Procedures

Procedures

A procedure is an algorithm, which is assigned to a name that is created by the user.

Example: To create an algorithm for a square of size 100.

Type:

```
> to square (or whatever name you wish to give to the shape)
```

Usually this brings up a separate dialogue box or a change in punctuation at the beginning of the line, where the algorithm is typed.

```
repeat 4[fd 100 rt 90]
```

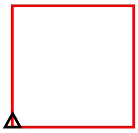
The procedure must be finished with the word:

```
end
```

In some cases (such as the online versions Turtle Academy or Logo Interpreter) the full algorithm is typed in one line.

```
> to square repeat 4[fd 100 rt 90] end
> square
```

This algorithm will draw a square like this.



Procedures with Variables

If you want a square of any size, where you tell Logo what size you want, you use a variable.

The variable has a colon in front and a name of your choosing.

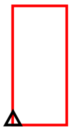
```
> to square :length
repeat 4[fd :length rt 90]
end
> square 100
```

Notice that the variable with colon in front is placed after the procedure name and in the algorithm for the procedure in place of the number.

You can use a number of variables:

```
> to rectangle :length :width
repeat 2[fd :length rt 90 fd :width rt 90]
end
> rectangle 150 60
```

This algorithm will draw a rectangle like this.

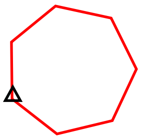


Procedures

Remember you can use calculations, so any sized regular polygon could be:

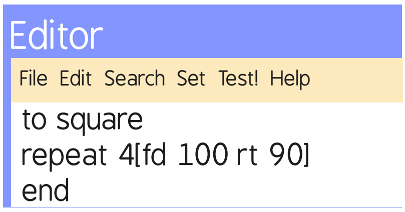
```
> to polygon :length :sides
repeat :sides[fd :length rt 360/:sides]
end
> polygon 40 7
```

This algorithm will draw a polygon like this.



Editing Procedures

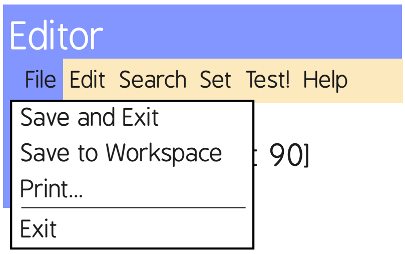
In MSWLogo and other Logo applications the edit command brings up an editor dialogue box allowing procedures to be edited. The procedure sometimes needs an inverted comma in front.



```
to square
repeat 4[fd 100 rt 90]
end
```

```
> edit "square
```

Edit the procedure as necessary, then save and exit.



In online versions like Turtle Academy and Logo Interpreter, which use single line procedure creation there is no edit. Procedures can be retyped which will override a previous algorithm of the same name. If the procedure was typed recently, use the up arrow to scroll through to the previous version, which can then be edited using the left and right arrow.

Type:

```
> to square repeat 4[fd 100 rt 90] end
> square
```

A square is drawn.

Press the up arrow twice and the algorithm will reappear in the command line. Use the left arrow to scroll through to the variable 100 and change to 150. Press enter.

```
> square
```

A second square of size 150 is drawn.

Programming Turtle Logo: Procedures

Please Note: There are 2 slightly different versions of the activity sheets used in this lesson; one for online programs such as Turtle Logo / Logo Interpreter and the other for MSWLogo.

Aim:

Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts.

Use sequence, selection, and repetition in programs; work with variables and various forms of input and output.

Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs in the context of using Turtle Logo to create and debug a procedure.

I can create and debug an algorithm to create a procedure.

Success Criteria:

I can write commands in the correct order.

I can correct any mistakes.

I can write a procedure.

I can use the commands fd, bk, rt, lt, cs, penup, pendown and repeat.

Key/New Words:

Algorithm, instructions, commands, forward (fd), left (lt), right (rt), move, turn, clear screen (cs), variable, penup, pendown, calculation, procedure.

Resources:

Lesson Pack

Desktop computer / laptop

Turtle Logo application: installed / online

Whiteboards / pens or books / pens for recording.

Preparation:

Procedures Activity Sheet - as required

Prior Learning: It will be helpful if children can use and understand the commands forward (fd), backward (bk), right (rt), left (lt) and repeat alongside a variable.

