

# Get started developing for quantum computers today!

Dr. Sarah Kaiser | Azure Community Live!

→ [sckaiser.dev](https://sckaiser.dev) | @crazy4pi314

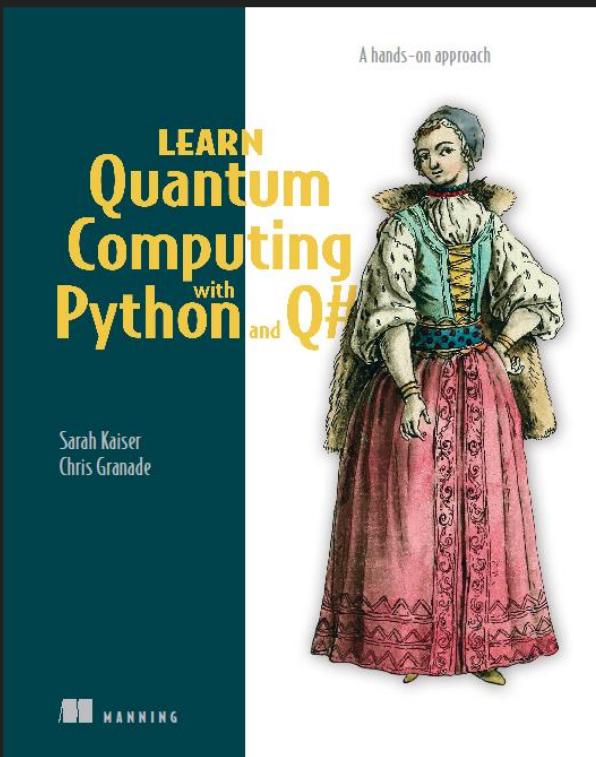
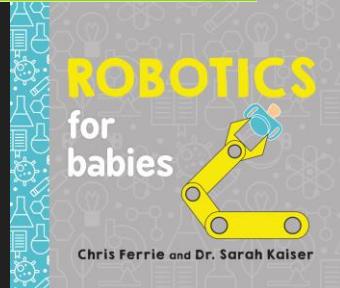
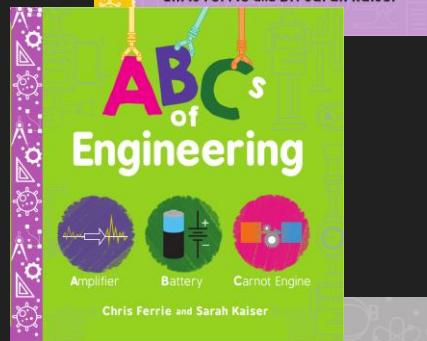
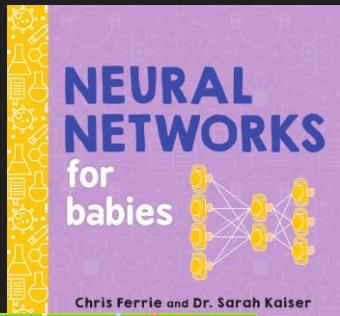
## How is Azure involved here?

Azure Quantum PREVIEW

Azure Quantum is a new service coming soon to Azure that will offer a number of different ways to explore quantum impact to your applications.

The Quantum Development Kit and Q# is the entry point!

# about\_me.md



# What have you heard about Quantum Computing?

Put it in the chat!



Antimatter seen in two places at once thanks to quantum experiment

The double-slit experiment is a classic demonstration that all particles of light and matter are also waves - and now...

NS New Scientist · 23 hours ago · ⚡

Physics



Which came first, the chicken or the egg?  
Both, quantum physics says

New research from University of Queensland physicists shows events aren't always as orderly as one might imagine...

USA TODAY · 16 hours ago · ⚡

Quantum



This Mad New Quantum Experiment Breaks The Idea of 'Before' And 'After'

For around a century it's been thought that particles don't have defined properties until we nail them down with a ...

ScienceAlert · 9 hours ago · ⚡

spooky. At least that's what Albert Einstein said when de...

Phys.org · 6 hours ago · ⚡

Quantum

Can a New Blockchain Counter the Quantum Computing Threat

There has been a strong focus on the effects of quantum computing on blockch...



NewsBTC · 57 mins ago



Could Quantum Computing Be the End of Free Will?

Faster, more powerful computing has the potential to revolutionize fields from drug delivery to freight transpor...

The Atlantic · 9 hours ago · ⚡

Quantum mechanics



There Are Two Kinds of Water in Every Glass, Thanks to Quantum Physics

In every glass of water you drink, there are two kinds of H<sub>2</sub>O. And scientists have shown that they have significa...

Live Science · 17 hours ago · ⚡

Quantum mechanics



The Next Web · 8 hours ago · ⚡

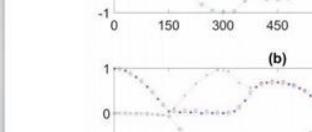


The promise of quantum computing supremacy is bunk

Google recently announced that the first demonstration of quantum supremacy could be just a few months away. I ...

The Next Web · 8 hours ago · ⚡

Quantum



Braiding may be key to using quantum computing

Over the past few years, physicists have found a new form of matter called time crystal...

Phys.org · 11 hours ago · ⚡

Quantum mechanics

Closed Loophole Confirms Unreality of the Quantum World

A quickly closed loophole has proved the "great smoky dragon" of quantum mechanics...

Quanta Magazine · 4 hours ago

# There is a lot of info out there

Revised Schrödinger's cat experiment challenges reality

Quantum mechanics has produced its share of weird ideas, not least of which is what's probably the world's...

