

Algorithms × Scripture



Twelve programming ideas, each paired with one passage that truly fits it. Quotes are from the King James Version. The full lessons are free at christcoded.bible/learn.

idea	the passage	in one line	in code
Variables <i>the basics</i>	<i>Genesis 17:5</i> thy name shall be Abraham	A name tied to a value. Reassign it and the old value is gone.	<code>name = "Abraham"</code>
If / else <i>the basics</i>	<i>Joshua 24:15</i> choose you this day whom ye will serve	A true-or-false question picks one road. Only one branch runs.	<code>if answer == "the LORD":</code>
For loops <i>the basics</i>	<i>Joshua 6:3</i> go round about the city once. Thus shalt thou do six days.	Repeat something a known number of times.	<code>for day in range(6):</code>
While loops <i>the basics</i>	<i>1 Kings 18:43</i> Go again seven times	Keep going until a condition changes.	<code>while sky != "cloud":</code>
Lists <i>the basics</i>	<i>Matthew 10:2</i> The first, Simon, who is called Peter	Items in order. Counting starts at zero.	<code>apostles[0] # "Simon"</code>
Functions <i>the basics</i>	<i>Genesis 41:26</i> The seven good kine are seven years	Something goes in, a value comes back. Write it once, call it often.	<code>def interpret(dream):</code>
Debugging <i>the basics</i>	<i>Matthew 7:5</i> first cast out the beam out of thine own eye	Check your own code first. Print what's inside to see clearly.	<code>print(x) # look first</code>
Dictionaries <i>collections</i>	<i>Genesis 2:19</i> whatsoever Adam called every living creature, that was the name thereof	Look a value up by its key.	<code>names["lion"]</code>
Booleans <i>collections</i>	<i>Matthew 5:37</i> let your communication be, Yea, yea; Nay, nay	True or False, nothing in between.	<code>ready = True</code>
Recursion <i>collections</i>	<i>Matthew 1:2</i> Abraham begat Isaac; and Isaac begat Jacob	Solve it by handing a smaller copy of the problem to yourself.	<code>return [p] + line(parent)</code>
Missing number <i>searching</i>	<i>Luke 15:4-7</i> leave the ninety and nine... until he find it	Expected total minus actual total. One pass.	<code>n*(n+1)//2 - sum(nums)</code>
Breadth-first search <i>searching</i>	<i>Exodus 13:17</i> God led them not through the way of the land of the Philistines, although that was near	Explore 1 step away, then 2, then 3: the first arrival is the shortest path.	<code>node = queue.popleft()</code>

How fast? Big-O in one breath

- `O(1)` same time, however big the input (a list lookup by index)
- `O(log n)` halve the problem each step (binary search)
- `O(n)` look at everything once (sum a list)
- `O(n2)` compare everything with everything (the slow lost sheep)

Beginner traps

- `=` assigns, `==` asks
- Indentation decides what's inside an `if` or a loop
- Lists start at `0`, not 1
- `"The"` and `"the"` are different strings