

Variables and nested loops



$\epsilon\pi$

ergopedia™

Essential

Physics

Objectives

- Specify variables
- Increment a variable in a loop
- Create and execute programs using nested loops



Assessment

1. Write a program that creates a *variable* for the drive distance of the ErgoBot, and causes it to take on values from 0.2 m to 2.0 m, in 0.2 meter increments.

Use as few commands as possible.



Assessment

2. The program below changes the value of a variable called “distance”. Record the values that “distance” takes on—from start to end of the program.

```
start      // distance keeps changing
variable distance = 0
variable add = 2
do
  do
    distance = distance + add
  loop 4
  distance = 0    // reset the value of “distance”
  add = add + 1   // increase the increment
loop 3
stop
```



New terms

- variable
- increment
- nested loop
- conditional statement

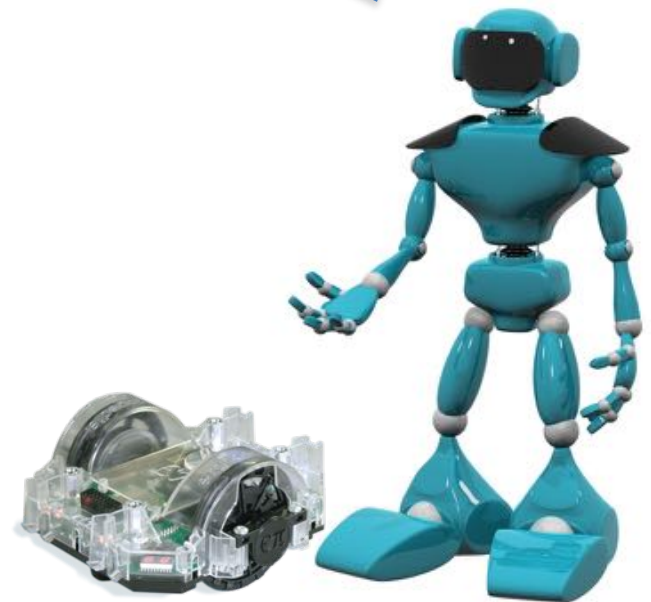


Going further

You are ready to master more fundamentals of programming, including:

- specifying variables
- incrementing variables
- using nested do loops

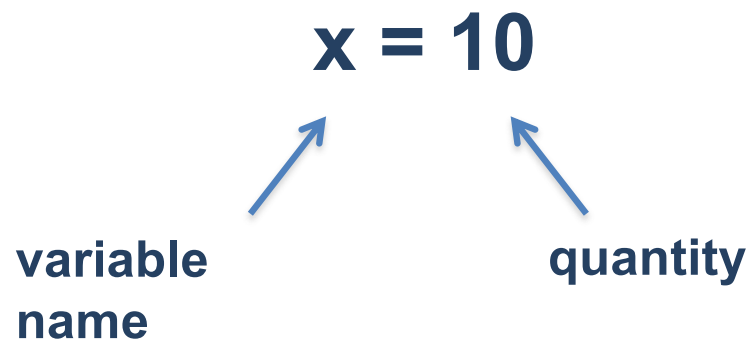
These skills let you program more complex behaviors for your ErgoBot.



Variables

A variable is a symbol used to represent a quantity that can change in value.

For example:



Variables are used in algebra AND in programming.



Specifying a variable

Variable names should start with a letter.
Here is how you create variables in an ErgoBot program:

variable x = 10

variable y = 1

variable dist = 0.5

variable zelda100 = 50

Try to use names that describe the variable, like "dist" for distance.



Variables as arguments

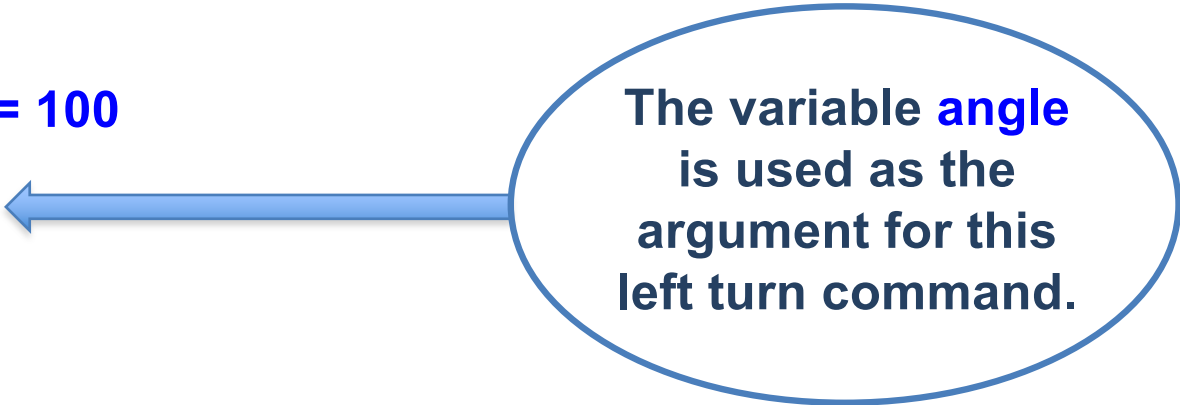
Variables can be used as arguments in commands.