We might only see time because we can't think in quantum physics

Humans need much more information to study a problem backwards in time than forwards, but a quantum computer...

ation Sci  
I Life

is sci-fi tech  
reported a p  
orthrop ... · 10

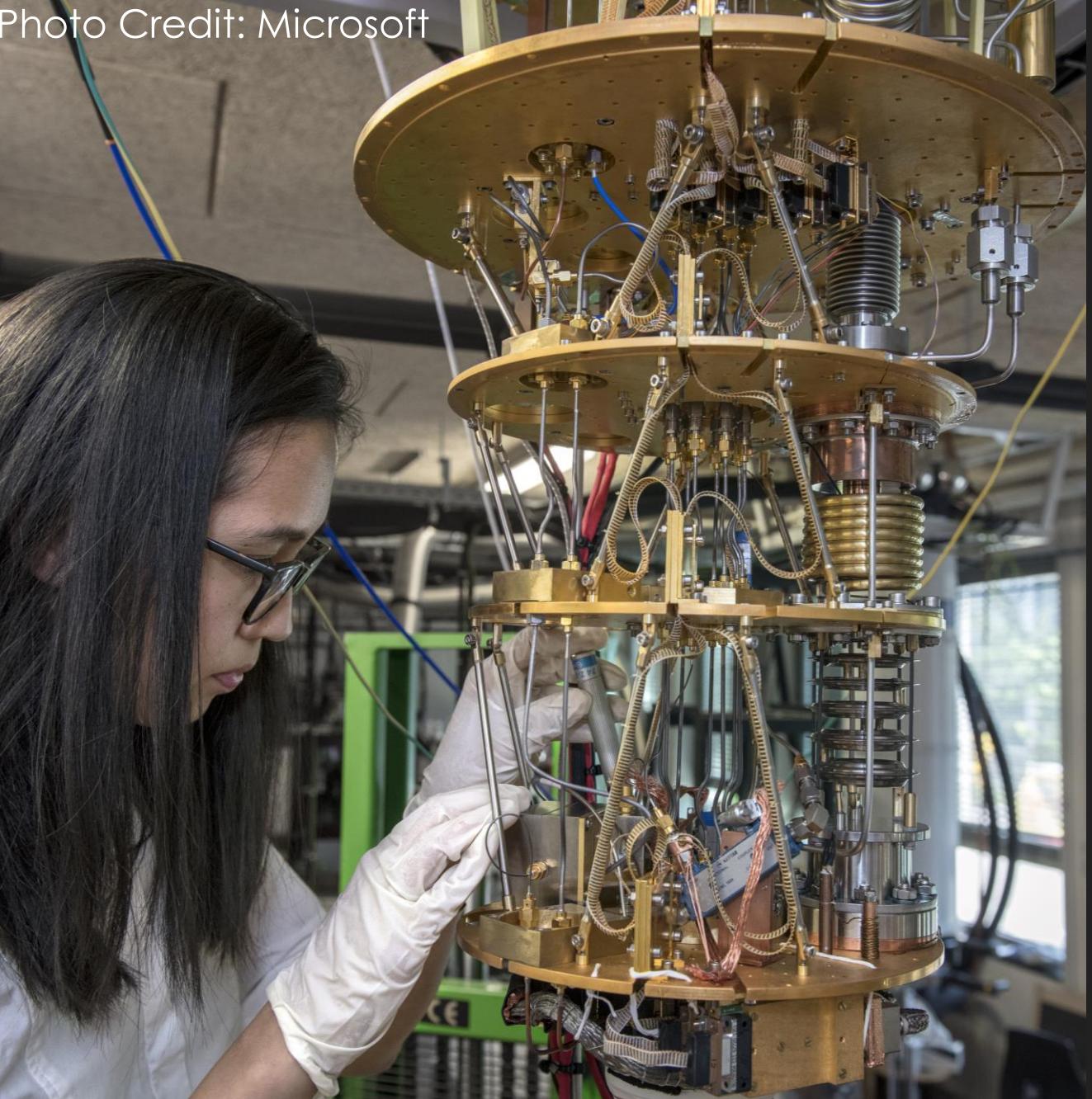
ics

ers with kn  
quantum mechanics break

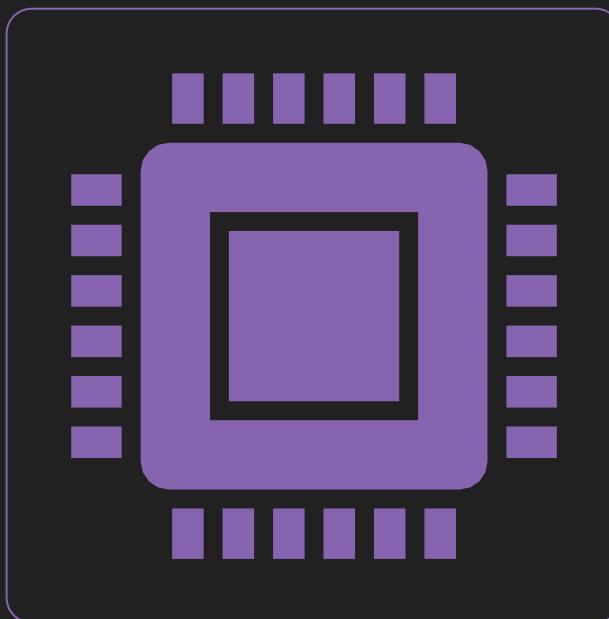
Quantum mechanics can't handle time... we know quantum mechanics.

Ars Technica · 3 hours ago · ⚡

# What in the heck is a quantum computer then?



# A new model of computing



- **Computer:**

A device for storing and manipulating information in the form of binary bits. The instructions for how to manipulate the information is given by a *program*.

- **Quantum Computer:**

A device for storing and manipulating information in the form of **quantum systems**. The instructions for how to manipulate the information is given by a *program*.

# A new model for information

We can simulate a computer using  $n$  bits.

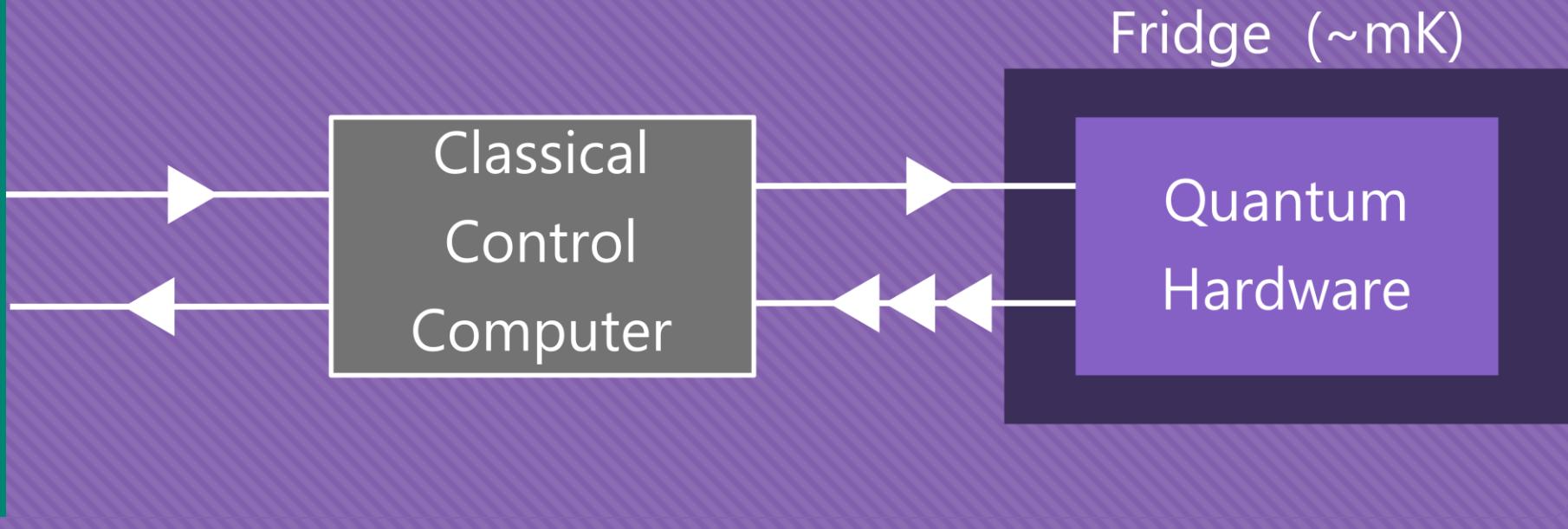
01000101000101001111011101...

