Structure	Code.org (Blockly)	Scratch	Pencil Code (Blocks)	Pencil Code (Text)	LEGO EV3	App Inventor	Python (CMU)
Events  An event runs code in response to user input or something that happens in a program.	when up 7 arrow  when actor 17 clicked  when actor 17 touches actor 27	when space ▼ key pressed  when this sprite clicked	<pre>click (e) -&gt;   moveto e  button 'Click', -&gt;   write 'clicked'  keydown 'X', -&gt;   write 'x pressed'</pre>	<pre>1 click (e) -&gt; 2 moveto e 3 4 button 'Click', -&gt; 5 write 'clicked' 6 7 keydown 'X', -&gt; 8 write 'x pressed'</pre>		when Canvas1 · .Touched  x y touchedAnySprite  do  when Button1 · .Click  do  when AccelerometerSensor1 · .Shaking  do	<pre>1 def onMousePress(x,y): 2     #Do something 3 4 def onKeyPress(key): 5     if key == 'space': 6         #Do something 7 8 def onStep(): 9     #Do something</pre>
Loops  A loop repeats a set of commands a fixed number of times or until a condition is met.	repeat ??? times do  while there is a hole ∨ do	repeat 10	for x in [010]  forever 10, ->	<pre>1 for [13] 2  fd 100 3 4 for x in [010] 5  label x 6 7 forever 10, -&gt; 8  fd 10 9  rt 3</pre>		for each number from to 5 by 11 do for each item in list get global myList do while test do	<pre>1 while y &lt; 400: 2     #Do something 3 4 for i in range(10): 5     #Do something 6 7 for element in group: 8     #Do something</pre>
Conditionals  A conditional splits a program's flow into two or more paths, based on a specified condition.	if nectar V = V 0 do if nectar V = V 1 do else	if then  if then  else	<pre>if x is 1  x = 0  thide() else show()  hide() hide() hide()</pre>	<pre>1 if x is 1 2    x = 0 3 4 if x is 10 5    hide() 6 else 7    show() 8 9 if (touches ball) 10    hide()</pre>		o if then else if then else	<pre>1 if x &lt; 100: 2     #Do something 3 4 if x&lt;100: 5     #Do something 6 elif x&lt;300: 7     #Do something 8 else: 9     #Do something</pre>
Functions  A function names a reusable set of instructions for performing a specific task or calculation.	when run move and get nectar  Function  move and get nectar  turn right ov  move forward v  get nectar  move backward v  turn left ov	define My Procedure input say input for 2 seconds  My Procedure "Hello world!"	hexagon =-> pen ▼blue for [16] fd ▼100 rt ▼60  do hexagon	<pre>1 hexagon =-&gt; 2  pen blue 3 for [16] 4  fd 100 5  rt 60 6 7 do hexagon</pre>	Functio 3	to myProcedure do  call myProcedure	<pre>1 def myFunction(a, b): 2          c = a * b 3          return c 4          myFunction(5, 7) 6          myFunction(2, 6) 7</pre>
Variables & Lists  A variable stores a single value. A list stores multiple values, each with its own index.		set my variable ▼ to 0  change my variable ▼ by 1  item 1 of My List ▼	<pre>label x  list = ["cheese", "milk", "butter"]  label list[0]  list[3] = "yogurt"</pre>	<pre>1 x = 1 2 3 label x 4 5 list = ["cheese", "milk", "butter"] 6 7 label list[0] 8 9 list[3] = "yogurt"</pre>		initialize global myVariable to 10  set global myVariable to 10  set Label1 . Text to get global myVariable initialize global myList to make a list select list item list get global myList index 1  length of list list get global myList index get global myList index 1	<pre>1 myVariable = 10 2  3 myShape.centerX = 200 4  5 myList = ['a','b','c'] 6  7 print(myList[0]) 8  9 for item in myList: print(item)</pre>
Objects Objects group data about the position, di- mensions, properties, and behaviors of sprites and shapes.						Components  Screen1  Button1  Slider1  A Label1  Canvas1  AccelerometerSensor1  Sound1  TextToSpeech1	