In the process of creating projects with Scratch, young people develop as computational thinkers. They learn concepts, engage in practices, and develop perspectives they can use to express their ideas with code. This list features fundamental computational concepts that are supported in Scratch.

<table>
<thead>
<tr>
<th>Concept</th>
<th>Explanation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>sequence</td>
<td>To create a program in Scratch, you need to think systematically about the order of steps.</td>
<td><img src="image" alt="Sequence Example" /></td>
</tr>
<tr>
<td>iteration (looping)</td>
<td><em>forever</em> and <em>repeat</em> can be used for iteration (repeating a series of instructions)</td>
<td><img src="image" alt="Iteration Example" /></td>
</tr>
<tr>
<td>random</td>
<td><em>pick random</em> selects random integers within a given range.</td>
<td><img src="image" alt="Random Example" /></td>
</tr>
<tr>
<td>conditional statements</td>
<td><em>if</em> and <em>if else</em> check for a condition.</td>
<td><img src="image" alt="Conditional Statements Example" /></td>
</tr>
<tr>
<td>boolean logic</td>
<td><em>and</em>, <em>or</em>, <em>not</em> are examples of boolean logic</td>
<td><img src="image" alt="Boolean Logic Example" /></td>
</tr>
<tr>
<td>variables</td>
<td>The variable blocks allow you to create variables and use them in a program. Variables can store numbers or strings. Scratch supports both global and object-specific variables.</td>
<td><img src="image" alt="Variables Example" /></td>
</tr>
<tr>
<td>lists (arrays)</td>
<td>The list blocks allow for storing and accessing a list of numbers and strings. This kind of data structure can be considered a “dynamic array.”</td>
<td><img src="image" alt="Lists Example" /></td>
</tr>
<tr>
<td>string manipulation</td>
<td>You can change or get information about strings of letters using <em>length of, letter of,</em> and <em>join.</em></td>
<td><img src="image" alt="String Manipulation Example" /></td>
</tr>
</tbody>
</table>

To learn more computational thinking, see [scratched.gse.harvard.edu/ct/](http://scratched.gse.harvard.edu/ct/)
<table>
<thead>
<tr>
<th>Concept</th>
<th>Explanation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>event handling</td>
<td><em>when key pressed</em> and <em>when sprite clicked</em> are examples of event handling – responding to events triggered by the user or another part of the program.</td>
<td><img src="image" alt="Event Handling Example" /></td>
</tr>
<tr>
<td>threads (parallel execution)</td>
<td>Launching two stacks at the same time creates two independent threads that execute in parallel.</td>
<td><img src="image" alt="Threads Example" /></td>
</tr>
<tr>
<td>coordination and synchronization</td>
<td><em>broadcast</em> and <em>when I receive</em> can coordinate the actions of multiple sprites. Using broadcast and wait allows synchronization.</td>
<td><img src="image" alt="Coordination and Synchronization Example" /></td>
</tr>
<tr>
<td>procedures</td>
<td>The <em>Make a Block</em> feature lets you define a new block that you use in your scripts. Defining a block can also be called naming a procedure. The Make a Block feature allows reusing code within a sprite, and can support modularity and abstraction.</td>
<td><img src="image" alt="Procedures Example" /></td>
</tr>
<tr>
<td>parameters</td>
<td>When you <em>Make a Block</em>, you can add number, string, and boolean parameters. This allows your new block to have inputs. You can use blocks with parameters to make recursive procedures.</td>
<td><img src="image" alt="Parameters Example" /></td>
</tr>
<tr>
<td>cloning</td>
<td><em>create clone</em> makes a copy of a sprite that exists until the project stops running. You can use it to dynamically create many copies of the same sprite with the same code.</td>
<td><img src="image" alt="Cloning Example" /></td>
</tr>
<tr>
<td>physical sensing</td>
<td>Blocks such as <em>loudness</em> allow interactions with microphones and other physical interfaces.</td>
<td><img src="image" alt="Physical Sensing Example" /></td>
</tr>
</tbody>
</table>

To learn more about Scratch see scratch.mit.edu/about