Learning Sequence

Some Regular Polygons: Children write a series of algorithms for regular polygons, remembering the commands used in the previous unit. (*Forward (fd), backward (bk), right (rt), left (lt) and repeat alongside a variable.*)

Write a Procedure / Edit a Procedure: Demonstrate how to write a procedure for a square. (*Most Turtle Logo applications use a similar method, but it does vary from version to version.*)

Write Your Own Procedures: Children copy the procedure for a square and then try other regular polygons. They can then try to write a procedure for an irregular shape (a rectangle).Children use the differentiated **Procedures Activity Sheets.**

★	Children use support to create the procedures.	★★	Children are guided through creating the procedures and are supported when using variables in procedures.	★★★	Children create procedures including one for a regular polygon.

Share: Children share their procedures for rectangles and other polygons with a partner, testing and debugging as necessary.

Procedures: Demonstrate how to write procedures for other regular polygons and how to save them.
Turn? Remind the children how to get Turtle Logo to calculate the angle of turn.
Use a Variable: If the application will allow, demonstrate how to add a variable to a procedure.

More Procedures: Children write other procedures and, if the application will allow, save the procedures.

Whole Class

Debugging a Procedure: Children look at the example of a procedure on the slide. Can you explain why it will not work?

Task*it*

***Procedure*it:** Children can make procedures for regular polygons and other shapes, using variables if possible.

***Challenge*it:** Use the **Challenge Cards** for extension activities.

Creating Procedures in Turtle Logo

Try the procedures below, remember to look for the > symbol that indicates where text must be typed and the lines following will be in a dialogue box.

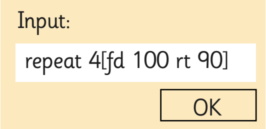
Procedure for a Square

Create a procedure for drawing a square of size 100.

> to square

repeat 4[fd 100 rt 90] <ok>

end <ok>

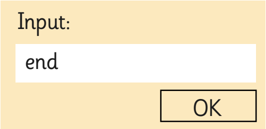


Input:

repeat 4[fd 100 rt 90]

Test the procedure by typing:

> square



Input:

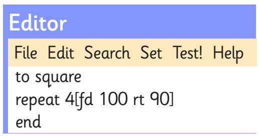
end

Editing the Procedure

If the procedure doesn't draw the square use the editor to edit the procedure.

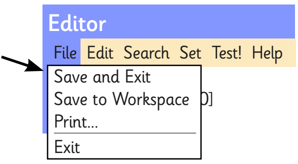
> edit square

Make sure the procedure is correct.



```
to square
repeat 4[fd 100 rt 90]
end
```

'Save and Exit' to save the procedure.



Procedure for Polygons

Create procedures for other polygons, remembering to press ok after each line in the dialogue box.

> to hexagon

repeat 6[fd 100 rt 60]

end

> to octagon

repeat 8[fd 75 rt 45]

end

Test the procedures by typing:

> hexagon

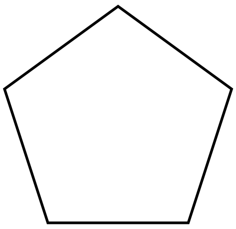
> octagon

Procedure for a Pentagon

Try to create a procedure for a pentagon.

You might want to try to work out the algorithm before you create the procedure.

Save, screenshot or snip your shapes and procedures.



Extension Task:

Procedure for a Rectangle

Create a procedure for drawing a rectangle of size 50 by 150.

> to rectangle

repeat 2[fd 50 rt 90 fd 150] <ok>

end <ok>

Test the procedure by typing:

> rectangle

Creating Procedures in Turtle Logo

Try the procedures below, remember to look for the > symbol that indicates where text must be typed and the lines following will be in a dialogue box.

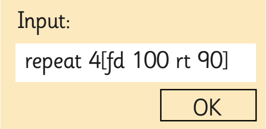
Procedure for a Square

Create a procedure for drawing a square of size 100.

> to square

repeat 4[fd 100 rt 90] <ok>

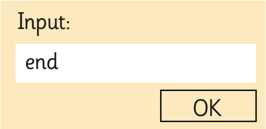
end <ok>



repeat 4[fd 100 rt 90]

Test the procedure by typing:

> square



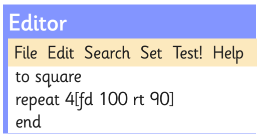
end

Editing the Procedure

If the procedure doesn't draw the square use the editor to edit the procedure.

