



Junior

omputer



code-it.co.uk

cience

Computer Science Concepts

- Decomposition
- Repetition repeat x loops
- If selection
- Random choice

Maths Concepts

- Using Y axis to position turtle
- Angles
- Random length and angle

Design a Tree in Logo

Program Aim Write a program that draws a different looking tree every time the program is run.

Hook

A games designer want to create a forest of trees as a backdrop to a scene in a game. To pay someone to draw these individually would be far too expensive. Write a program that draws a different looking tree every time the program is run.

Differentiation and Assessment for Learning At the beginning of each session the *learning intention sheet* is shared and the learning journey expanded through success criteria. Pupils feed their progress back to the teacher through annotating this sheet with smiley faces at the end of the lesson. Teachers can also annotate the sheet to indicate those who need more or less help in future lessons. These extra resources can be found on the code-it.co.uk website.

Resources

Tree Pupil Booklet Main
Tree Pupil Booklet Able

12, Multi Tree

11, Format tree options

10, Use selection to randomise direction

9, Use repeat & random to code branches

8, Code a branch

7, Code the trunk

6, Draw and decompose a tree

5, Intro nesting pattern making

4, Intro procedures

3, Intro repeat command

2, Draw square , triangle, hexagon etc

1, Draw simple lines in commander FD RT LT

Learning Path

1, Draw simple lines in the commander

The commander can be found at the bottom of the program in FMSLogo. In many other programming languages it would be called the interpreter. We can type commands directly in on this line and we can see what was typed in the immediate past just above it. Demonstrate how you can use simple logo commands such as **forward 50 (FD 50) and right 90 (RT 90)**. Encourage pupils to take the line for a walk for a few minutes. Pupils may ask questions such as how do I go down. You may need to explain that all positions are relative to the arrow head turtle.

2, Draw square, triangle, hexagon etc

Set pupils the task of drawing a square. As pupils complete this challenge extend it by adding another 2D shape such as a triangle or hexagon. Leave the odd numbered sides other than the triangle such as the pentagon and heptagon to last. Once most pupils have done at least a couple move on.

3, Introduce the repeat command

Make the connection with the repeat x times Scratch block if pupils have used that previously. Create a square using the **repeat 4 [FD 100 RT 90]** in the commander.

5-7 session module of work



Design a Tree P2

3, Introduce the repeat command continued

If pupils have never encountered a repeat loop before then this is a great opportunity to help their perception by getting them dancing using repeated moves. Choose a popular dance that all the pupils know such as Gangnam Style and ask them to show you the steps. Identify what parts repeat and create a one word name or symbol for them. Pupils can now create a dance on their whiteboards using these symbols or names and indicating how many times these should be repeated (lasso x 4). They can dance each others creations. This might take only 5-10 minutes.

Now allow pupils time to recreate some of the regular 2D shapes created earlier using the repeat command.

Change the text size on the editor using the set command so pupils can see what you are doing

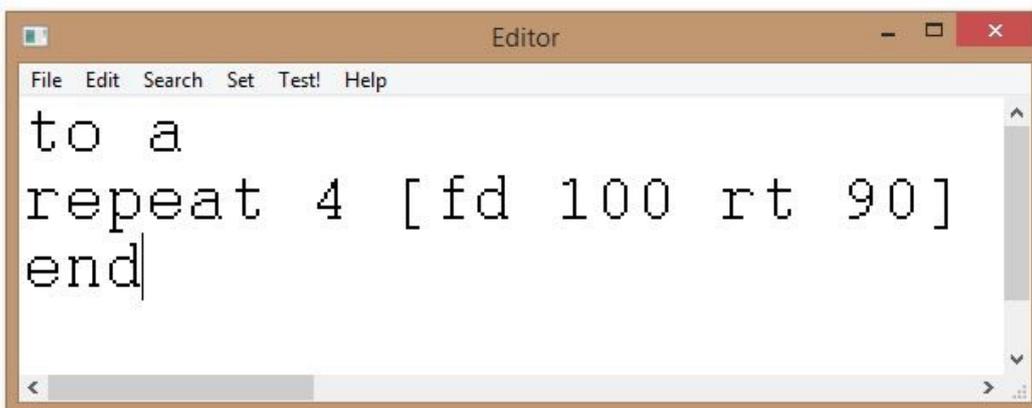
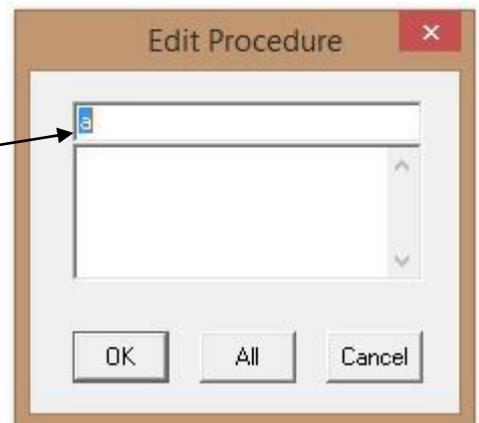
4, Introduce Procedures

Once most pupils have done at least one shape created using the repeat command in the commander move on and introduce the idea of keeping your creations by writing them as procedures. A procedure is a program that has been remembered by logo and can be run by directly typing its name into the commander or using it within another procedure.