start

variable angle = 100

left (angle)

stop



The variable **angle** is used as the argument for this left turn command.

A blue oval contains the text 'The variable **angle** is used as the argument for this left turn command.' A blue arrow points from the right side of this oval to the word '(angle)' in the 'left (angle)' command line.

Changing a variable

Variables can change their value as a program runs.
Here is an example. What does this program do?

start

variable angle = 100

left (angle)

angle = angle + 50

right (angle)

stop



Changing a variable

Variables can change their value as a program runs.
Here is an example. What does this program do?

start

variable angle = 100

left (angle)

angle = angle + 50

right (angle)

stop

turn left 100°

increase angle to 150°

turn right 150°

Incrementing a variable

Changing a variable inside a loop is called incrementing the variable. What does this program do?

start

variable angle = 90

do

left (angle)

right (angle)

angle = angle + 90

loop 4

stop



Incrementing a variable

Changing a variable inside a loop is called incrementing the variable. What does this program do?

start

variable angle = 90

do

left (angle)

right (angle)

angle = angle + 90

loop 4

stop

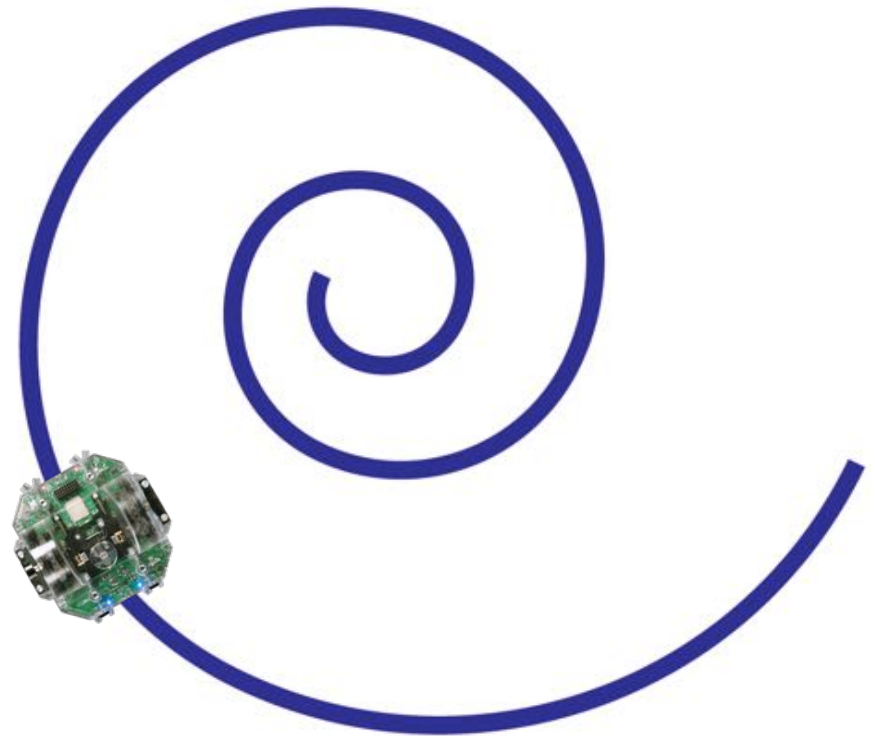
The ErgoBot turns left and right — first with 90° turns, then 180° , 270° , and 360° .

Investigation

Part 1: Incrementing variables

1. Open your programming window and set the ErgoBot to drive mode.
2. Program your ErgoBot to move in a spiral path, starting at the center of the spiral.

Longer programs take longer to download. Be patient.