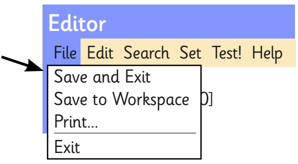
> edit square

Make sure the procedure is correct.



```
to square
repeat 4[fd 100 rt 90]
end
```

'Save and Exit' to save the procedure.



Procedure for Polygons

Create procedures for other polygons, remembering to press ok after each line in the dialogue box.

> to hexagon

repeat 6[put in the algorithm for each side turning 60]

end

> to octagon

repeat 8[put in the algorithm for each side turning 45]

end

Test the procedures by typing:

> hexagon

> octagon

Procedure for a Pentagon

Try to create a procedure for a pentagon.

You might want to try to work out the algorithm before you create the procedure.

Save, screenshot or snip your shapes and procedures.

Procedure for a Rectangle.

Create a procedure a rectangle of size 50 by 150.

> to rectangle

repeat 2[put in the algorithm for 2 sides] <ok>

end <ok>

Test the procedure by typing:

> rectangle

Extension Tasks:

Procedures with Variables

Create a procedure for a square with a variable length of side that you choose each time.

> to square :length repeat

4[fd :length rt 90]

end

Try out the procedure by typing:

> square 100

Try other sizes.

Create a procedure for a rectangle with a variable length and width that you choose each time.

> to rectangle :length :width

repeat 2[fd :width rt 90 fd :length rt 90]

end

Try out the procedure with different size rectangles.

Creating Procedures in Turtle Logo

Try the procedures below, remember to look for the > symbol that indicates where text must be typed and the lines following will be in a dialogue box.

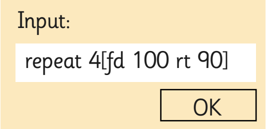
Procedure for a Square

Create a procedure for drawing a square of size 100.

> to square

repeat 4[fd 100 rt 90] <ok>

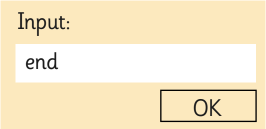
end <ok>



```
repeat 4[fd 100 rt 90]
```

Test the procedure by typing:

> square



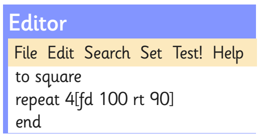
```
end
```

Editing the Procedure

If the procedure doesn't draw the square use the editor to edit the procedure.

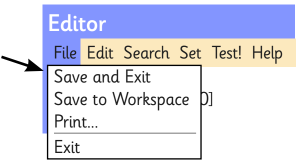
> edit square

Make sure the procedure is correct.



```
to square
repeat 4[fd 100 rt 90]
end
```

'Save and Exit' to save the procedure.



Procedure for Polygons

Create procedures for other polygons, remembering to press ok after each line in the dialogue box.

The turns for a pentagon, hexagon and octagon are 72, 60 and 45.

Test the procedures.

Save, screenshot or snip your shapes and procedures.

Procedure for a Rectangle

Create a procedure for a rectangle of size 50 by 150.

Test the procedure by typing:

> rectangle

Extention Tasks:

Procedures with Variables

Create a procedure for a square with a variable length of side that you choose each time.

> to square :length repeat

4[fd :length rt 90]

end

Try out the procedure by typing:

> square 100

Try other sizes.

Create a procedure for a rectangle with a variable length and width that you choose each time.

> to rectangle :length :width

Complete the algorithm.

Try out the procedure with different size rectangles.

Create a procedure for a polygon of any length of side and any number of sides.

> to polygon : length :sides

Hint: remember the turn for a polygon is 360 divided by the number of sides.

Test your algorithm by drawing a nonagon.

Creating Procedures in Turtle Logo

Try the procedures below, remember to look for the > symbol, this indicates where text must be typed.

Procedure for a Square

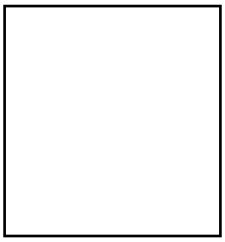
Create a procedure for drawing a square of size 100.

