

Welcome! We'll begin shortly.

If you'd like to participate in the hands-on activity, please set up your device now.

Please join our VEXnet Server

Network Name:

Main Stage

Password:

vex12345

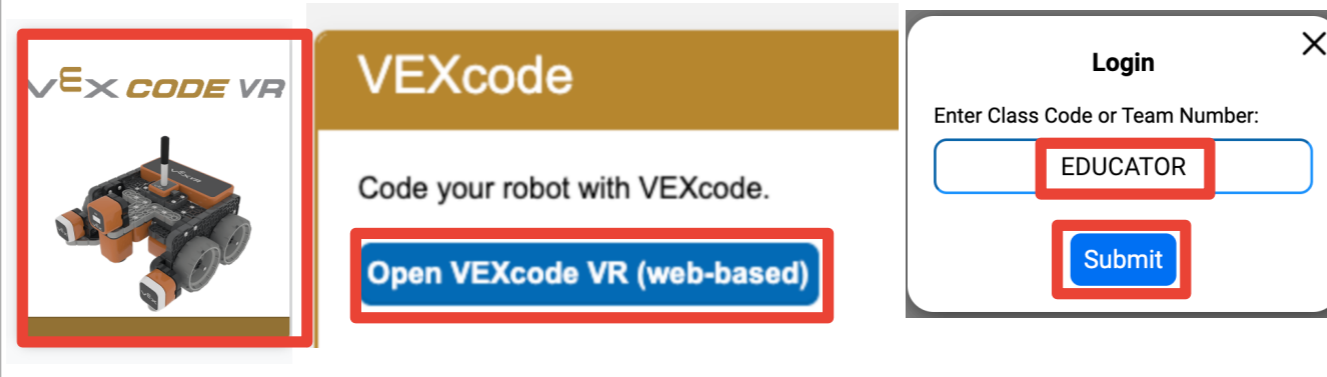
Once connected, visit

vexnet.com



The screenshot shows the VEXnet Server interface. At the top, it says "VEX VEXnet Server". Below that, it states: "Everything you need to teach with VEX can be accessed directly from this local server. The VEXnet Server is ideal for classrooms, workshops, and events where internet access is limited. Select a platform tile below to access VEXcode, curriculum, build instructions, and resources from the VEX Library and API Reference." There is a yellow warning banner that says: "Certificate warning: VEXnet SSL cert(s) are 14 days from expiring. Expires: 5/7/2026, 6:02:22 PM. Plan renewal." Below the banner is a grid of ten platform tiles, each with an icon and a label: VEX 123, VEX GO, VEX AIM, VEX IQ First year, VEX IQ Second year, VEX EXP, VEX VS, VEX CTE Workcell, VEX AIR, and VEX CODE VR.

1. Select VEXcode VR on the VEXnet Server home screen
2. Open VEXcode VR (web-based)
3. Login with code: EDUCATOR





VEXcode VR has been officially included in the **Learning Cabinet** – a curated discovery platform for EdTech solutions that demonstrate evidence of impact and potential to scale.

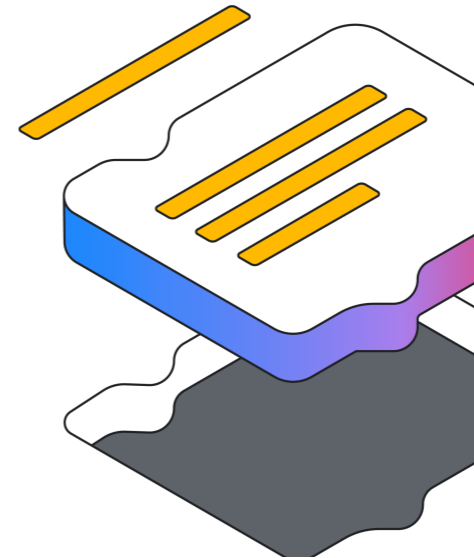
The Learning Cabinet is guided by the EdTech for Good Framework, created by **UNICEF**, the **Ministry for Foreign Affairs of Finland**, **Arm**, and the **Asian Development Bank (ADB)**, in collaboration with global partners.



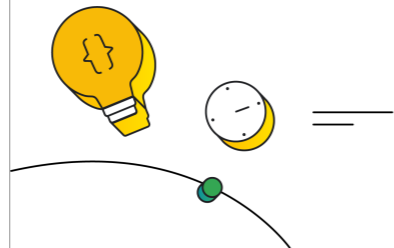
Python Coding in VEXcode VR

Catherine Kettner and Sardorbek Omonkulov

Software Developers
VEX Robotics



Agenda



- 01 Welcome & Introduction
- 02 Using Python in VEXcode VR
- 03 Castle Crasher Activity
- 04 Bonus Activity
- 05 Learn More

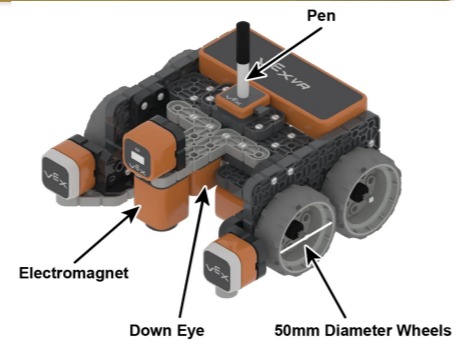
The VEX Continuum



The VEX Continuum

- The VEX Continuum provides a cohesive STEM learning plan from Kindergarten through high school and beyond
- It connects a series of VEX platforms through shared resources, curriculum, and materials. This continuity helps students grow their STEM skills year after year
- One of the biggest advantages is VEXcode continuity
 - Across VEX platforms, students see a familiar toolbar, coding structure, and interface
 - That consistency makes it easier to transition from elementary to middle to high school
- VEXcode VR complements these platforms by adding virtual learning and practice opportunities
 - VEXcode VR fits into the Continuum by giving students another way to practice, explore, and problem-solve
 - VR is a complement to physical platforms, not a replacement. That is why you'll see here that VEXcode VR spans across all of the continuum.

