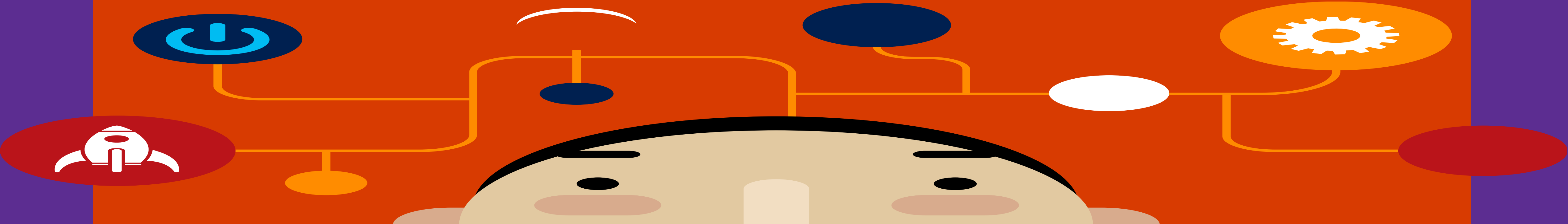
Introduce Coding Using Microsoft TouchDevelop & Kodu Game Labs in Your Library

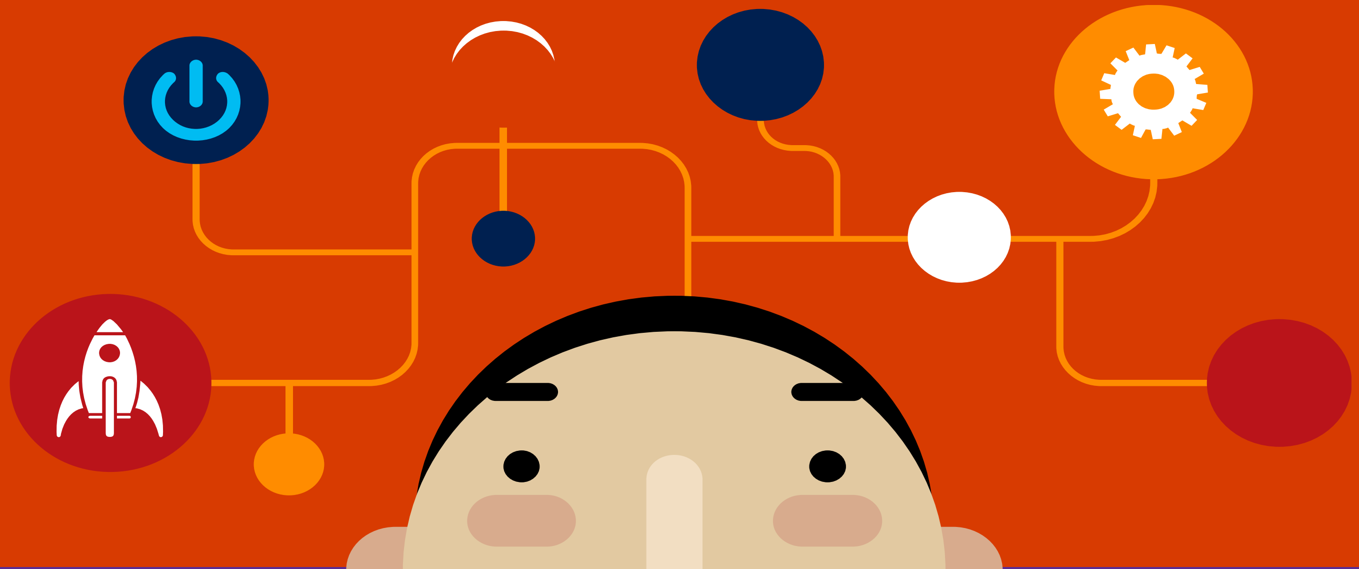
Angela Carlson, Microsoft
Elizabeth Iaukea, Washington State Library

