

# Simple Quantum Computing in 150 Lines of Python

[Originalartikel](#)

[Backup](#)

<html> <p>What does it take to build a quantum computer? Lots of exotic supercooled hardware. However, creating a simulator isn't nearly as hard and can give you a lot of insight into how this kind of computing works. A simulator doesn't even have to be complicated. Here's one that exists in about <a href="https://github.com/adamisntdead/QuSimPy" target="\_blank">150 lines of Python code</a>.</p> <p>You might wonder what the value is. After all, there are plenty of well-done simulators including <a href="https://hackaday.com/2018/01/24/quantum-weirdness-in-your-browser/">Quirk</a> that we have looked at in the past. What's charming about this simulator is that with only 150 lines of code, you can reasonably read the whole thing in a sitting and gain an understanding of how the different operations really affect the state.</p> <p>One thing to note is that the 150 lines quoted includes comments, so the actual code bulk is even less. The majority of the program handles the quantum register and a generic way to apply a matrix to the quantum state. That leaves most of the operations as just simple definitions of matrices. For example:</p> <pre class="brush: python; title: ; notranslate" title=" "> # Hadamard Gate 'H': np.multiply(1. / np.sqrt(2), np.matrix([ [1, 1], [1, -1] ])),</pre> <p>Using the simulator involves writing more Python code, so really this is a library more than a simulator like Quirk. For example, here's part of the example code included:</p> <pre class="brush: python; title: ; notranslate" title=" "> ##### # Swap 2 Qubits # ##### # Here, We Will Apply a Pauli-X Gate / NOT Gate # To the first qubit, and then after the algorithm, # it will be swapped to the second qubit. Swap = QuantumRegister(2) # New Quantum Register of 2 qubits Swap.applyGate('X', 1) # Apply The NOT Gate. If Measured Now, it should be 10 # Start the swap algorithm Swap.applyGate('CNOT', 1, 2) Swap.applyGate('H', 1) Swap.applyGate('H', 2) Swap.applyGate('CNOT', 1, 2) Swap.applyGate('H', 1) Swap.applyGate('H', 2) Swap.applyGate('CNOT', 1, 2) # End the swap algorithm print('SWAP: |' + Swap.measure() + '>') # Measure the State, Should be 01 </pre> <p>If you want to learn more about quantum computing, we did a series recently that started <a href="https://hackaday.com/2018/01/24/quantum-weirdness-in-your-browser/">simply</a>, looked at <a href="https://hackaday.com/2018/01/31/quantum-communications-in-your-browser/">communications</a>, and wrapped up with a look at <a href="https://hackaday.com/2018/02/07/quantum-searching-in-your-browser/">Grover's algorithm</a>.</p> </html>

From:  
<https://schnipsl.qgelm.de/> - **Qgelm**

Permanent link:  
<https://schnipsl.qgelm.de/doku.php?id=wallabag:simple-quantum-computing-in-150-lines-of-python>

Last update: **2021/12/06 15:24**