To create a procedure File then Edit

Type in a name into the top the name must not have any spaces and cannot be a logo command. You couldn't use forward as a name for a procedure.

An editor will appear like this one below.



Show pupils how to type within the **to a** and **end** by going to the end of the first line and creating a space between using the return or enter keys.

Make sure pupils know that they mustn't add

anything to the top line or end line or delete either lines. There is an exception to this when creating some form of variables but pupils don't need that information yet. Encourage pupils to convert some of their shape programs into procedures and test them from the command line.



Design a Tree P3

5, Introduce Nesting using Patterns

After all pupils have written at least one procedure introduce the idea of putting one procedure inside another one. The easiest way to do this is to use their square or triangle and put them in a repeat command with a turn. Warn pupils not to put a procedure within itself. Allow pupils plenty of time to experiment.

```
File Edit Search Set Test! Help  
to pat1  
repeat 20 [square rt 20 triangle]  
end
```

6, Draw and Decompose a Tree

Before pupils can program a randomising tree they need to draw or examine a simple tree and decompose its main elements. Looking carefully at all the tree properties such as where they attach to other parts of the tree and what size they are. Share the hook outlined on P1. Hand out the tree design booklets one between two. Ask them to draw a tree (Able group) making sure the trunk is on the Y axis. Get main group to label the parts of the tree and answer the questions. As pairs finish encourage them to work on the properties of the trunk, branch & branches on the next page (Able group) and write a trunk and branch algorithm (Main group). Once a suitable amount of detail has been added let pupils move on to coding the trunk. Encourage pupils to work in pairs but code separately. You may need to remind some pupils about the pen up and pen down button.

A tree is a good thing to draw for younger pupils as precision in angles is not required. The able group have a much wider choice of commands and are asked to work out the properties of their trunk and branch before coding. The main group have a more limited selection of blocks and need to think of a simple algorithm for trunk and branch before coding. Generally I would use their Maths groups and any pupils who attend programming clubs as a way to group pupils.

7, Code the trunk

There are many ways to do this. Our job as teacher is to encourage them to experiment and find a way that suits them but still fulfils the design criteria. The trunk is the obvious place to start as it is in effect just one line. Pupils will need a starting position (**SETPOS 0 -180**) and need to point up (**SETHEADING 0**) They could then use **FD** or use the **SETPOS** and a higher place on y axis than -180. **PU & PD** is crucial for this.

```
to trunk  
pu  
setpos [0 -150]  
pd  
setpos [0 -20]  
pu  
end
```



Design a Tree P4

8, Code a branch

Encourage able pupils to use the properties of the tree sheet to help them draw one branch. This will need a start position, a number of turns and move distances between the turns. Every branch design in your able group could be different from every other one. As before pupils need lots of time to work this out for themselves. Encourage your main ability pupils to refer back to their algorithms they created earlier. These will push them towards using Y axis for the trunk and move and angle turns for the branch.

At some time during this block you will need to remind pupils of nesting using one procedure inside another. <http://www.ictvideohelp.co.uk/logo/nesting/nesting.html> Show pupils how we could write a procedure called tree and put procedures called truck and branch inside it.

9, Use repeat and random to code multiple branches

Introduce the RANDOM command as a way of changing each branch so it is drawn differently RANDOM 70 will draw a line between 0 and 69 long. Give pupils time to use this in their planning. They can now type FD RANDOM 50 and the line will go forward between 0 and 49 RT RANDOM 90 will turn right between 0 and 89 degrees.

Once pupils have experimented with random in one branch ask them how they can repeat their first branch in a different position? Can they find a way to repeat it without creating new code for every new branch? Remind pupils of the repeat command if they need reminding.

<http://www.ictvideohelp.co.uk/logo/repeat/repeat.html>

```
to branch2
repeat 20 [branch]
end
```

Repeat and nesting code example

```
to branch
pu
setpos [0 -20]
pd
setheading 0
fd 70
lt 30
fd 90
fd 20
rt 40
fd 80
pu
end
```

Branch code example



Design a Tree P5

```
to branch  
pu  
setpos [0 -20]  
pd  
setheading 0  
leftorright  
fd random 70  
lt random 30  
fd random 90  
leftorright  
fd random 20  
rt random 40  
fd random 80  
pu  
end
```

**Branch with random code
and leftorright procedure
nested inside**

10, Use IF selection to randomise the direction of turn right or left
Show pupils how it is possible to create code to randomly choose 0 or 1 and then check to see which one was chosen before turning right if 1 was chosen and left if 2 was chosen.

```
to leftorright  
make "lr random 2  
if :lr = 1 [rt random 70]  
if :lr = 0 [lt random 70]  
end
```

NOTE making a variable and using it use different prefixes " and :

Leftorright selection make "lr is a variable

11, Format Tree Options

There are lots of extension options we can include such as colour of tree. Thickness of trunk/branches. Can we add sub branches? Can we add leaves ?

12, Multi Tree Options

Can pupils code a forest of trees

I use FMS Logo which can be downloaded free here

<http://sourceforge.net/projects/fmslogo/>

Logo video help files can be found here

<http://www.ictvideohelp.co.uk/logo.html>