We can simulate a **quantum computer** using  $2^n$  complex numbers.

$$\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{2^n-1} \end{bmatrix}$$

## Dev Machine

```
Quantum Program  
H(q[0]);  
CNOT(q[1], q[2]);  
CNOT(q[0], q[1]);  
H(q[1]);  
let x = M(q[0]);  
let z = M(q[1]);  
// ...
```



Think specific **hardware accelerator (GPU)**

# What might quantum computers be good for?

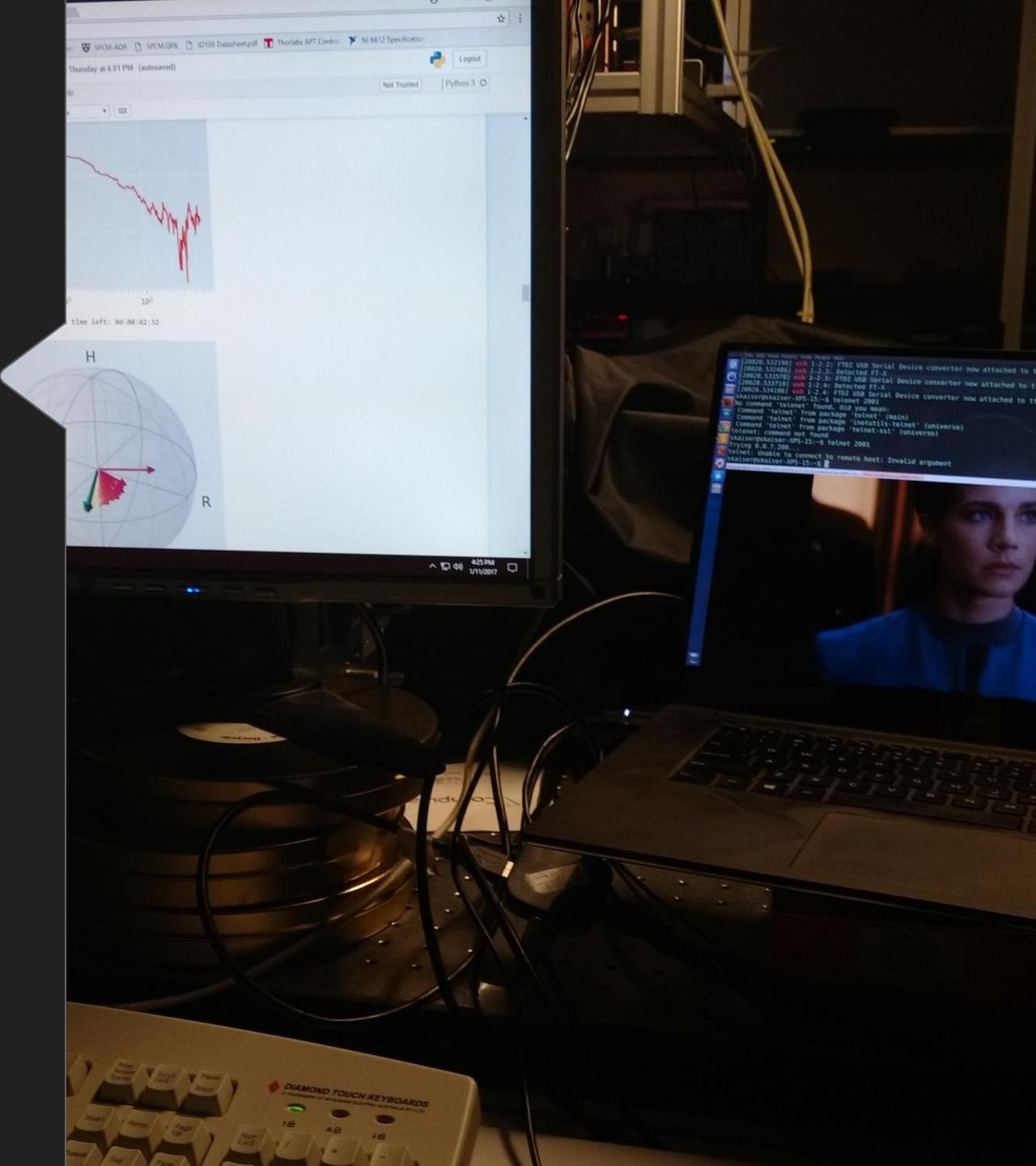
*Are these long-term or short-term applications?*

Quantum computing offers a lot of promise, including:

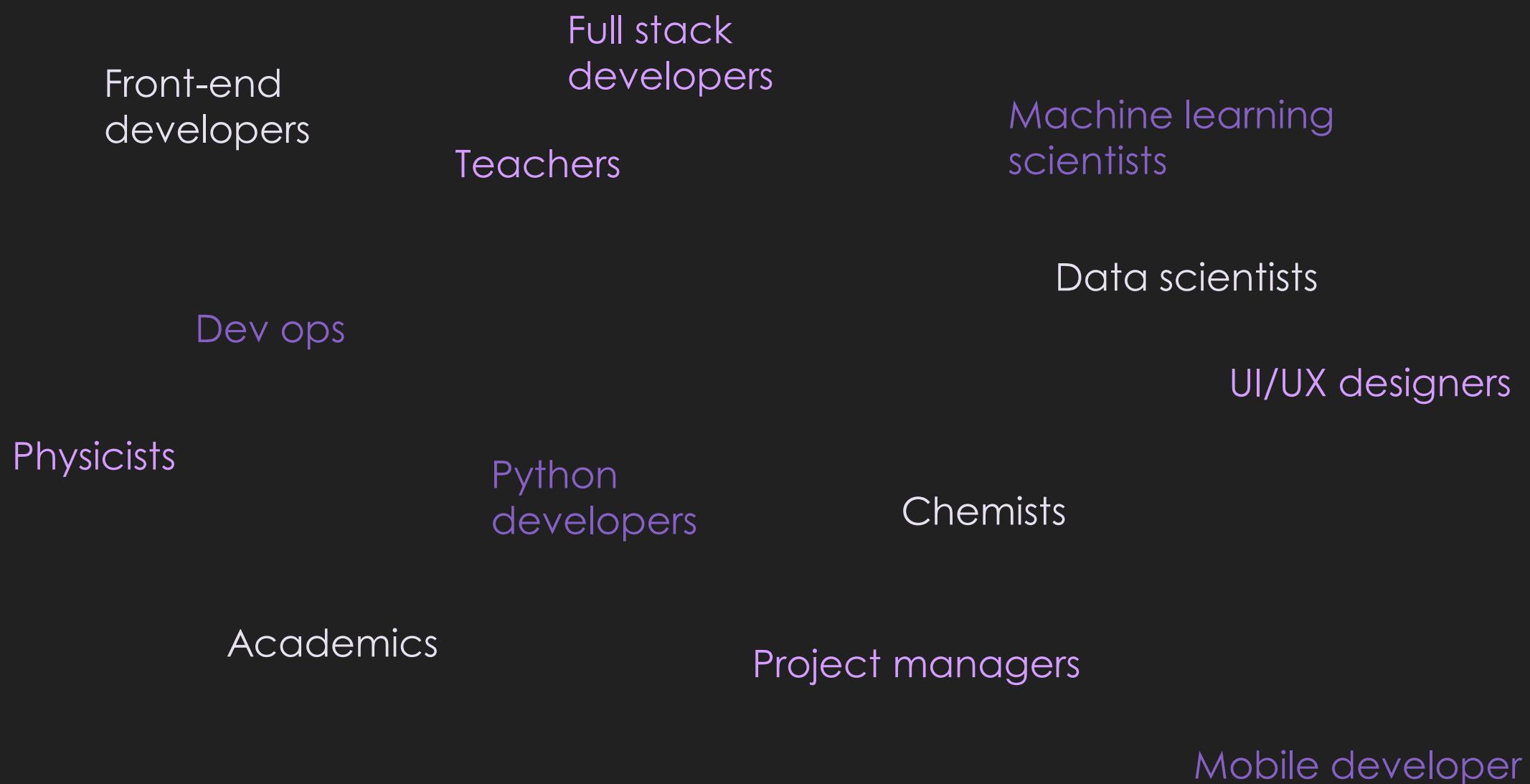
- Cryptography
- Machine learning
- Chemistry and materials science

We can understand timelines by building concrete **applications**.

- How many qubits will we need?
- What are the right quantum algorithms?
- Who is going to develop it?



# Who can program a quantum computer?





# EVERYONE!

Front-end  
developers

Full stack  
developers

Machine learning  
scientists

Dev ops

Data scientists

Physicists

Python  
developers

UI/UX designers

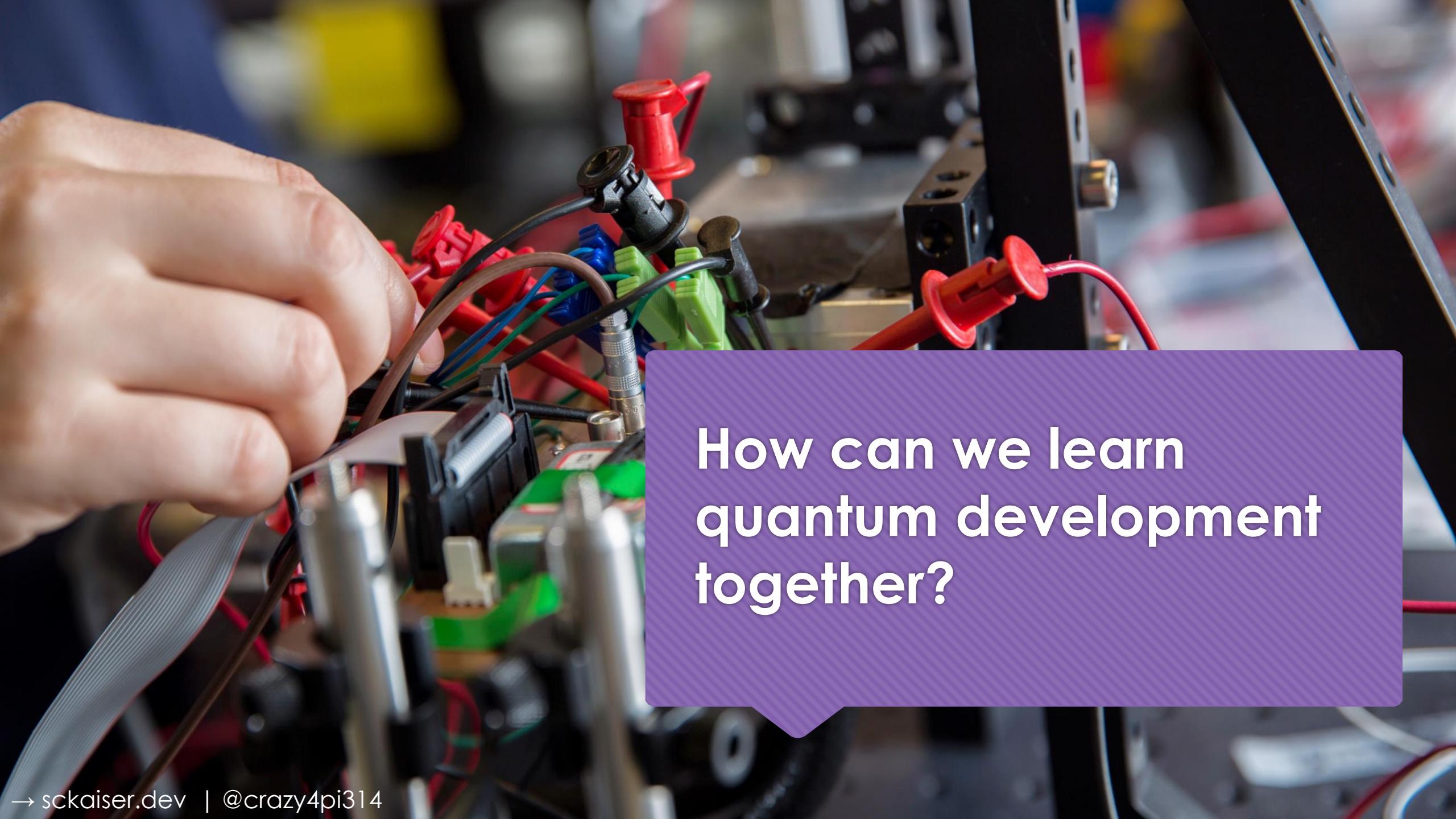
Academics

Chemists

Project managers

Mobile developer

Teachers



How can we learn  
quantum development  
together?