> to square repeat 4[fd 100 rt 90] end

Test the procedure by typing:

> square

Think about what each part of the procedure is doing.



Procedure for Polygons

Create procedures for other polygons.

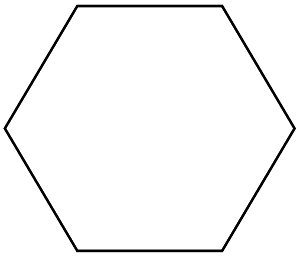
> to hexagon repeat 6[fd 100 ____ 60] end

> to octagon repeat 8[____ 100 rt 45] end

Test the procedures by typing:

> hexagon

> octagon



Extension Tasks:

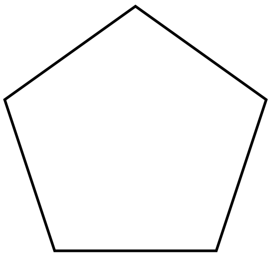
Procedure for a Pentagon

Try to create a procedure for a pentagon.

You might want to try to work out the algorithm before you create the procedure.

Hint: Try 72 for the size of angle.

Save, screenshot or snip your shapes and procedures.



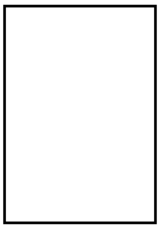
Procedure for a Rectangle

Create a procedure for drawing a rectangle of size 50 by 150.

> to rectangle repeat 2[put in the algorithm for 2 sides] end

Test the procedure by typing:

> rectangle



Creating Procedures in Turtle Logo

Try the procedures below, remember to look for the > symbol, this indicates where text must be typed.

Procedure for a Square

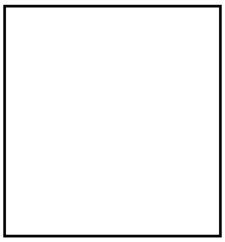
Create a procedure for drawing a square of size 100.

> to square repeat 4[fd 100 rt 90] end

Test the procedure by typing:

> square

Think about what each part of the procedure is doing.



Procedure for Polygons

Create procedures for other polygons.

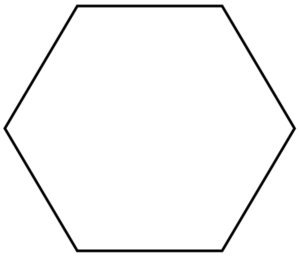
> to hexagon repeat 6[fd 100 ____ 60] end

> to octagon repeat 8[____ 100 rt 45] end

Test the procedures by typing:

> hexagon

> octagon



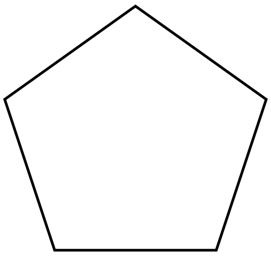
Procedure for a Pentagon

Try to create a procedure for a pentagon.

You might want to try to work out the algorithm before you create the procedure.

Hint: Try 72 for the size of angle.

Save, screenshot or snip your shapes and procedures.



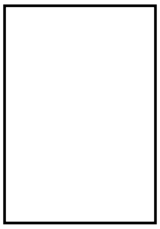
Procedure for a Rectangle

Create a procedure for drawing a rectangle of size 50 by 150.

> to rectangle repeat 2[put in the algorithm for 2 sides] end

Test the procedure by typing:

> rectangle



Extension Tasks:

Create a procedure for a square with a variable length of side that you choose each time.

> to square :length repeat 4[fd :length rt 90] end

Try out the procedure by typing:

> square 100

Try other sizes.

Create a procedure for a rectangle with a variable length and width that you choose each time.

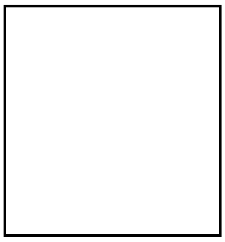
> to rectangle :length :width repeat 2[fd :width rt 90 fd :length rt 90] end

Try out the procedure with different size rectangles.

Creating Procedures in Turtle Logo

Try the procedures below, remember to look for the > symbol, this indicates where text must be typed.

Procedure for a Square



Create a procedure for drawing a square of size 100.

