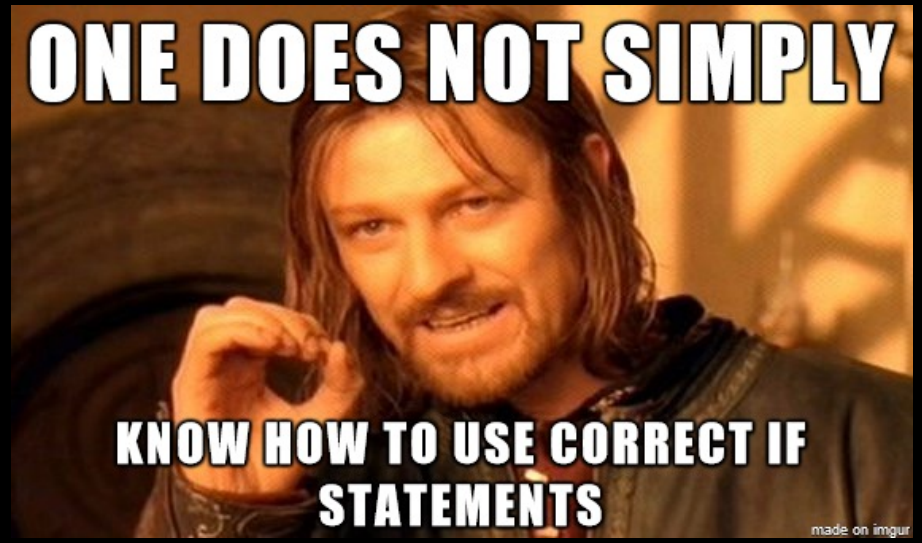
```
if precipitation:
    if temperature > 0:
        print("Bring your umbrella!")
    else:
        print("Wear your boots and winter coat!")
```

- Equivalent code:

```
if precipitation and temperature > 0:
    print("Bring your umbrella!")
elif precipitation:
    print("Wear your boots and winter coat!")
```

Choose your own adventure!

- Let's take a look at how this works in Python!
  - Nested if statements



Open your notebook

Click Link:
2. Nested ifs

APS106

UNIVERSITY OF TORONTO

String Comparisons and More “if” Statements.

Week 2 | Lecture 1 (2.1.2)

if nothing else, write #cleancode