Washington Library Employees 2016 Conference, Chelan, WA, USA, Earth



Experience with Coding Prepares Students for the 21st Century Workplace

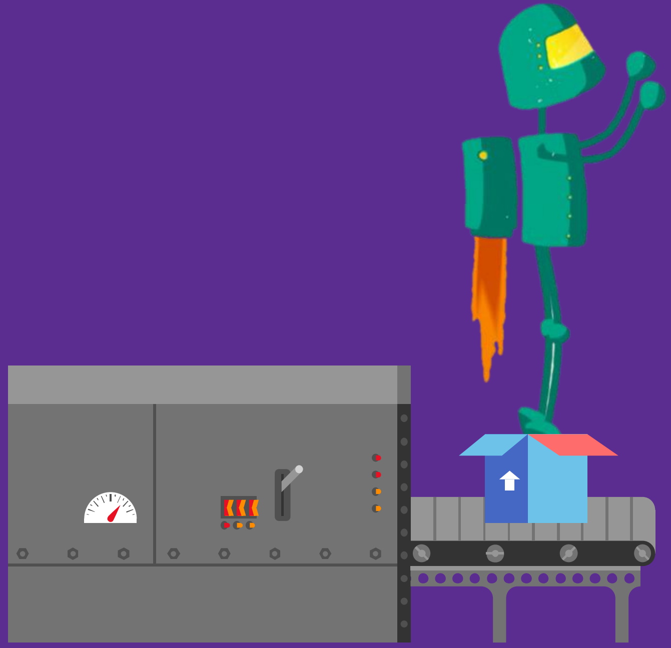
- problem solving
- critical thinking
- data analysis
- computational thinking