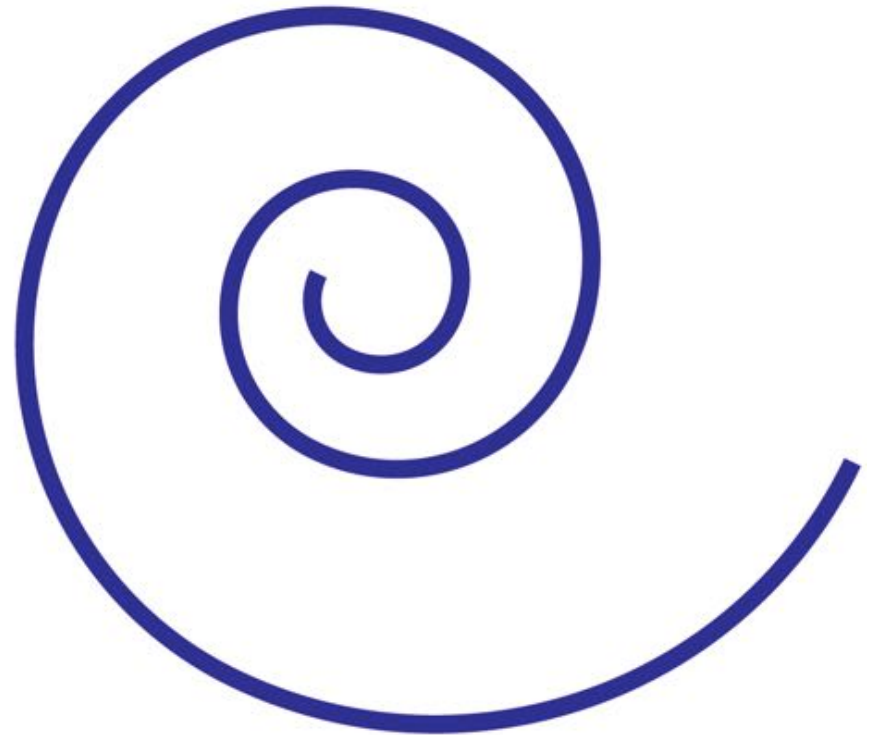


Investigation

Part 1: Incrementing variables

3. Your program must use at least one variable as an argument for a navigational command.
4. Your program must increment the variable inside a loop.
5. Demonstrate your completed program for your teacher.

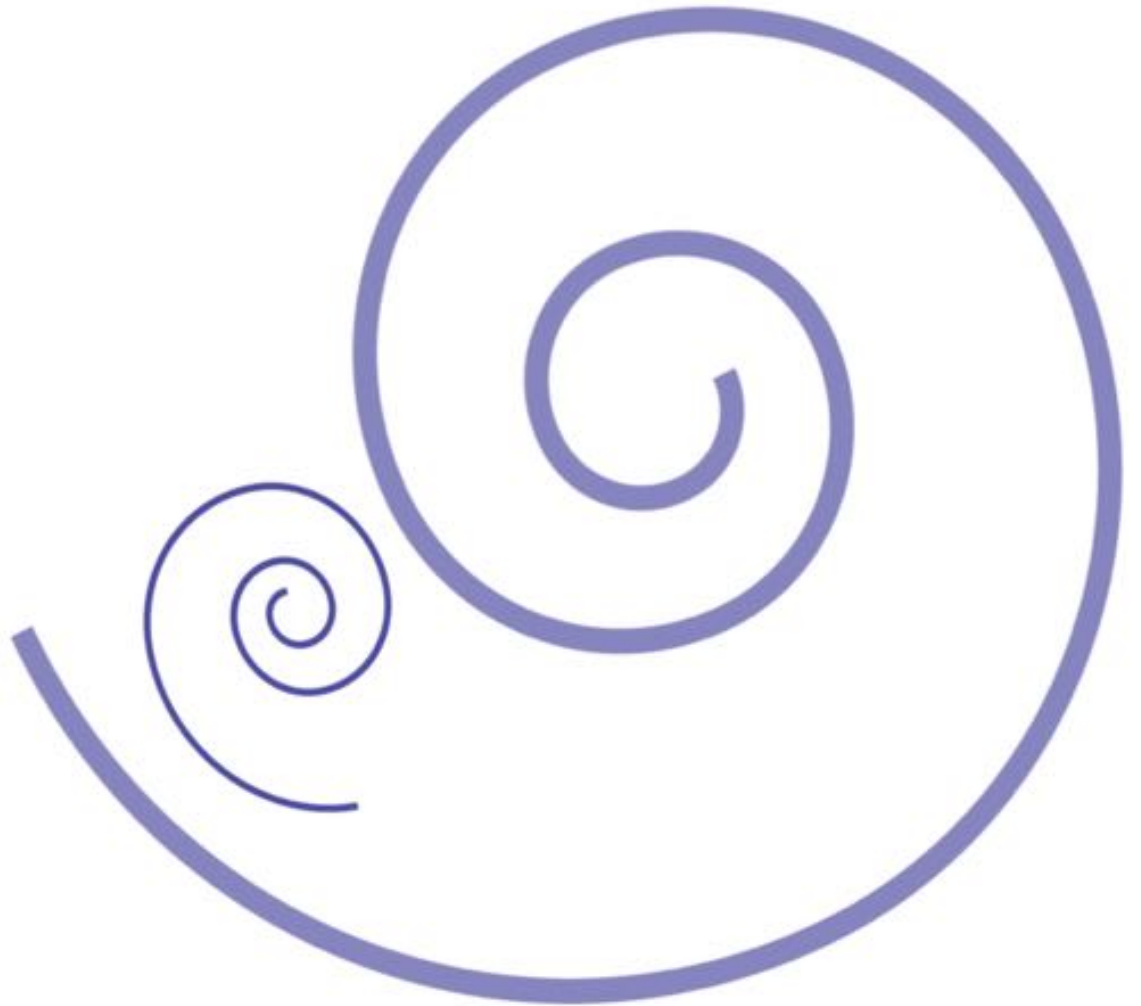
Do include comments!!



