# coding in python

week 2

# this week's topic

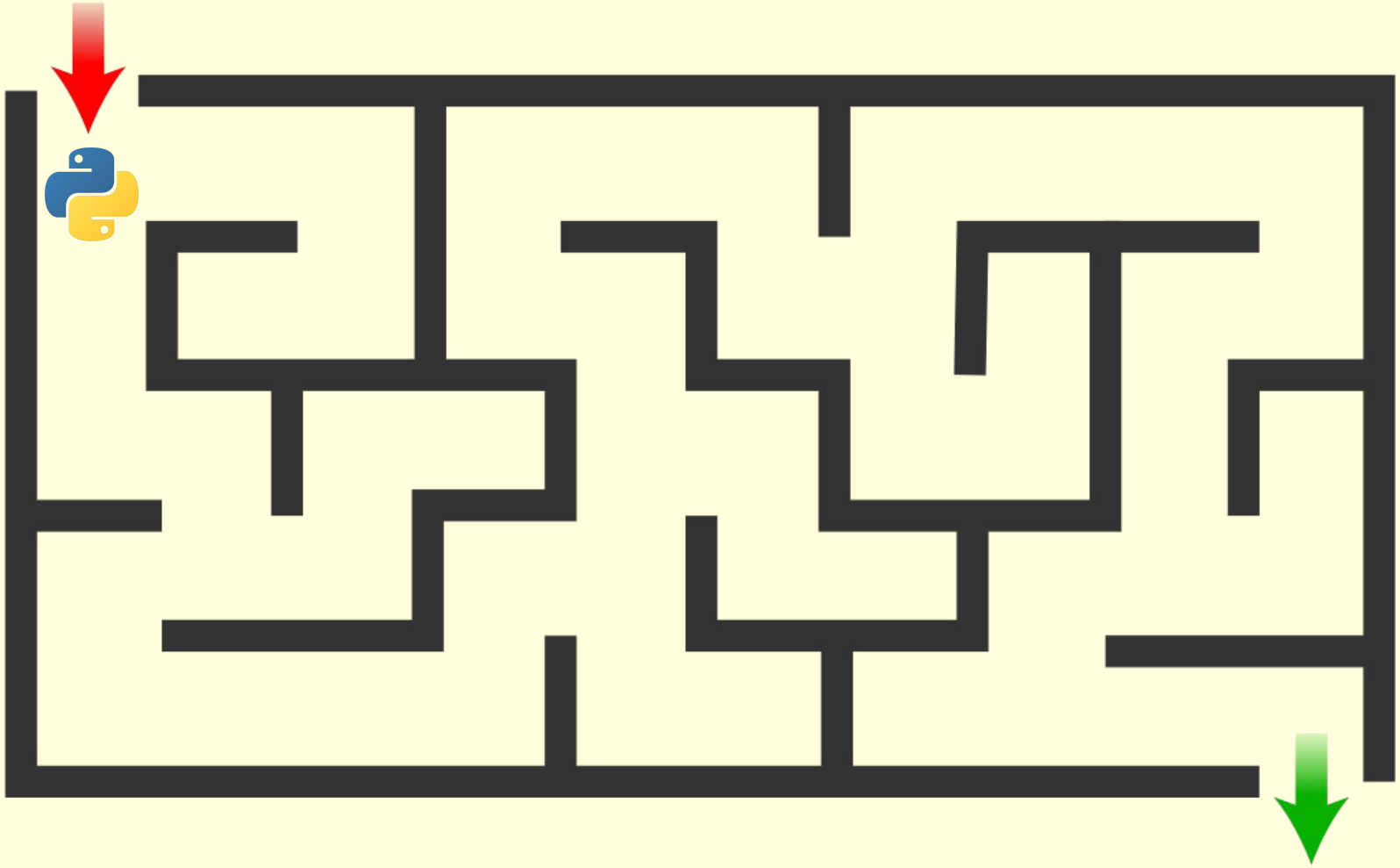
- we will be learning about making decisions using python
- to make decisions, we use *if* and *else*
    - will also learn about *elif*, which is basically just a form of if