Algorithm.qs x

DeutschJozsa.csproj

```
You, a few seconds ago | 1 author (You)
1  namespace DeutschJozsa {
2      open Microsoft.Quantum.Diagnostics;
3      open Microsoft.Quantum.Intrinsic;
4      open Microsoft.Quantum.Measurement;
5
6      operation IsOracleBalanced(oracle : ((Qubit, Qubit) ⇒ Unit)) : Bool {
7          using ((control, target) = (Qubit(), Qubit())) {
8              H(control);
9              X(target);
10             H(target);
11
12             oracle(control, target);
13
14             H(target);
15             X(target);
16
17             return MResetX(control);
18         }
19     }
20
21     operation RunDeutschJozsaAlgorithm() : Unit {
22         Fact(not IsOracleBalanced(ZeroOracle), "Test failed for zero oracle.");
23         Fact(not IsOracleBalanced(OneOracle), "Test failed for one oracle.");
24         Fact(not IsOracleBalanced(IdOracle), "Test failed for id oracle.");
25     }
26 }
```

# Meet Q# and the Microsoft Quantum Development Kit

6      operation IsOracleBalanced(oracle : ((Qubit, Qubit) ⇒ Unit)) : Bool {

- Q# is a **domain-specific** programming language used for expressing quantum algorithms.

- The quantum development kit is **open source!**

# Q# is built on .NET Core

The .NET Core SDK is **cross-platform** so you can use Q# on your favorite operating system!

A screenshot of the Visual Studio IDE interface. On the left, the Solution Explorer shows a project structure with files like 'Operations.cs', 'djs.qs', and 'Program.cs'. The 'Operations.cs' file contains C# code for quantum operations. The 'djs.qs' file contains Q# code for teleportation. The 'Program.cs' file contains a main entry point. On the right, a terminal window shows the output of running the application with the command 'dotnet run'. The output indicates successful teleportation rounds 0 through 5.

```
operation IsOracleBalanced( oracle : ((Qubit, Qubit) => Unit)) : Bool {
    mutable result = Zero;
    using ((control, target) = (Qubit(), Qubit())) {
        H(control);
        X(target);
        H(target);
        oracle(control, target);
        H(target);
        X(target);
        set result = MResetX(control);
    }
    return result == One;
}

~/Quantum/Samples/src/Teleportation$ dotnet run
Round 0:      Sent False,      got False.
Teleportation successful!!

Round 1:      Sent True,      got True.
Teleportation successful!!

Round 2:      Sent True,      got True.
Teleportation successful!!

Round 3:      Sent True,      got True.
Teleportation successful!!

Round 4:      Sent False,      got False.
Teleportation successful!!

Round 5:      Sent True,      got True.
```

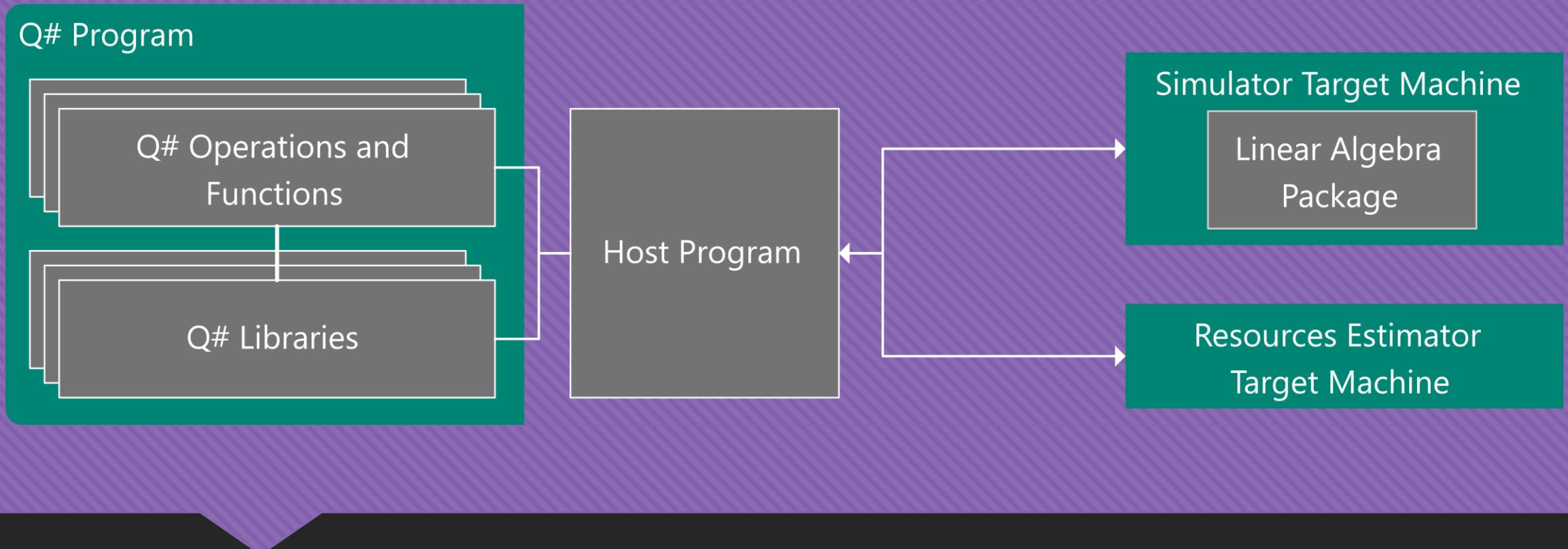
```
// Copyright (c) Microsoft Corporation. All rights reserved.
// Licensed under the MIT License.

namespace Microsoft.Quantum.Samples.Python {
    open Microsoft.Quantum.Primitive;
    open Microsoft.Quantum.Canon;

    function HelloWorld(pauli : Pauli) : () {
        Message($"Hello, world! {pauli}");
    }

    operation NoisyHadamardChannelImpl(depol : Double, target : Qubit) : () {
        body {
            let idxAction = Random([1.0 - depol, depol]);
            if (idxAction == 0) {
                H(target);
            } else {
                PrepareSingleQubitIdentity(target);
            }
        }
    }

    function NoisyHadamardChannel(depol : Double) : (Qubit => ()) {
        return NoisyHadamardChannelImpl(depol, _);
    }
}
```