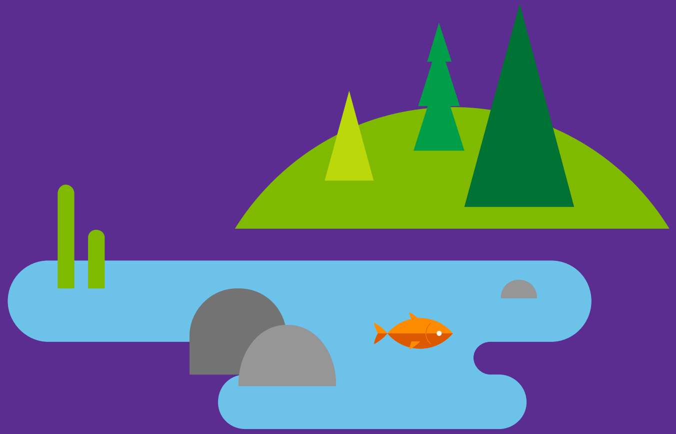


Hour of Code™

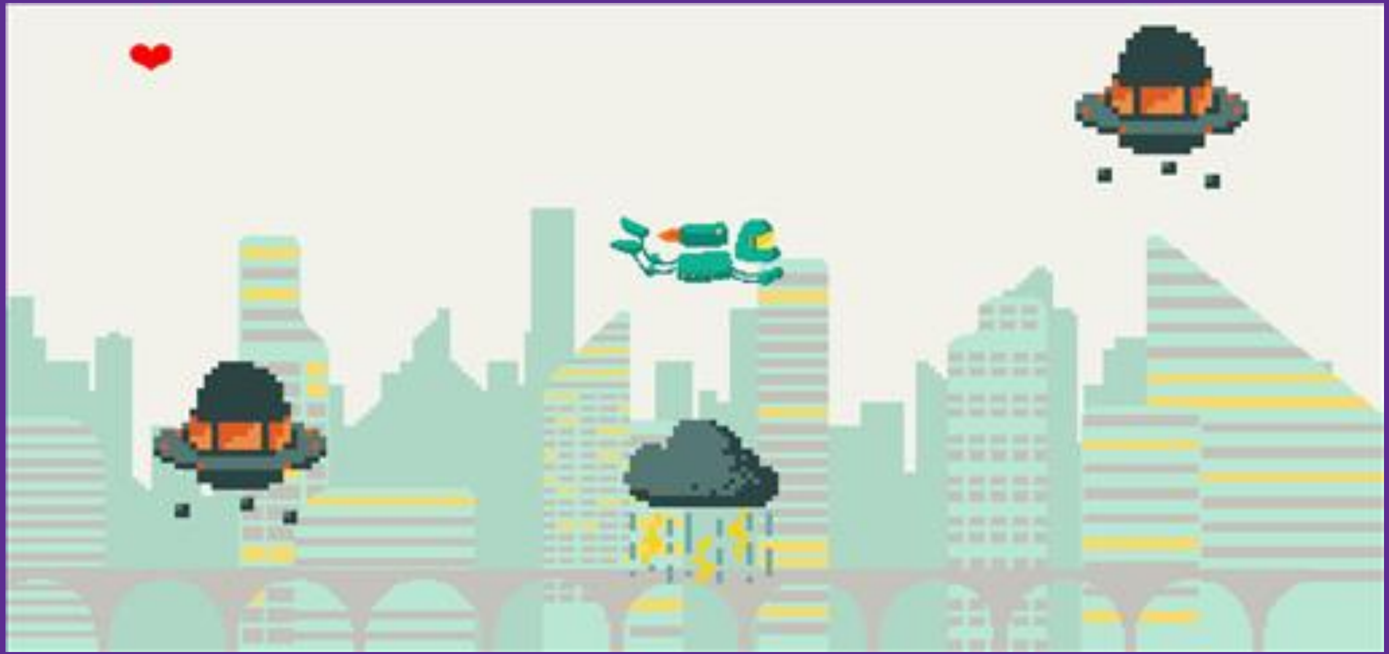
TouchDevelop

Minecraft



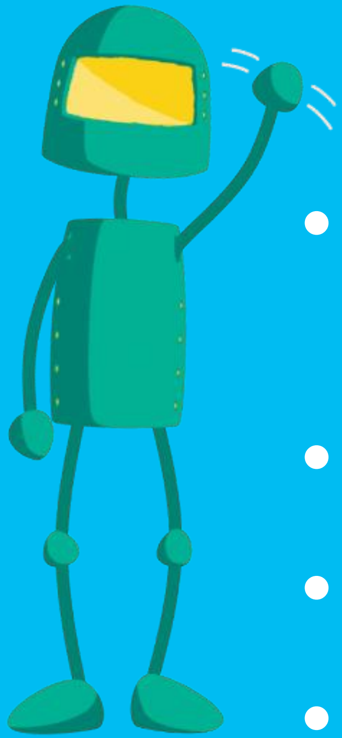


Coding Jet Pack Jumper



- Fix a game through a simple set of challenges so a crazy robot can make it through a maze of wacky obstacles
- Ideal for beginners ages 8+
- Duration: Up to one hour

Kodu Makerspace & Learn to Code Flatverse Curriculum for Camps & Clubs



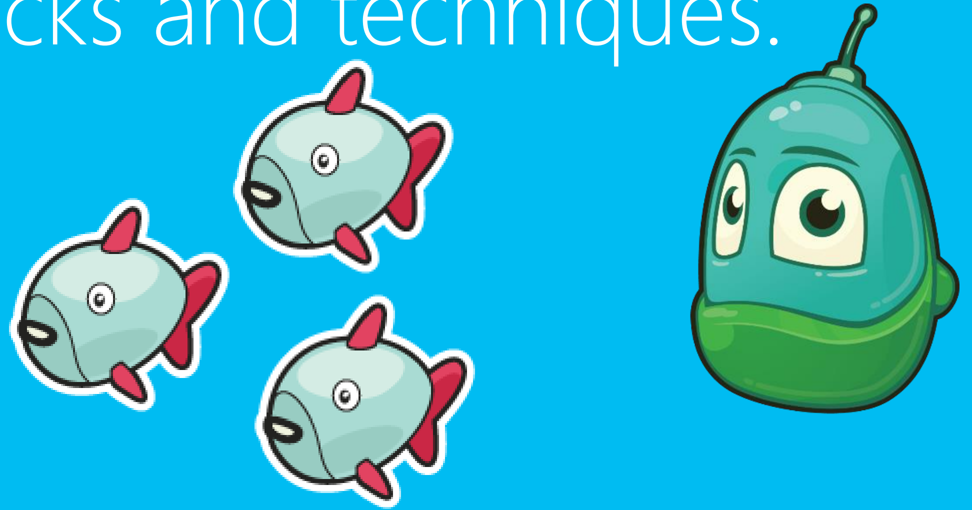
- Spiraling sequence of introductory computer science exploration
- No coding experience required to teach or learn
- Each camp is divided into 4, 2 hour sessions
- Kodu: Beginner coding, best for ages 8+
- Flatverse: Intermediate coding, best for ages 12+