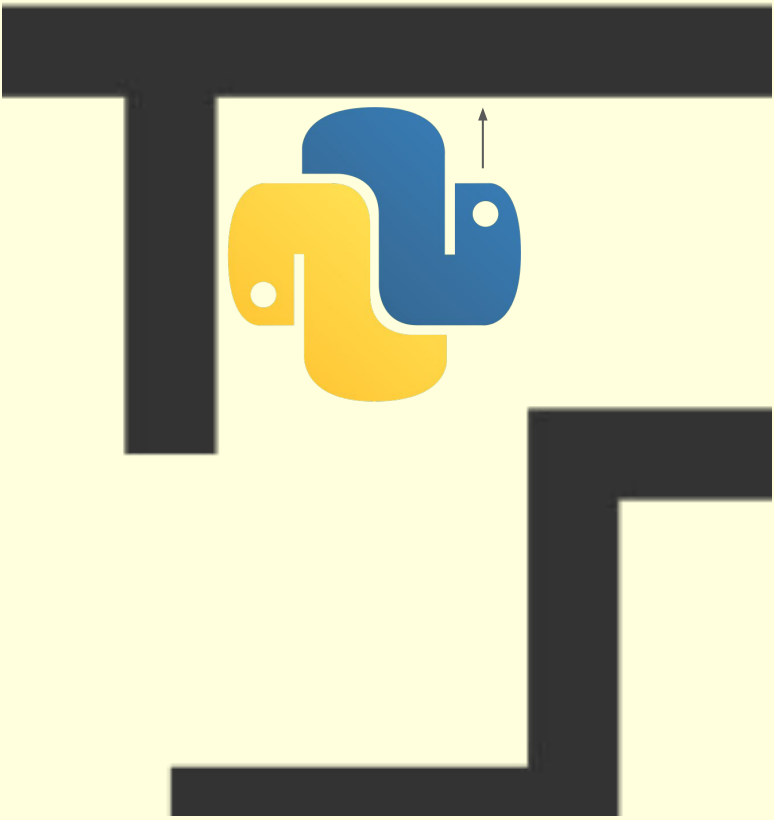
# why do we need to make decisions?

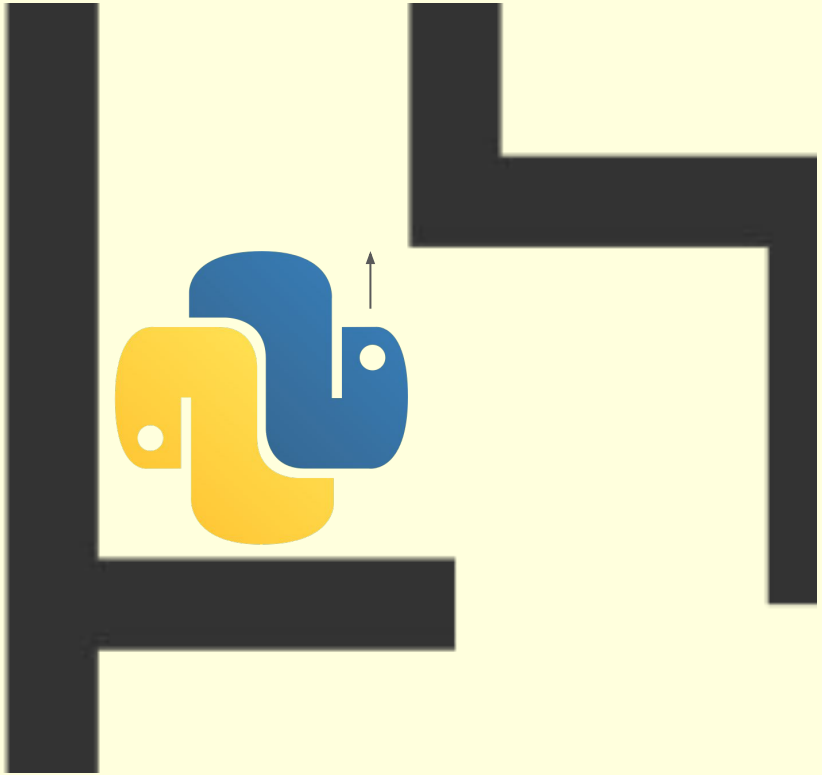
- we don’t want our programs to do the same thing every time
- we want our program to be able to tell the difference between different values and do the right thing

