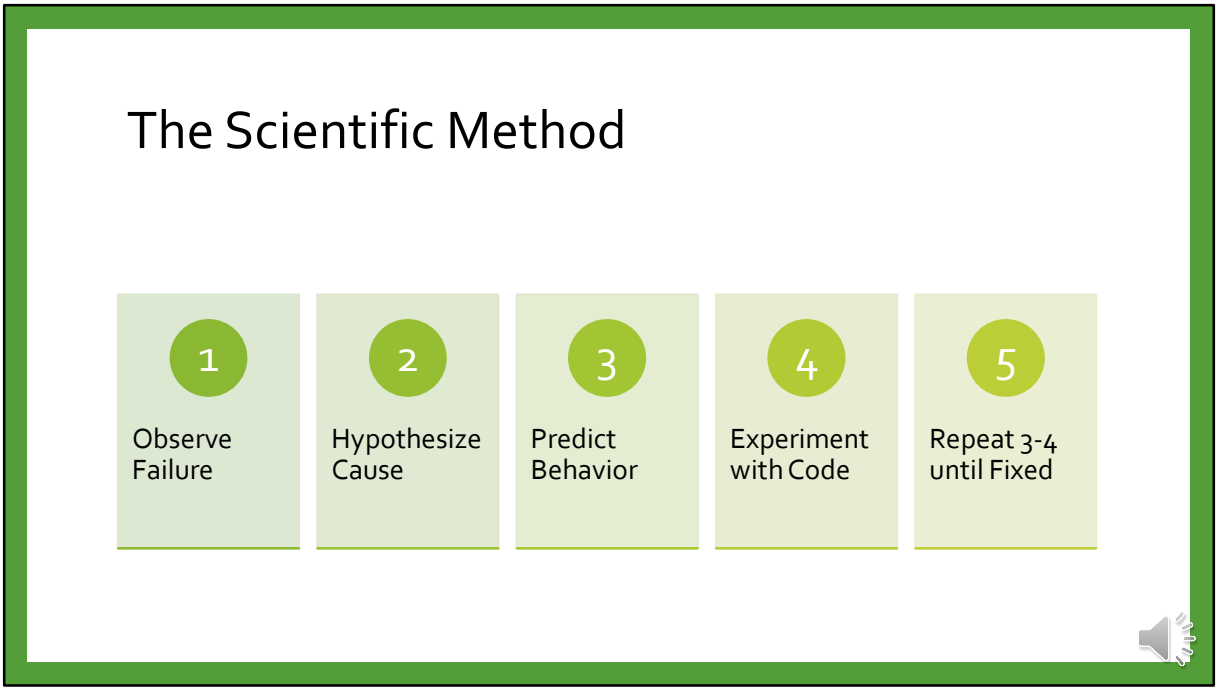
DEBUGGING

An Introduction to Computer Science

Let's learn about Debugging.

What is Debugging?

When a program doesn't work, we say that it has a "bug" in it.
When we get rid of bugs, we call it "Debugging".
The first actual case of debugging a computer program was when Admiral Grace Hopper found a moth trapped inside of a relay of the Mark II Computer.
Nowadays, our bugs usually don't involve live insects.

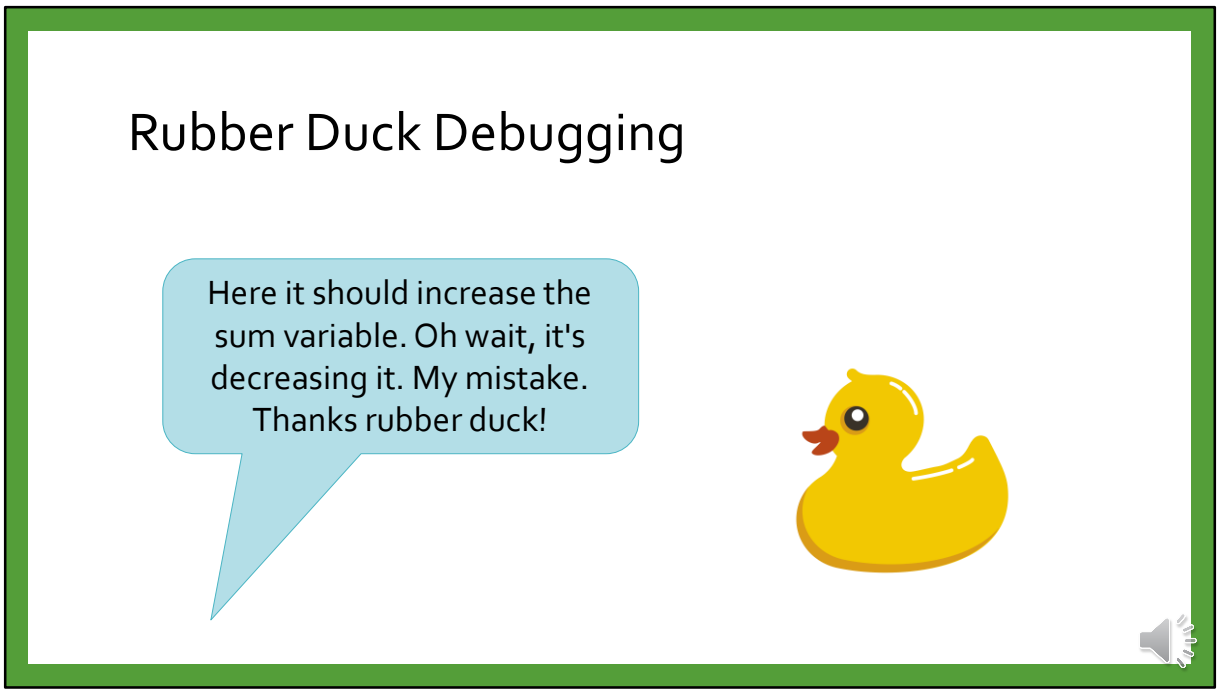


When you're debugging code, you should approach it using the scientific method.
First, we observe a failure in our code.
Second, develop a hypothesis as to why the code failed.
Third, use this hypothesis to make a prediction.
Fourth, test the prediction by experimenting with the code and making observations.
Fifth, repeat steps 3 and 4 until you have fixed the code.
This may be intuitive to some people, but formally following this method can really help train yourself as a programmer!

Wolf Fencing

```
command = input("What are the dimensions")
stop = "stop" in command
- - - - - - - - - - - - - - - - - - - -
print("Works fine until here")
- - - - - - - - - - - - - - - - - - - -
area = length ** width
volume = height * area
set_size(volume)
```

"There's one wolf in Alaska, how do you find it?
First build a fence down the middle of the state, wait for the wolf to howl, determine which side of the fence it is on.
Repeat process on that side only, until you get to the point where you can see the wolf.
In other words; put in a few "print" statements until you find the statement that is failing (then maybe work backwards from the "tracks" to find out where the wolf/bug comes from).
This method is also known as "Print Debugging".



Okay, this will sound weird.
First, get yourself a rubber duck or a wise stuffed animal.
Second, politely inform the duck that you will be explaining your code to it.
Third, explain what your code is supposed to do in general, and the go over each line of code to the rubber duck.
At some point, you will tell the rubber duck what you are supposed to be doing next, and you will realize that's not actually what your code does.
Fourth, thank the duck for its excellent work.
This may sound crazy, but rubber ducks are actually the world's best coders.