Investigation

Questions for Part 1

- a. How would you edit your program to change the direction of your spiral?
- b. How could you change the tightness of the spiral?
- c. Describe two ways to edit your program to lengthen the spiraling path.



Nested loops

Programmers often “nest” one loop inside another.

```
do
  // commands here
  do
    // commands here
    loop 5
  // commands here
loop 10
```

Commands inside the inner loop in this example will repeat 50 times!

Do you see why?

Nested loops

Programmers often “nest” one loop inside another.

```
do
  // commands here
  do
    // commands here
    loop 5
  // commands here
loop 10
```

Nesting loops allows you to program more complex behaviors.

Indenting the loops will help you see them clearly.

Example program

Discuss: What does this code block cause the ErgoBot to do?

```
variable ang = 30  
do  
  do  
    right (ang)  
    ang = ang + 10  
  loop 4  
ang = 30  
loop 2
```



Nested loops

Discuss: What does this program cause the ErgoBot to do?

```
variable ang = 30
do
  do
    right (ang)
    ang = ang + 10
  loop 4
ang = 30
loop 2
```

The ErgoBot turns right 30° ,
then 40° , 50° , and 60°
(totaling 180°).

Then it repeats these turns a
second time. It ends facing
where it started.

Test your knowledge

Why do you need the command circled in red?

```
variable ang = 30
```

```
do
```

```
do
```

```
right (ang)
```

```
ang = ang + 10
```

```
loop 4
```

```
ang = 30
```

```
loop 2
```

What happens if
you leave it out?

Test your knowledge

This command resets the angle variable to its initial value.

```
variable ang = 30
do
  do
    right (ang)
    ang = ang + 10
  loop 4
  ang = 30
loop 2
```

Including the command causes
produces these right turns:

30° , 40° , 50° , 60°
30° , 40° , 50° , 60°

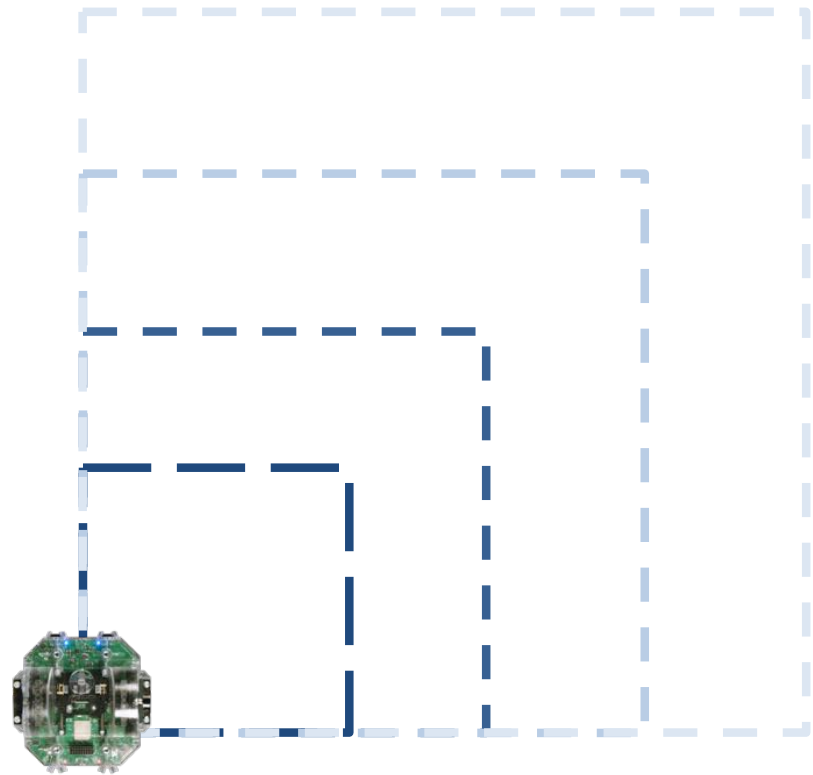
Leaving it out gives:

30° , 40° , 50° , 60°
70° , 80° , 90° , 100°

Investigation

Part 2: Nested loops

1. Write a program to drive the ErgoBot in squares that keep increasing in size.
2. Use nested loops. Use as few commands as possible. DO include comments.
3. Demonstrate your completed program for your teacher.



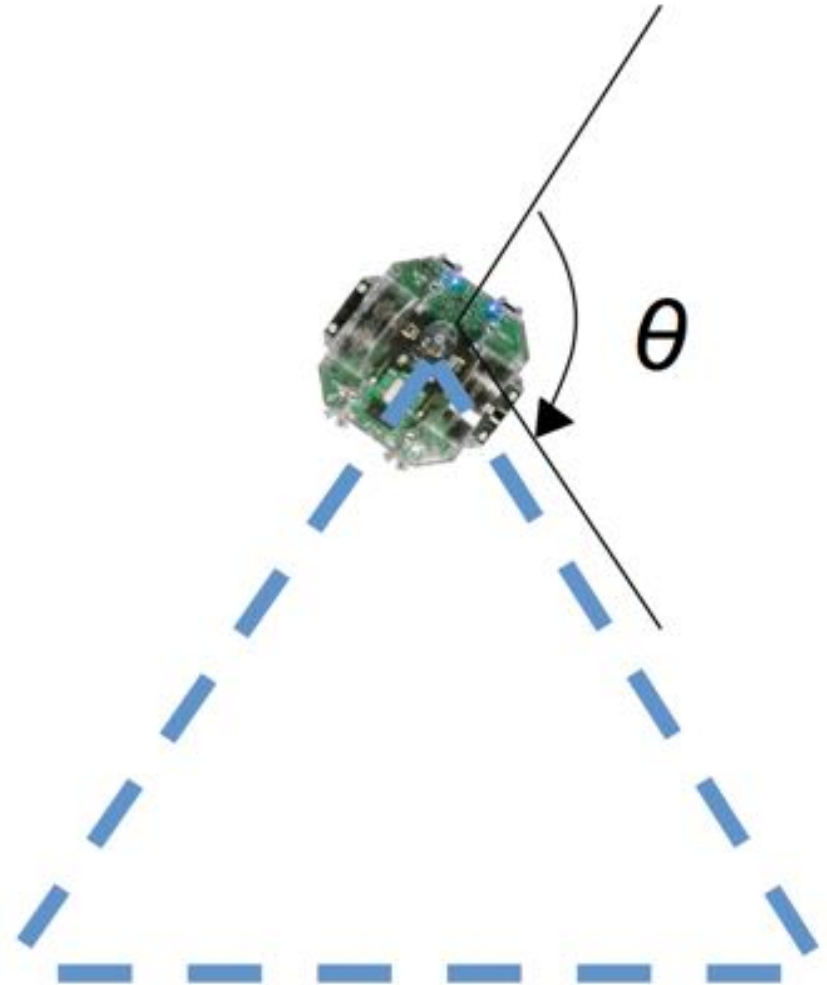
Investigation

Questions for Part 2

- How can you edit your program so that the ErgoBot drives in *triangles* of increasing size?

What turn angle is needed to create an equilateral triangle?
Does the angle change as the triangle grows?

When you are programming,
you are also doing math!



Investigation

Questions for Part 2

- b. How can you edit your program so that the ErgoBot drives in rectangles that get longer but stay the same width?



