DRONE
LOGISTICS SERVICE

DLS in the classroom:
Math Editions

Lesson 2: What's In A Name?

Problem Solving | Numberlines | Variable Coding

Drone Logistics Company Learning

DLS Learning
https//dronelsorg/

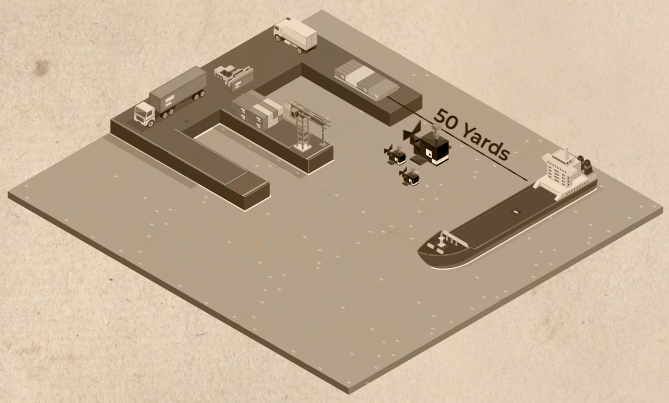
EXTRA

Junior coders save the day by writing code that helped deliver the supply boxes when the S.S Blue Waters could not dock. A whale and her calves popped in for a visit. The ship could not dock. The impressive coders had the Tello delivery drone flying payloads back and forth over the whales.

All the supply boxes were delivered in no time.

All the captains around the world are asking for their own version of the code. The only problem is they don't all have the same amount of supply boxes they need delivered. Each ship has a different amount and different requirements.

What will our star coders do next?!



Instructions:

Adjust your code so that we have variables we can set at the top of our code. Use clear variables names that make your code easy to understand.

Duck, duck, goose:

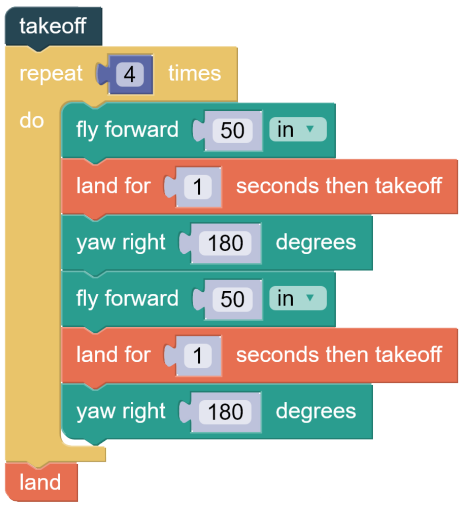
Selecting names for our variables can be tricky. Your code will work no matter what you call your variables, but it will just be harder to read for other people. For example:
dtd = bad distance = good deliveryDistance = better
Let's see what variables we need to create. We can look at our current code to get clues.

Each ship will have a different number of supply boxes.
totalSupplyBoxes (Not every ship will have 20 supply boxes)

Different drones will have a different payload.
dronePayload (Not all drones will a max payload of 4)

Fly forward 50 inches. This is the distance between the ship and the delivery point.
deliveryDistance (It's not aways going to be 50 yards)

The number of times we fly there and back.
totalDeliveries (This should be a formula, not a number)

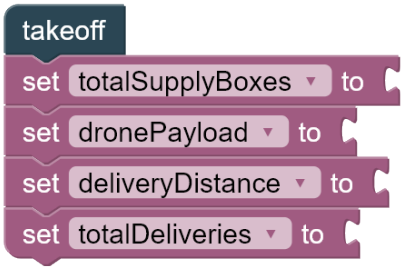


DRONE LOGISTICS SERVICE

Tool Time:

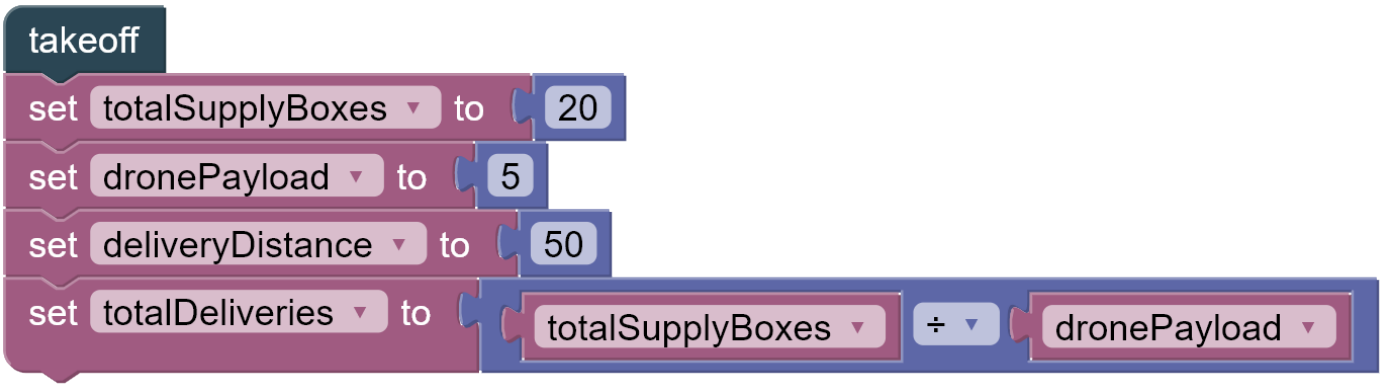
Click on Variables -> Create variable...
Create 4 variables named:
totalSupplyBoxes | dronePayload | deliveryDistance | totalDeliveries

Drag each of the set variables directly under your takeoff block. We are going to be using the math values from our first lesson.



totalSupplyBoxes = 20
dronePayload = 5
deliveryDistance = 50
* totalDeliveries = [totalSupplyBoxes ÷ dronePayload]

The reason we use this formula is because DroneBlocks can help us with the math. The total number of deliveries will always be the total number of supply boxes divided by the drone's max payload. **This is our first math formuia in our code*



Look how fancy that looks. Now let's replace the static numbers in our code with our newly created variables.

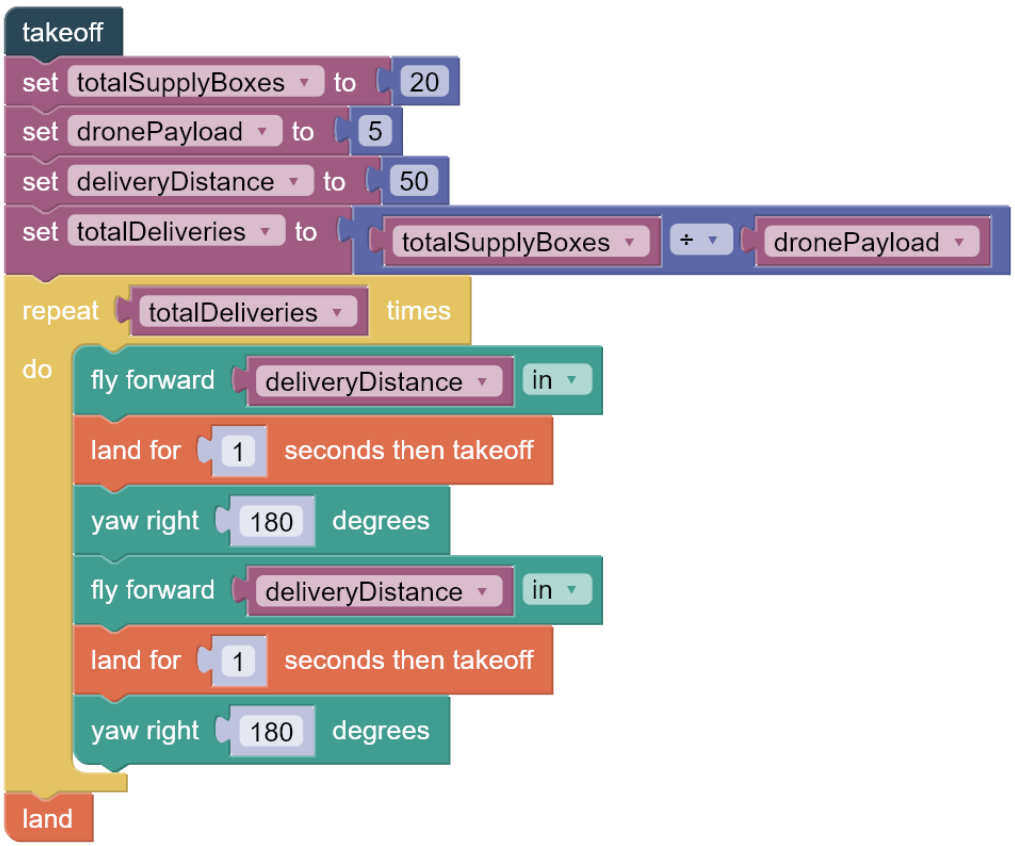
This looks amazing.