Kodu Makerspace

- Introduce Kodu Game Lab and demonstrate how anyone can create rich and exciting games.
- Explore the design process and cover skills needed to create worlds and games with Kodu Game Lab.
- Learn how interesting and powerful games can be created with simple building blocks and techniques.

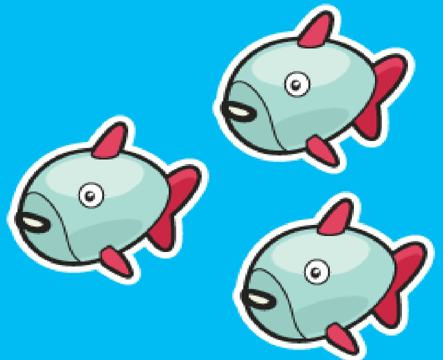
Kodu Makerspace – Camp Overview

- Introduce Kodu Game Lab and demonstrate how anyone can create rich and exciting games.
- explore the design process and cover skills needed to create worlds and games with Kodu Game Lab.
- Learn how interesting and powerful games can be created with simple building blocks and techniques.



Kodu Makerspace – Camp Format

- Some activities are interactive tutorials and others camper-driven, with open-ended game design.
- Each session includes collaborative design and development activities.
- Specific skills covered: analyzing and revising game characters, writing code to create in-game actions, collaboratively planning and creating a Kodu game, listening to peer feedback, and iterating the design process.



Learn to Code Flatverse Camp Overview

- Intro to basic computer science (programming/coding) concepts: screen coordinates, random numbers, objects and functions, variables and parameters, user input, loops, conditionals
- Use TouchDevelop - an interactive, online programming environment - to create and publish your own video game

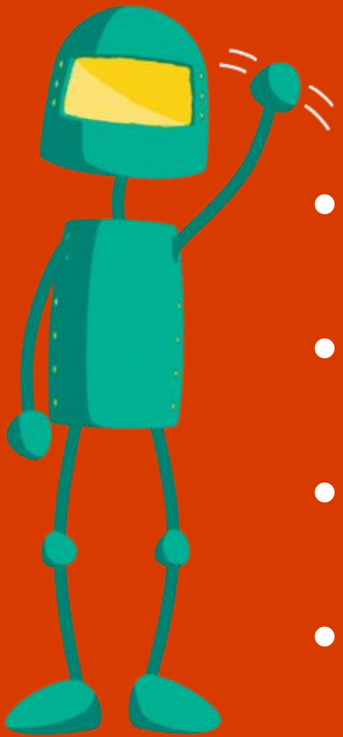
Flatverse Camp overview

All 4 Sessions have similar format:

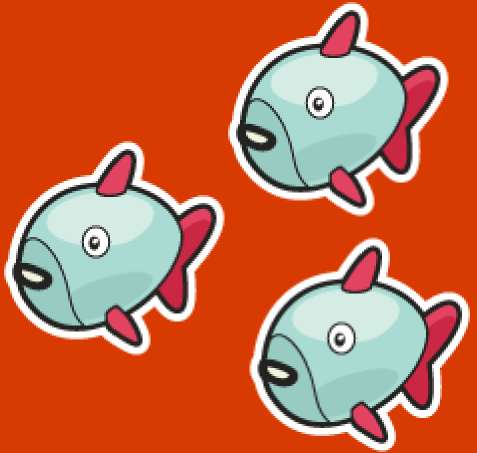
- Touch Develop functionality/1-2 computer science concepts
- Work with CS concepts in a standalone mini script
- Apply concepts into your Flatverse game