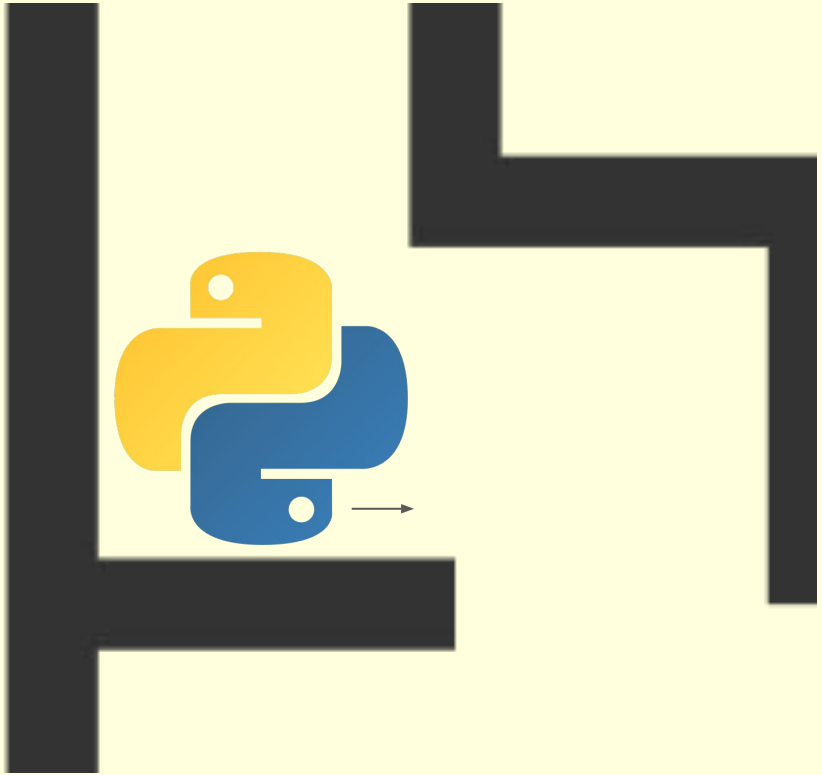
# for example...

- what if we want a robot to find its way through a maze?
- let’s say that the robot has a camera that can see what is in front of it







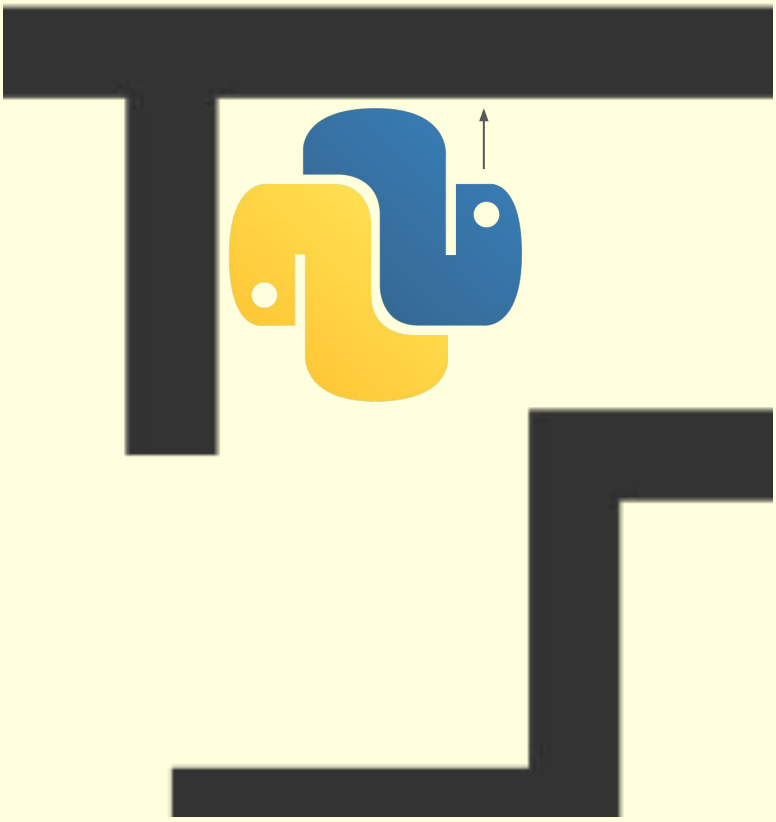


# so how do we make decisions in python?

- we use *if* and *else*
- if something is true, let’s do one thing
- if it is not true, let’s do something else

# think in english, then code

- if you think about it, this makes sense as if we were speaking in english
- let's go back to the maze example



# so how do we write the code?

```
>>> if something is true:
        we do this
    else:
        we do that
```

- notice the indents and the colons

# what if we want there to be more options?

- if and else only give you two paths
- what if we want to have more paths?