Mathematics and programming

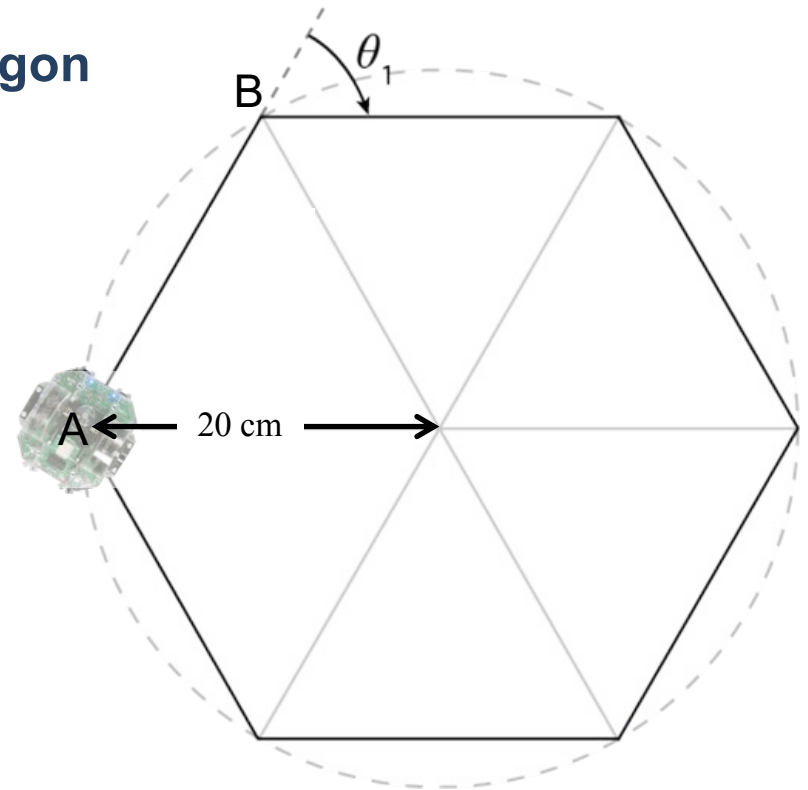
How can we make the ErgoBot trace a hexagon inscribed in a circle of radius 20 cm?

Let's look at the geometry of the motion.

ErgoBot starts at position A.

How far does it have to move to get to B?

What turn angle θ_1 must it use at B?



Mathematics and programming

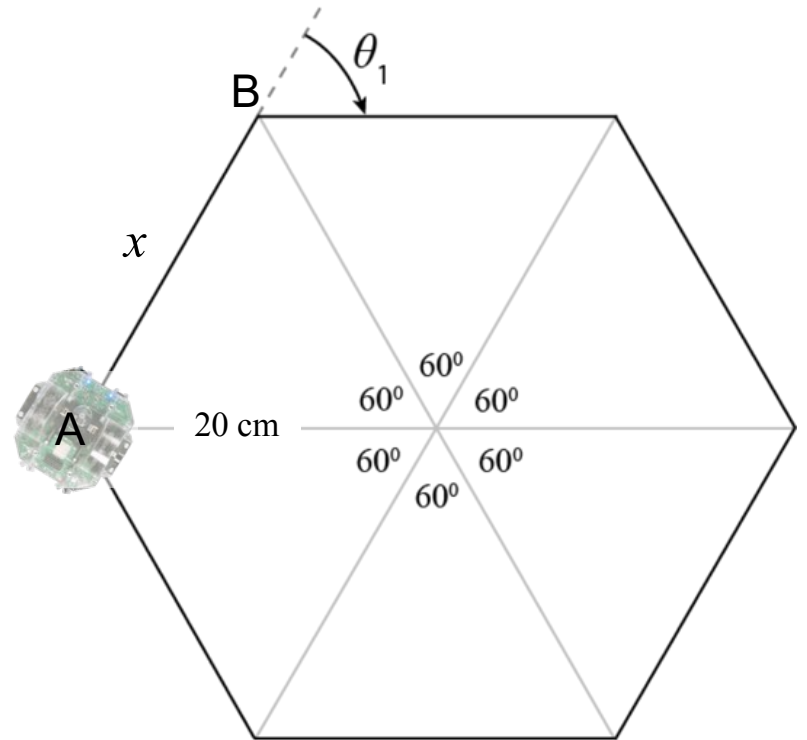
How far does it have to move to get to B?

**These 6 triangles are equilateral.
Do you see why?**

The distance x must equal the radius of the circle: 20 cm.

The first command is:

```
drive (0.2) // drive for 20 cm
```



Mathematics and programming

What turn angle θ_1 must it use at B?