What is VEXcode VR?



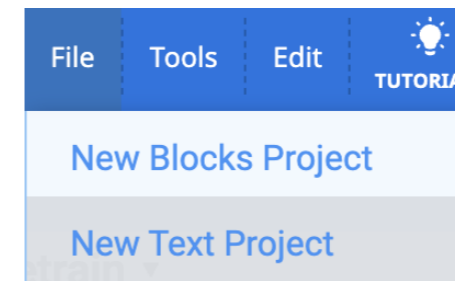
What is VEXcode VR?

- VR stands for Virtual Robot - Code a real VEX robot with a drivetrain, and preconfigured sensors.
- allows users to code a virtual robot, using either blockly's block-based coding interface or a text-based Python interface
- 20+ virtual robot playgrounds

VEXcode VR offers a Python text-based coding interface as an alternative to blocks.

Create a Python project:

1. Open the File menu
2. Select “New Text Project.”
3. Select the Castle Crasher+ Playground from the playground selection scene



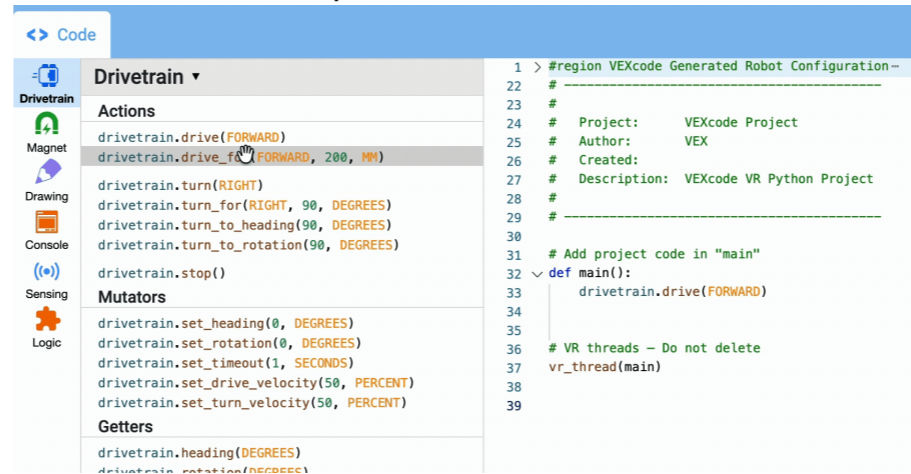
To start coding in Python, VEXcode supports conversion of blocks projects to Python, and creation of new Text Projects through the File > New Text Project menu

Python Toolbox

The Code Toolbox allows users to see available robot commands in VEXcode. Drag and drop commands from the toolbox into the workspace

Python Text Editor

The template of each project will include commented project description fields, and comments describing where project code should be added



Use Autocomplete to code efficiently

- Type in commands using VEXcode's smart Autocomplete. Only devices and commands available your project show up in the Autocomplete list.
- Syntax Highlighting (colorization) aids in reading and debugging code by visually distinguishing different code elements

```
30
31 # Add project code in "main"
32 def main():
33     dr
34     [x] drivetrain
35 # VR threads – Do not delete
36 vr_thread(main)
```

Activity : Castle Crasher+

Program your robot to crash into as many buildings as possible and knock them over.

You can work with the people around you, so feel free to partner up or work in a small group.

Start by:

- Use autocomplete to drive your robot forward
- Press **Start** to run your code

```
drivetrain.drive_for(FORWARD, 200, MM)
```



OBJECTIVE :

Knock over all the buildings in the **Castle Crasher+** Playground

TIME LIMIT: 10-15 minutes



Bonus: Find the hidden tool and pick it up using the magnet to knock over more castle blocks!

Share and discuss

Discuss with someone next to you or with your partner.

- What was different about Python versus Blocks?
- How did testing your code help?
- What challenges did you run into?
- How did you solve them?

Time: 5 minutes



Activity Part 2 : Castle Crasher+ beta-code.vex.com

Helpful Hints:

- 💡 Use the multiple Eye and Distance Sensors on the VR Robot+ to create effective algorithms for your projects.
- 💡 The red border around the playground can be detected by the Eye Sensors.



OBJECTIVE

:

Knock over ONLY rocks with the tool in the **Castle Crasher+** Playground

TIME LIMIT: 10-15 minutes

Share and discuss

Discuss with someone next to you or with your partner.

- What was your plan before you started?
- How did testing your code help?
- What challenges did you run into?
- How did you solve them?

Time: 5 minutes





Want to Learn More?

Join VEX PD+ as an All-Access Member!

- Schedule a 1-on-1 Session in VEX PD+
- **Take a VEX Masterclass** — Making the Most Of Teaching with VEXcode VR



To learn more about VEX PD+ go to pd.vex.com