Booleans, Checks and if's. — Chapter 5 in Book.

Often in your code you want to check if something is True or False before doing something else.

Wordle inputs:

To play wordle you input a 5 letter word. BUT before checking if you got it right your code might want to check if it's 5 letters.

one equals defines a variable.

```
my_word = "cat"
length_of_word = len(my_word)

Print(length_of_word == 5)
```

two equals checks the condition.

FALSE.

The two things have to match EXACTLY.

```
name = "Joanna"
name == "joanna"  ⟶ False.
```

Other Comparisons:

==	is equal
!=	is not equal
<	less than
>	greater than
<=	less than or equal
>=	greater than or equal.

AND — checkes TWO (all) True

OR — checks at least ONE True.

Say we are looking for an age range: between 15 and 30

age = 21
age >= 15 and age <= 30

TRUE

age < 15 or age > 30

what does this test?
Tests if age is OUTSIDE THE RANGE
result? FALSE.

OTHER FUNCTIONS — can return True or False.

my_names = ["Joanna", "Ross", "Joanna", "Bongo"]

"Joanna" in my_names True
"Bella" in my_names False
"ross" in my_names False has to be exact!!

Sometimes we want to set variables to a boolean value.

can_edit = False .... so throughout the program you can check this.

If statements check to see if something is True before running the code.

```
if <conditional test>:
    do stuff.
```

Indentation just like for loops.

Guess my word:

```
my_word = "cat"

your_guess = "dog"

if my_word == your_guess:
    Print("You Win")
```

But shouldn't we print something if you lose?

```
my_word = "cat"
your_guess = "dog"

if my_word == your_guess:
    print("You Win")
else:
    print("Try again")
```