# why we would want more than two paths

- for example, what if we want to make a program that can talk?
- you wouldn’t want to talk to someone who could only say two things, right?

# how do we do this?

- we use **elif**
- **elif** is short for ‘else if’
- remember, you can only use **if** once, but you can use **elif** as many times as you want

```
>>> if something is true:
        we do this
    elif something else is true:
        we do this instead
    else:
        we do that
```

# things to remember

- **if** always comes first
- you must have both **if** and **else**
- we need to tell python what code to run in each case
  - all **if**'s and **else**'s must have colons at the end
  - the code that we want to run has to be indented
  - to tell python the code is over, we just remove the indent

# how do we ask python to find if something is true?

- well, we have to compare things
- let’s figure out how to do that

# comparing things

- this is important and tricky! remember this!
- when we are comparing things, we use ==

```
>>> if thing1 == thing2:
        print (“thing1 is equal to thing2”)
```

- why? well, remember how we set variables

# something new and important: booleans

- booleans are the simplest kinds of values in Python
- they can only be True and False
- just as with any values, we can put them in a variable

```
>>> my_boolean = True

>>>my_other_boolean = False
```

# how can we use them?

- booleans are perfect for if statements!
- remember, if statements check if something is true or not to make a decision

```
>>> my_boolean = True

>>> if my_boolean == True:

        print (“It’s true!”)
```

# let try something simple

1. create a boolean that represents whether it is going to rain or not
2. write an **if else** statement
    a. if it is going to rain, then print “Bring an umbrella!”
    b. if it isn’t going to rain, then print “All clear!”

# other comparisons

- we can compare words, also known as strings
- we can compare numbers
- we do not need to learn them all now

# comparing words (strings)

- from now on, we will use the word ‘strings’ to call values that are words and letter surrounded by quotation marks
- to compare strings, all you have to do is use ==

```
>>> if “thing1” == “thing2”:
        print (“equal”)
```

# activity

- let’s make a simple program that can talk
- using an if else statement, write a program where python will give a response to something you say to it
- use **elif**! you want different answers for different responses

- for example, if you said ‘hello python’, you would want python to say hello back
- but if you said ‘bye python’, you would want python to say bye back
- and if you said nothing, then you would want python to say something else

- you can use a variable to talk to python, or you can use **input()**
- have at it!

# comparing numbers

- comparing numbers is easy too
- we can use == to check if they are the same
- or, we can use
  - >, to check if the left value is greater than the right value
  - <, to check if the left value is less than the right value
  - >=, to check if the left value is greater than or equal to the right value
  - <=, to check if the left value is less than or equal to the right value

```
>>> if 1 == 1:
        print (“one equals one!”)

>>> if 2 > 1:
        print (“two is greater than one!”)
```

```
>>> if 1 < 2:
        print (“one is less than two!”)
>>> if 2 >= 2:
        print (“two is greater than or equal to two!”)
>>> if 2 <= 2:
        print (“two is less than or equal to two!”)
```

# time to put it to the test!

- we will be making a small game about guessing numbers!
- copy the provided code into your program
- inside the indented area, you will see an area between comments where you can write your code
- comments in python are created using #

# getting started

- if you look around, you will see some preset variables you can use
- a boolean
  - basically, while the boolean is False, python will keep asking for new numbers and checking them.
  - If python finds that it becomes True, the program will end.

- a number
  - this number is just the number we want to guess
  - you can change it all you want
- another number (inside the indented area)
  - this number is set to be equal to whatever number the player guesses

# remember when coding

- Remember that we have three variables we need to use
  - guessed_right, a boolean
  - target_number, a number
  - inputted_number, a number

- Remember that if you want the game to end, set the boolean equal to True

# using english in if else statements

- there are some english words that we can use in if else statements
- they should make sense if you think about it as english!

- and
  - check if multiple things are true
- or
  - check if one thing is true out of several different things
- not
  - check if something is False

# homework

- finish the talking exercise and the guess numbers exercise
- play around with them!
  - can you change the variables and still make it work?
  - can you shorten it and still make it work?
- look up **random()** in python
  - it generates random numbers
  - figure out how it works and find a use for it in the guess numbers game!