# Q# Architecture



tomography-sample.py x



```
1 import qsharp
2 from qsharp.tomography import single_qubit_process_tomography
3
4 import matplotlib.pyplot as plt
5
6 import qutip as qt
7 qt.settings.colorblind_safe = True
8
9 experiment = qsharp.compile("""
10 open Microsoft.Quantum.Samples.Python;
11
12 operation Experiment(prep : Pauli, meas : Pauli) : Result {
13     return SingleQubitProcessTomographyMeasurement(
14         prep, meas, NoisyHadamardChannel(0.1)
15     )
```

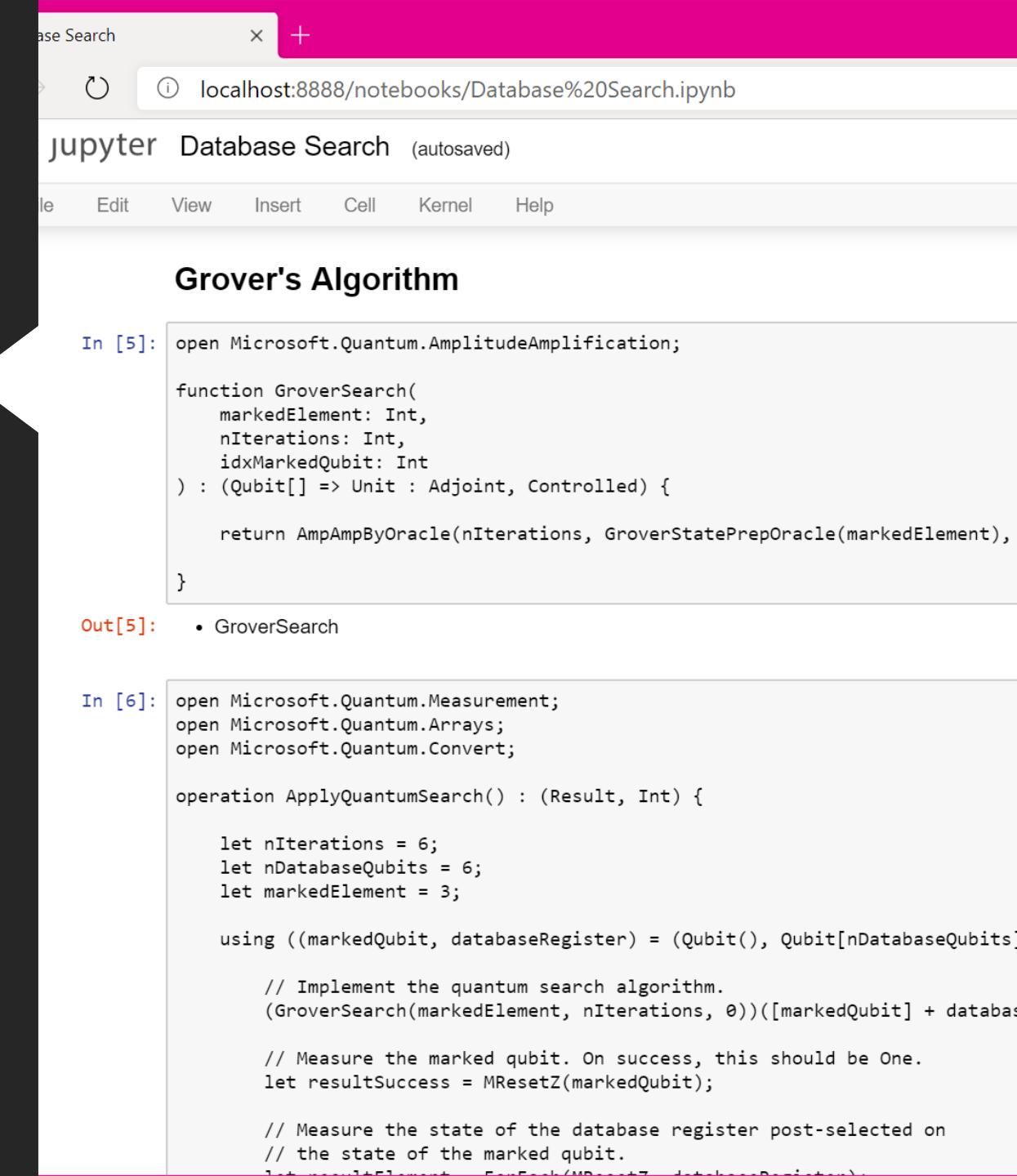
# Q# ❤️ Python

Q# and the QDK is **interoperable** with Python 3.6+!

# Jupyter Notebooks for Q#

- Jupyter Notebook support for Q# is provided by IQ# kernel.

- Exploring Q# with Jupyter allows users to focus on learning the new language instead of the rest of the stack.



The screenshot shows a Jupyter Notebook interface with the title "Database Search" and subtitle "jupyter Database Search (autosaved)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, and Help. The notebook content displays Grover's algorithm code in Q#:

```
In [5]: open Microsoft.Quantum.AmplitudeAmplification;

function GroverSearch(
    markedElement: Int,
    nIterations: Int,
    idxMarkedQubit: Int
) : (Qubit[] => Unit : Adjoint, Controlled) {

    return AmpAmpByOracle(nIterations, GroverStatePrepOracle(markedElement),
}

Out[5]: • GroverSearch
```

Below, another cell shows the implementation of the Grover search algorithm:

```
In [6]: open Microsoft.Quantum.Measurement;
open Microsoft.Quantum.Arrays;
open Microsoft.Quantum.Convert;

operation ApplyQuantumSearch() : (Result, Int) {

    let nIterations = 6;
    let nDatabaseQubits = 6;
    let markedElement = 3;

    using ((markedQubit, databaseRegister) = (Qubit(), Qubit[nDatabaseQubits])

        // Implement the quantum search algorithm.
        (GroverSearch(markedElement, nIterations, 0))([markedQubit] + database

        // Measure the marked qubit. On success, this should be One.
        let resultSuccess = MResetZ(markedQubit);

        // Measure the state of the database register post-selected on
        // the state of the marked qubit.
        let resultElement = FApply(MResetZ(databaseRegister))
```