Now any time a captain of a ship wants to program the drone all they need to do is enter in the following:

totalSupplyBoxes
dronePayload
deliveryDistance

Amazing. Your code has become really user friendly.

Dont' forget to test it in the simulator and see what happens.



DRONE LOGISTICS SERVICE

It Was Only A Matter Of Time:

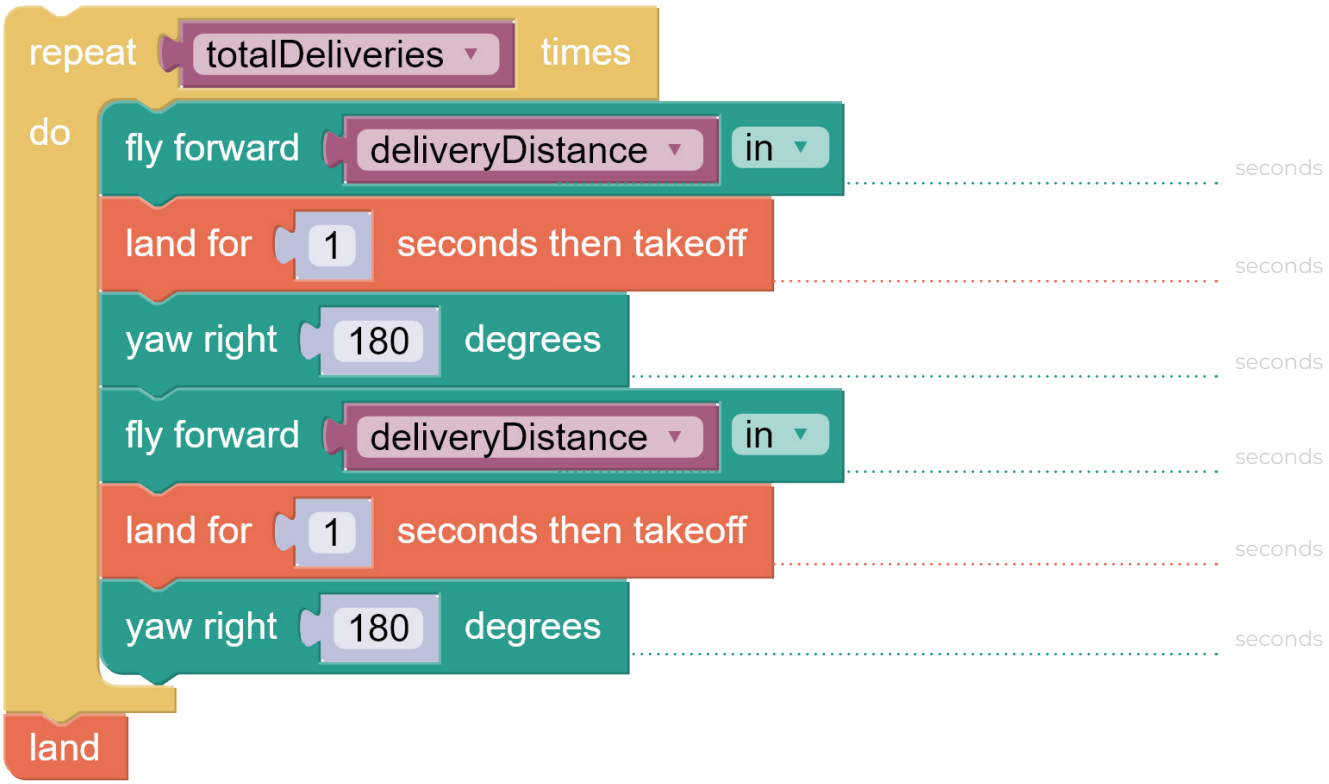
Your code is a great success! Ships all over the world are using your code... wait that's this? An incoming chat from one of your users? Let's hear them out.