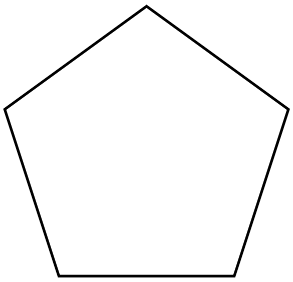
```
> to square repeat 4[fd 100 rt 90] end
```

Test the procedure by typing:

```
> square
```

Think about what each part of the procedure is doing.

Procedure for Polygons



Create procedures for other polygons.

Think about:

- what the procedure needs to be called
- how many times it needs to be repeated
- the size of sides and angles

Hint: The turns for a pentagon, hexagon and octagon are 72, 60 and 45.

Test then, save, screenshot or snip your shapes and procedures.

Procedure for a Rectangle

Create a procedure for drawing a rectangle of size 50 by 150.

How will you add the 2 different sized lengths to the procedure?

Test the procedure by typing:

```
> rectangle
```

Extension Tasks:

Create a procedure for a square with a variable length of side that you choose each time.

```
> to square :length repeat 4[fd :length rt 90] end
```

Try out the procedure by typing:

```
> square 100
```

Try other sizes.

Create a procedure for a rectangle with a variable length and width that you choose each time.

```
> to rectangle :length :width complete the algorithm
```

Try out the procedure with different size rectangles.

Create a procedure for a polygon of any length of side and any number of sides.

```
> to polygon : length :sides
```

Hint: remember the turn for a polygon is 360 divided by the number of sides.

Test your algorithm by drawing a nonagon.

Programming Turtle Logo

Challenge Cards

Programming Turtle Logo

Challenge Cards

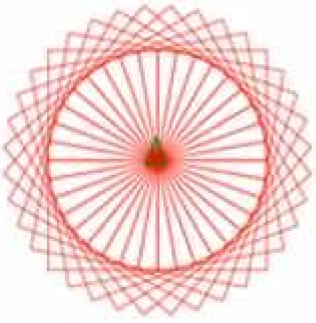
Programming Turtle Logo

Challenge Cards

Programming Turtle Logo

Challenge Cards

Create a procedure for a square of side 100. Use the procedure to write an algorithm to repeat and rotate the square.

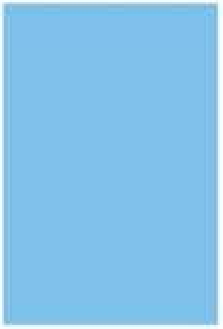


Create a procedure for a square with a variable length side, using :length as the variable.

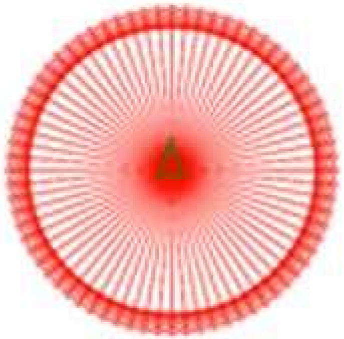
Start
To square :length

Create a procedure for a rectangle with variable length and width using :length and :width.

Start
To rectangle :length :width



Create a procedure for an equilateral triangle of side 120. Use the procedure to write an algorithm to repeat and rotate the triangle.



Create a procedure for a square of side 100. Use the procedure to write an algorithm to repeat and rotate the square.

Answer:

```
Top left
To square
repeat 4[fd 100 rt 90]
end
```

Create a procedure for a square with a variable length side, using :length as the variable.

Answer:

```
To square :length
repeat 4[fd :length rt 90]
end
```

Create a procedure for a rectangle with variable length and width using :length and :width.

Answer:

```
To rectangle :length :width
repeat 2[fd :length rt 90 fd :width rt 90]
end
```

Create a procedure for an equilateral triangle of side 120. Use the procedure to write an algorithm to repeat and rotate the triangle.

Answer:

```
to triangle
repeat 3[fd 120 rt 120]
end
```

Programming Turtle Logo

Challenge Cards

Programming Turtle Logo

Challenge Cards

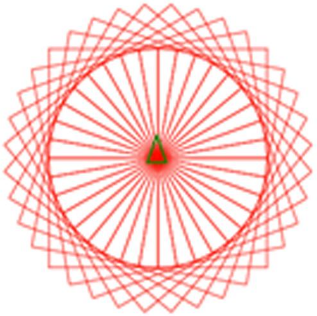
Programming Turtle Logo

Challenge Cards

Programming Turtle Logo

Challenge Cards

Create a procedure for a square of side 100. Use the procedure to write an algorithm to repeat and rotate the square.