# Azure Quantum

PREVIEW

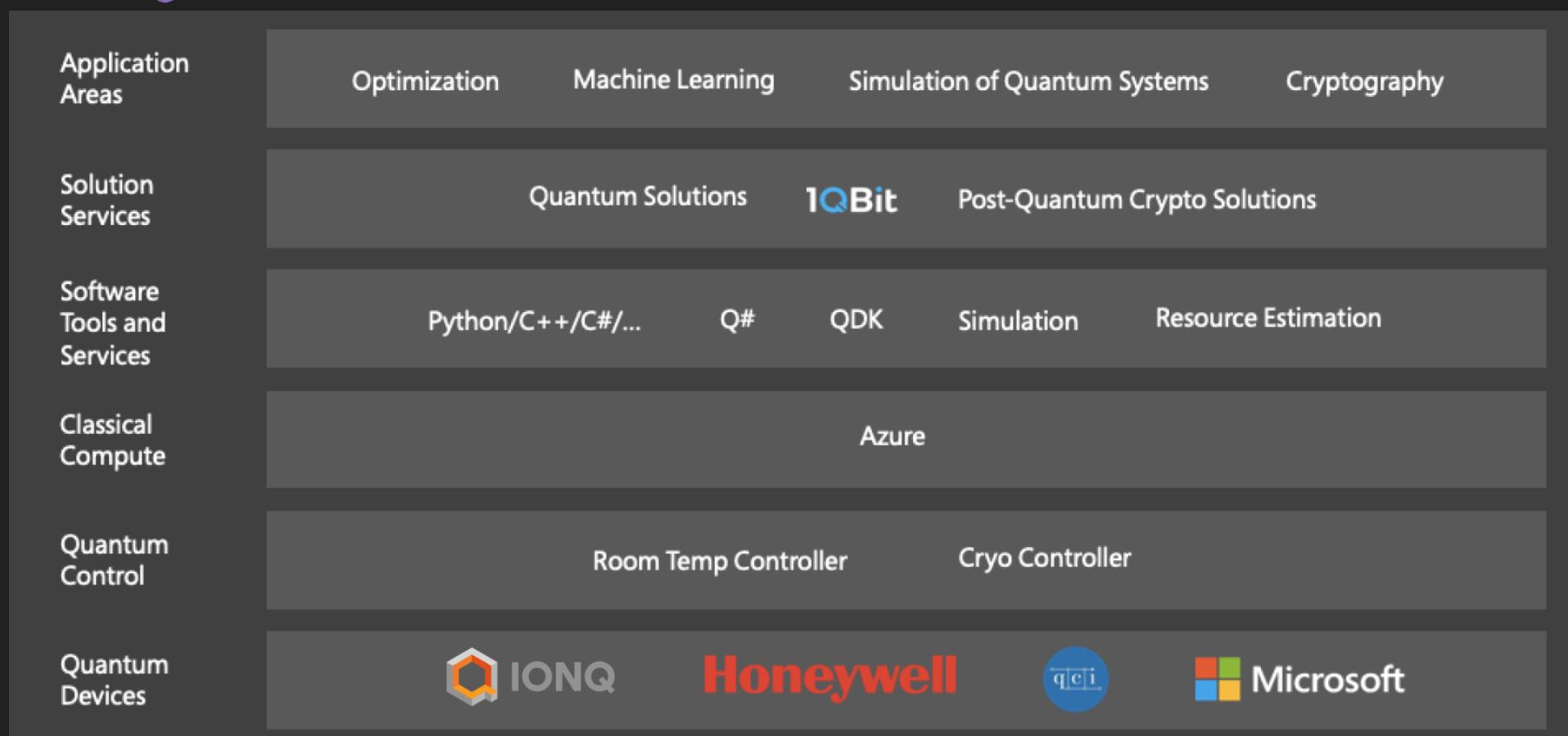
[azure.com/quantum](https://azure.com/quantum)

# Quantum Impact on Azure

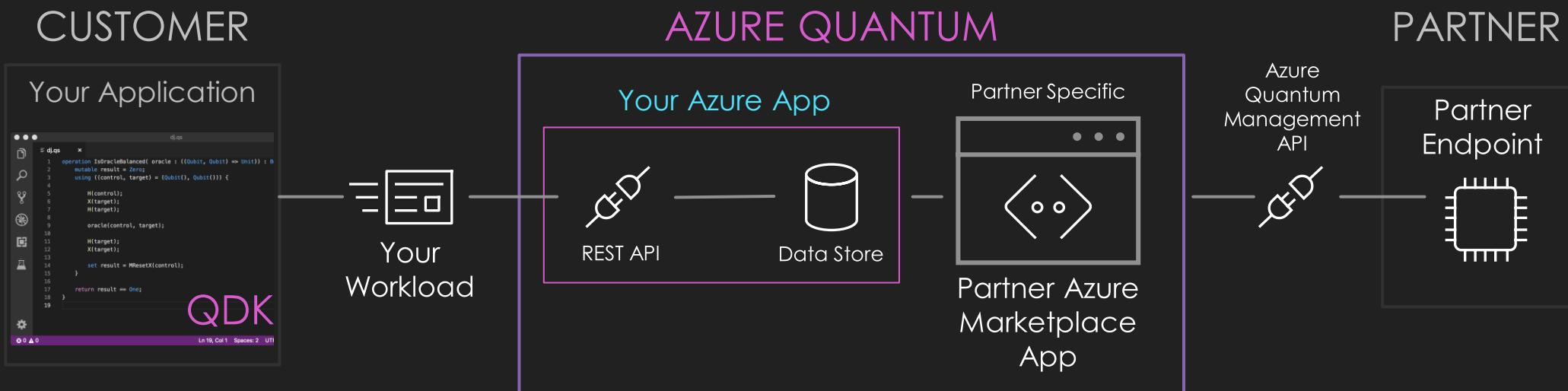
What services will be offered?