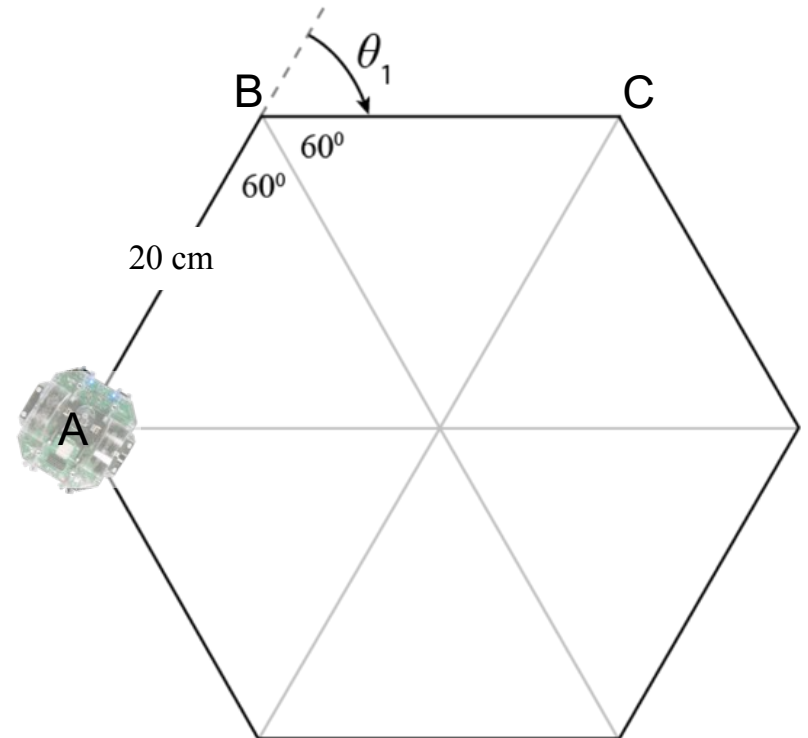
The turn angle must equal 60° .

$$\theta_1 = 180^\circ - 60^\circ - 60^\circ = 60^\circ$$

The second command is:

right (60) // turn 60° clockwise

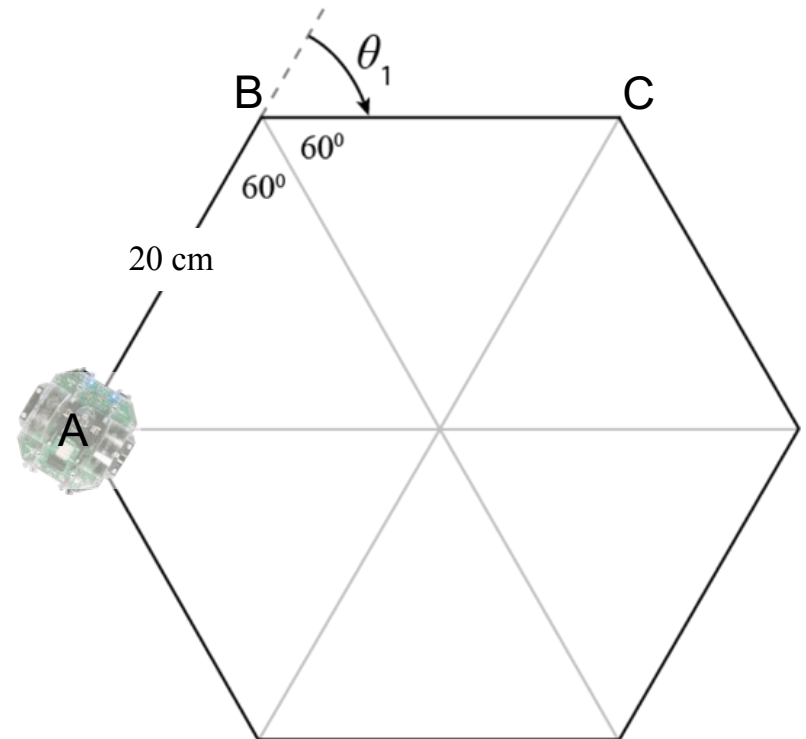
The drive and turn commands must repeat 6 times to make a hexagon.



Mathematics and programming

And the complete program is:

```
// Drive the ErgoBot in a hexagon
start
variable distance = 0.2 // set the drive distance
variable angle = 60 // set the turn angle
do // start a loop
    drive (distance)
    turn (angle)
loop 6 // repeat 6 times
stop
```



Sensors

Controlling a robot using pre-programmed maneuvers is not the most accurate method.