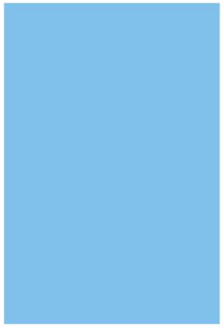


Create a procedure for a square with a variable length side, using :length as the variable.

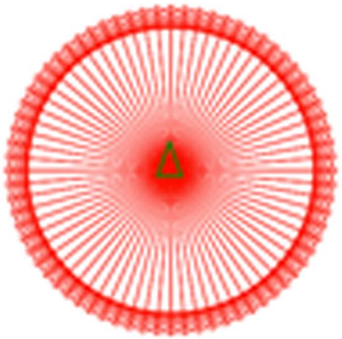
Start
To square :length

Create a procedure for a rectangle with variable length and width using :length and :width

Start
To rectangle :length :width



Create a procedure for an equilateral triangle of side 120. Use the procedure to write an algorithm to repeat and rotate the triangle.



Create a procedure for a square of side 100. Use the procedure to write an algorithm to repeat and rotate the square.

Answer:

```
Top left
To square
repeat 4[fd 100 rt 90]
end
```

Create a procedure for a square with a variable length side, using :length as the variable.

Answer:

```
To square :length
repeat 4[fd :length rt 90]
end
```

Create a procedure for a rectangle with variable length and width using :length and :width

Answer:

```
To rectangle :length :width
repeat 2[fd :length rt 90 fd
:width rt 90]
end
```

Create a procedure for an equilateral triangle of side 120. Use the procedure to write an algorithm to repeat and rotate the triangle.

Answer:

```
to triangle
repeat 3[fd 120 rt 120]
end
```

Avery Template: Name Badge Label, 8 per sheet | **Compatible Products**:15395, 25395, 42395, 45395, 48395, 5395, 8395, 88395, 85395.

Programming Turtle Logo | Procedures

I can create and debug an algorithm to create a procedure.		
I can write commands in the correct order.		
I can correct any mistakes.		
I can write a procedure.		
I can use the commands fd, bk, rt, lt, cs, penup, pendown and repeat.		

Programming Turtle Logo | Procedures

I can create and debug an algorithm to create a procedure.		
I can write commands in the correct order.		
I can correct any mistakes.		
I can write a procedure.		
I can use the commands fd, bk, rt, lt, cs, penup, pendown and repeat.		

Programming Turtle Logo | Procedures

I can create and debug an algorithm to create a procedure.		
I can write commands in the correct order.		
I can correct any mistakes.		
I can write a procedure.		
I can use the commands fd, bk, rt, lt, cs, penup, pendown and repeat.		

Programming Turtle Logo | Procedures

I can create and debug an algorithm to create a procedure.		
I can write commands in the correct order.		
I can correct any mistakes.		
I can write a procedure.		
I can use the commands fd, bk, rt, lt, cs, penup, pendown and repeat.		

Programming Turtle Logo | Procedures

I can create and debug an algorithm to create a procedure.		
I can write commands in the correct order.		
I can correct any mistakes.		
I can write a procedure.		
I can use the commands fd, bk, rt, lt, cs, penup, pendown and repeat.		

Programming Turtle Logo | Procedures

I can create and debug an algorithm to create a procedure.		
I can write commands in the correct order.		
I can correct any mistakes.		
I can write a procedure.		
I can use the commands fd, bk, rt, lt, cs, penup, pendown and repeat.		

Programming Turtle Logo | Procedures

I can create and debug an algorithm to create a procedure.		
I can write commands in the correct order.		
I can correct any mistakes.		
I can write a procedure.		
I can use the commands fd, bk, rt, lt, cs, penup, pendown and repeat.		

Programming Turtle Logo | Procedures

I can create and debug an algorithm to create a procedure.		
I can write commands in the correct order.		
I can correct any mistakes.		
I can write a procedure.		
I can use the commands fd, bk, rt, lt, cs, penup, pendown and repeat.		