- **Quantum solutions** like pre-built solvers and algorithms that run at industrial scale
- **Quantum software**, including simulators and resource estimation tools, scaled by Azure compute
- **Quantum hardware** system options with a variety of different qubit architectures
- **Scale, security, and support** from Azure and trusted partners

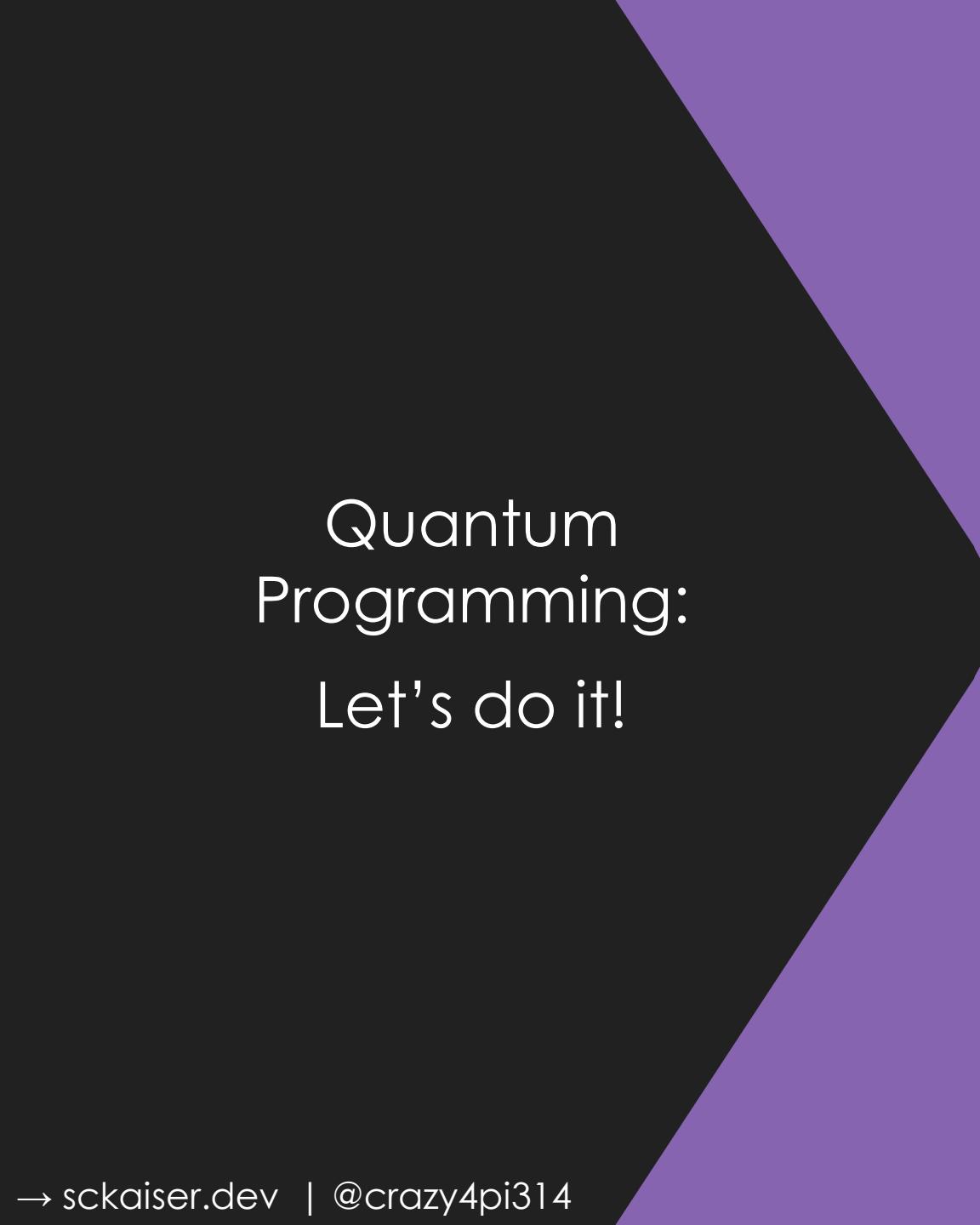
# Azure Quantum Stack



# Azure Quantum Workflow



<https://bit.ly/azurequantum-ignite>



Quantum  
Programming:  
Let's do it!

# Demo

# OK, but what can $\sqrt{-1}$ do?



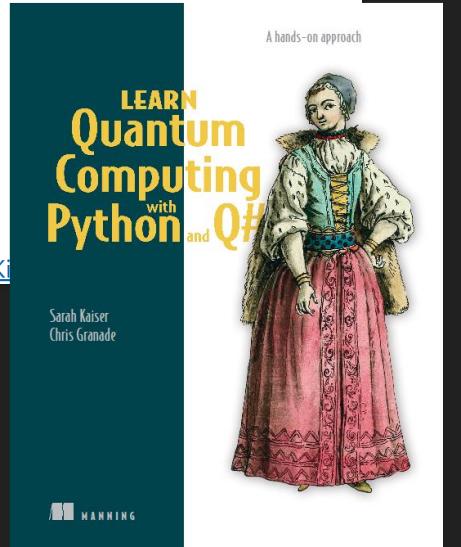
- Try Q# for yourself!
- Join a meetup!
- Learn by teaching
  - Write blogposts
  - Make tutorials
- Make the community better than you found it!
  - Contribute to docs
  - Fix bugs
  - Act intentionally to include everyone and expand the community

# Quantum programming resources

- Community projects:
  - [qsharp.community](https://qsharp.community)
  - [wiqca.dev](https://wiqca.dev)
- Documentation:
  - [docs.microsoft.com/quantum](https://docs.microsoft.com/quantum)
- Books:
  - [Learn Quantum Computing with Python and Q#](#)



The screenshot shows the Microsoft Quantum documentation page for "Writing a Quantum Program". The page includes a navigation bar with "Docs", a search icon, and a "Sign in" button. Below the title, it says "2017-12-10 • 13 minutes to read • Contributors" followed by a list of contributors and a "all" link. A sidebar titled "In this article" lists "What You'll Learn", "Creating a Bell State in Q#", and "Estimating Resources". The main content area starts with "What You'll Learn" and a list of four bullet points: "How to set up a quantum solution and project", "The components of a Q# operation", "How to call a Q# operation from C#", and "How to build and execute your quantum algorithm". It then moves to the "Creating a Bell State in Q#" section, which includes a note about installing the Microsoft Quantum Development Kit.



The book cover for "Learn Quantum Computing with Python and Q#" features a woman in a historical-style pink dress and blue vest. The title is in large yellow letters, and the authors' names are at the bottom. The publisher logo "HANING" is at the bottom left.

Connect with me!



@crazy4pi314

sckaiser.dev

- Twitch streaming quantum development Wed/Sat 12pm PDT
- WIQCA talk on quantum RAM and quantum machine learning this 13 May, 5 pm PDT : [wiqca.dev](#)