Many robots contain sensors that allow them to sense their environment and adjust their behaviors.



Types of sensors

Robots use many different kinds of sensors, including:

- touch sensors
- cliff sensors
- temperature sensors
- sound sensors
- light sensors (IR and color)
- GPS



Sensors

The ErgoBot has sensors in its wheels that detect their rotation.

This cleaning robot uses touch sensors to detect walls and other obstacles.



Decision making

Conditional statements allow a robot to make decisions based on sensor information.

IF a given condition is true
THEN take a certain action.



Conditional statements

For example:

IF you sense a barrier
THEN turn around.



Robotics is used in many fields



Advanced sensors and clever programming allow robots to perform a wide variety of dangerous, difficult, or repetitive tasks in fields such as medicine, manufacturing, defense, and agriculture.



Robotics as a career

Knowledge of robotics and engineering can open the doors to career opportunities in an exciting and growing field.

If you are interested, look for opportunities to join a robotics team or take more classes in programming.



Assessment

1. Write a program that creates a *variable* for the drive distance of the ErgoBot, and causes it to take on values from 0.2 m to 2.0 m, in 0.2 meter increments.

Use as few commands as possible.



Assessment

1. Write a program that creates a *variable* for the drive distance of the ErgoBot, and causes it to take on values from 0.2 m to 2.0 m, in 0.2 meter increments.

Use as few commands as possible.

```
start  
variable dist = 0.2  
do  
  drive (dist)  
  dist = dist + 0.2  
loop 10  
stop
```



Assessment

2. The program below changes the value of a variable called “distance”. Record the values that “distance” takes on—from start to end of the program.

```
start      // distance keeps changing
variable distance = 0
variable add = 2
do
  do
    distance = distance + add
  loop 4
  distance = 0    // reset the value of “distance”
  add = add + 1   // increase the increment
loop 3
stop
```



Assessment

2. The program below changes the value of a variable called “distance”. Record the values that “distance” takes on—from start to end of the program.

```
start      // distance keeps changing
variable distance = 0
variable add = 2
do
  do
    distance = distance + add
  loop 4
  distance = 0    // reset the value of “distance”
  add = add + 1   // increase the increment
loop 3
stop
```

Answer:

0, 2, 4, 6, 8

0, 3, 6, 9, 12

0, 4, 8, 12, 16



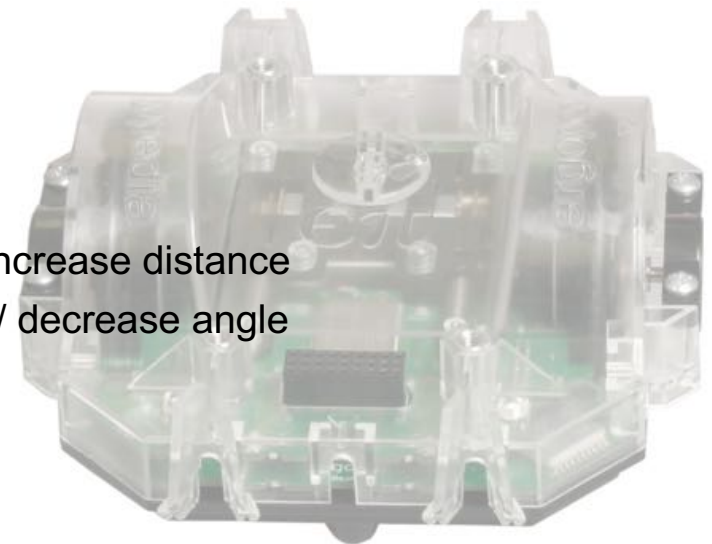
Example program: *Incrementing variables*

Program your ErgoBot to move in a spiral path, starting at the center of the spiral. Your program must increment a variable inside a loop.

Three ways to accomplish the goal:

- increase distance between turns
- decrease turn angle
- increase distance and decrease angle

```
// Investigation Part 1, Incrementing variables
start
variable angle = 120
variable dist = 0.10
do
  drive (dist)
  left (angle)
  dist = dist + 0.10 // increase distance
  angle = angle - 12 // decrease angle
loop 10
stop
```



Example program: *Nested loops*

Write a program to drive the ErgoBot in squares that keep increasing in size. Use nested loops.

Possible solution:

```
// Investigation Part 2, Nested loops
start
variable length = 0.20
do
  do           // drive in a square
    drive (length)
    left (90)
  loop 4
  length = length + 0.20 // increase sides
loop 4
stop
```