"Hello Coder! Your code has made our shipping job so much easier. Thank you so much! Sometimes we have to get our supply boxes delivered quickly. Is there any way you could speed up the current delivery process?

Thank you so much!"
disconnect sound

This user has a good point. Can we make the process quicker? Let's get some more information before we start coding. We are going to call up our current code and see how long each block takes.

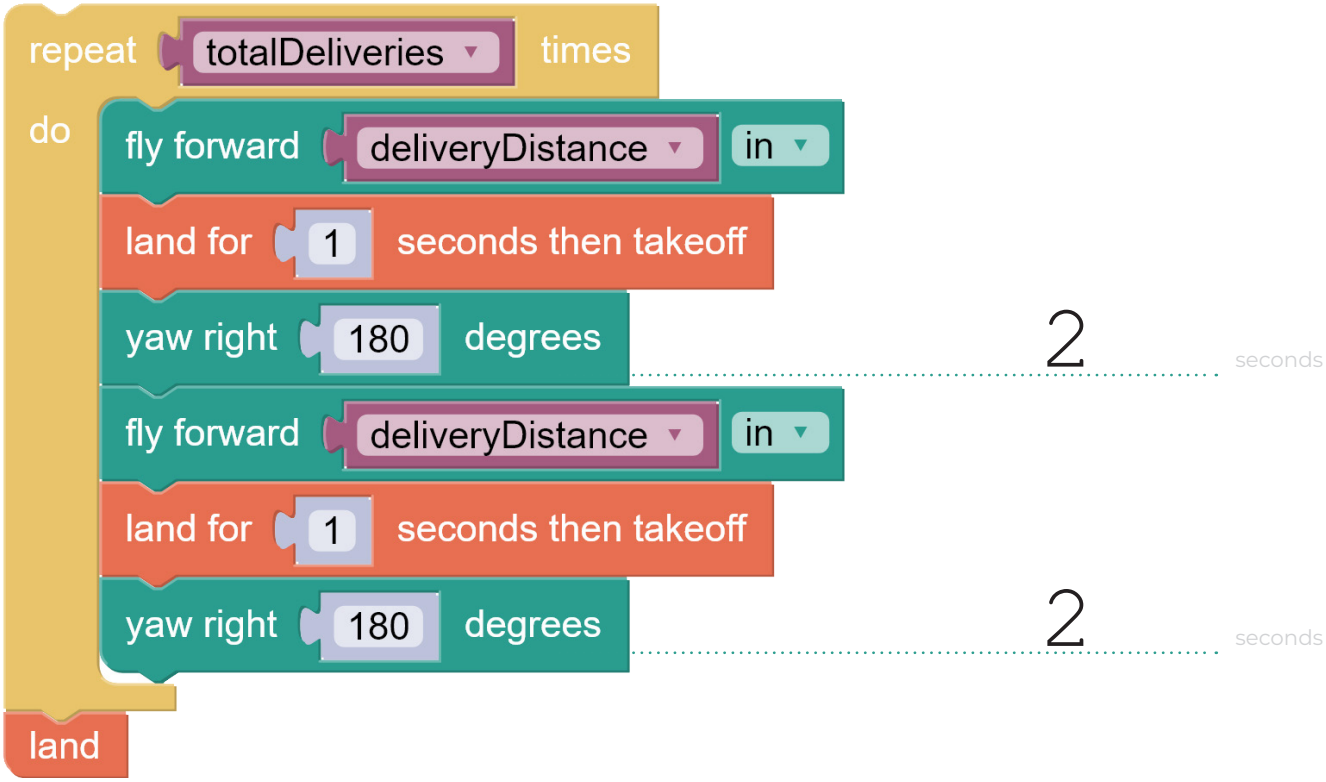
Launch your code in the simulator and time how long each of the following takes to run:



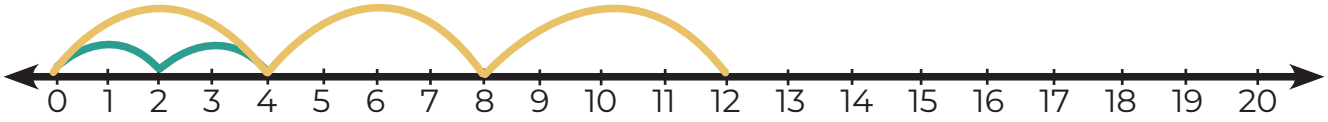
Looking at the time it takes each step let's see what steps we can improve.
Fly forward: We need this because otherwise the drone can't move forward.
land for seconds: This is the "drop off payload" action. This is essential code.
Yaw right: Wait a minute. Drones can fly in any directions. Maybe we can change this?
Fly forward: We need this because otherwise the drone can't move forward.
land for seconds: This is also the "collect payload" action. This is essential code.
Yaw right: Wait a minute. Drones can fly in any directions. Maybe we can change this?

Looking At The Numbers:

Drones do not need to turn around necessarily unless you want to film, see something from the camera or use object avoidance detection. This means we can remove the yaw right part of our code and replace it with fly back deliveryDistance block. Our new code will have the drone flying back and forth without turning around. Amazing. How many seconds will that save us per delivery?



Wow, not having to yaw right is going to save us **4 seconds** per delivery. **4 seconds** might not seem like a lot, but it all adds up over time. Look at this timeline below:



This a great way to introduce multiplication on the number. Right here we have two hops of 2 (for each 2 second yaw right). We also have 3 hops of 4 (for each delivery). We can use a number line to count have many seconds are saved after each delivery. In the above example we have saved 12 seconds after 3 deliveries.

Math Problems:

How much time will be saved after the following deliveries?

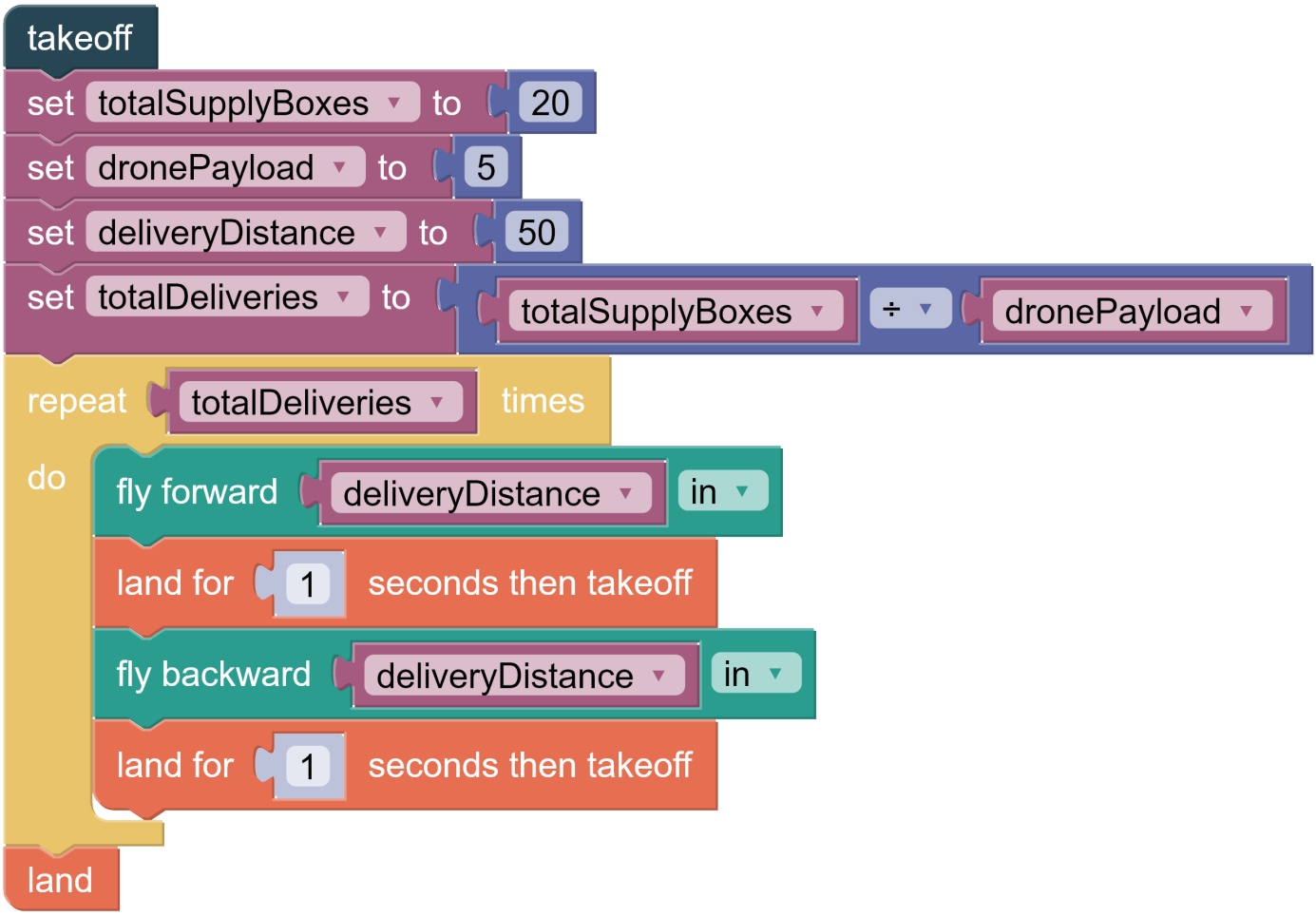
2 deliveries	(2 x 4 = __)	4 deliveries	(4 x 4 = __)
5 deliveries	(5 x 4 = __)	9 deliveries	(9 x 4 = __)

How many delivieries need to be made to save the following amount of seconds:

8 seconds	(8 ÷ 4 = __)	12 seconds	(12 ÷ 4 = __)
24 seconds	(24 ÷ 4 = __)	48 seconds	(48 ÷ 4 = __)

Updating The Code:

It is time update our code. We are going to remove all “yaw right” blocks and replace them with fly backward blocks.



Remember earlier when I said variables were awesome? Well, they are. Here is another reason why. The drone needs to fly backward the same distance as it flies forward each time. Without variables we would have to put in 50 inches twice. Should we need to adjust our code, ie: another delivery distance is 30 inches, we would need to change 50 inches to 30 inches in both places. With variables, we just adjust the variable once.

Just a like magic.

Now that we have varialbes it is much easier for you to create some example equations for your class. It is much easier to update the totalSupplyBoxes and dronePayload to create some math problems for your students to solve.

Bonus Questions:
Here are some slightly trickier questions that would require a bit of knowledge of time calculations as well as division.

How many delivieries need to be made to save the following amount of seconds:

1/2 a minute	(?? ÷ 4 = __)	1 minute	(?? ÷ 4 = __)
5 minutes	(?? ÷ 4 = __)	10 minutes	(?? ÷ 4 = __)

Amazing work on completing this lesson! In the next lesson we are going to be using the Yards to do some math with 3 digit numbers.