Final Session concludes with students sharing the game they've built

Flatverse & Kodu Camp Resources



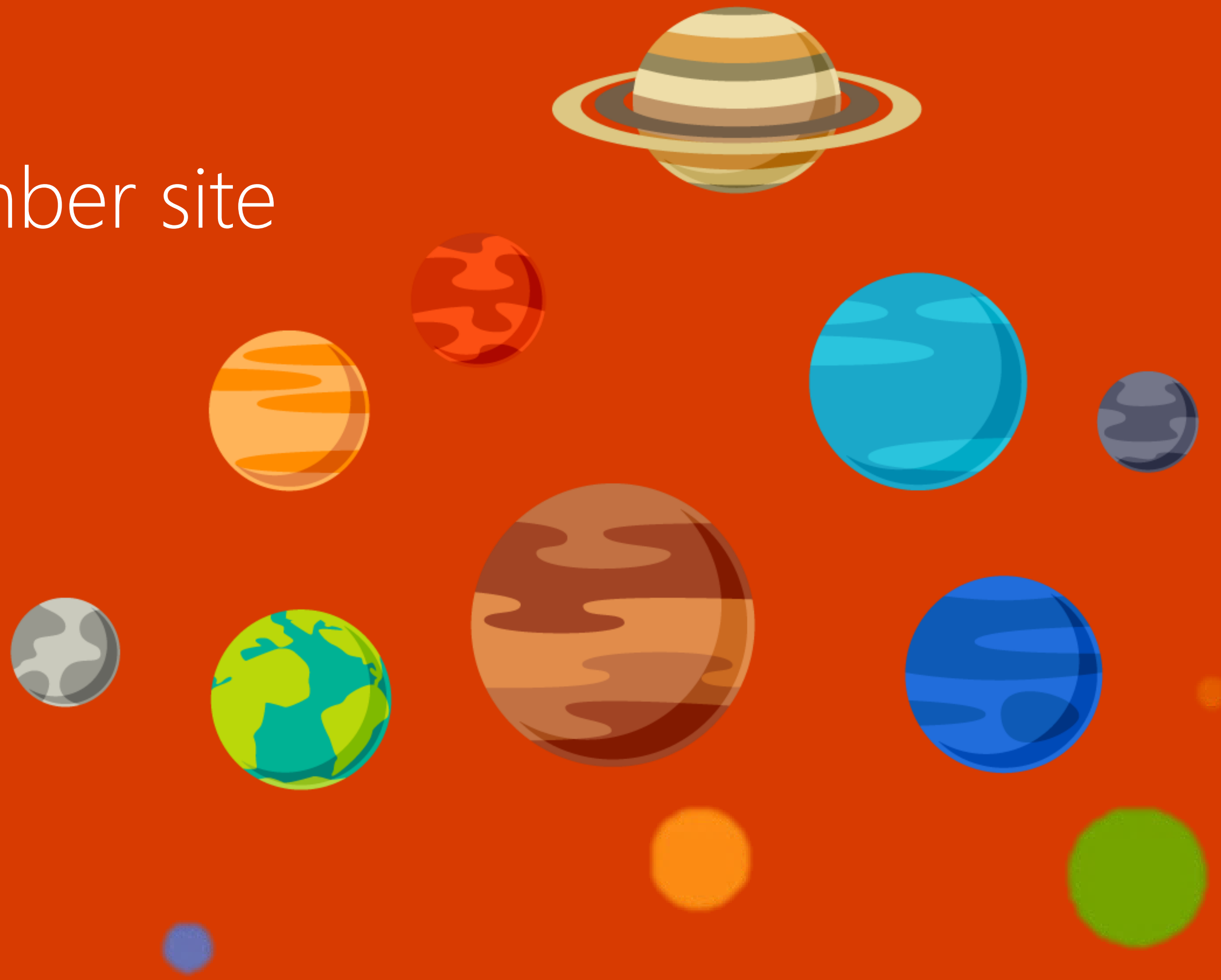
- Course Overview Document
- Session Leader Guides – one for each session
- PowerPoint slide deck for each session
- Web links and Tutorial scripts resource
- Student Workbook



Finding Camp Resources

Microsoft Imagine Academy Member site

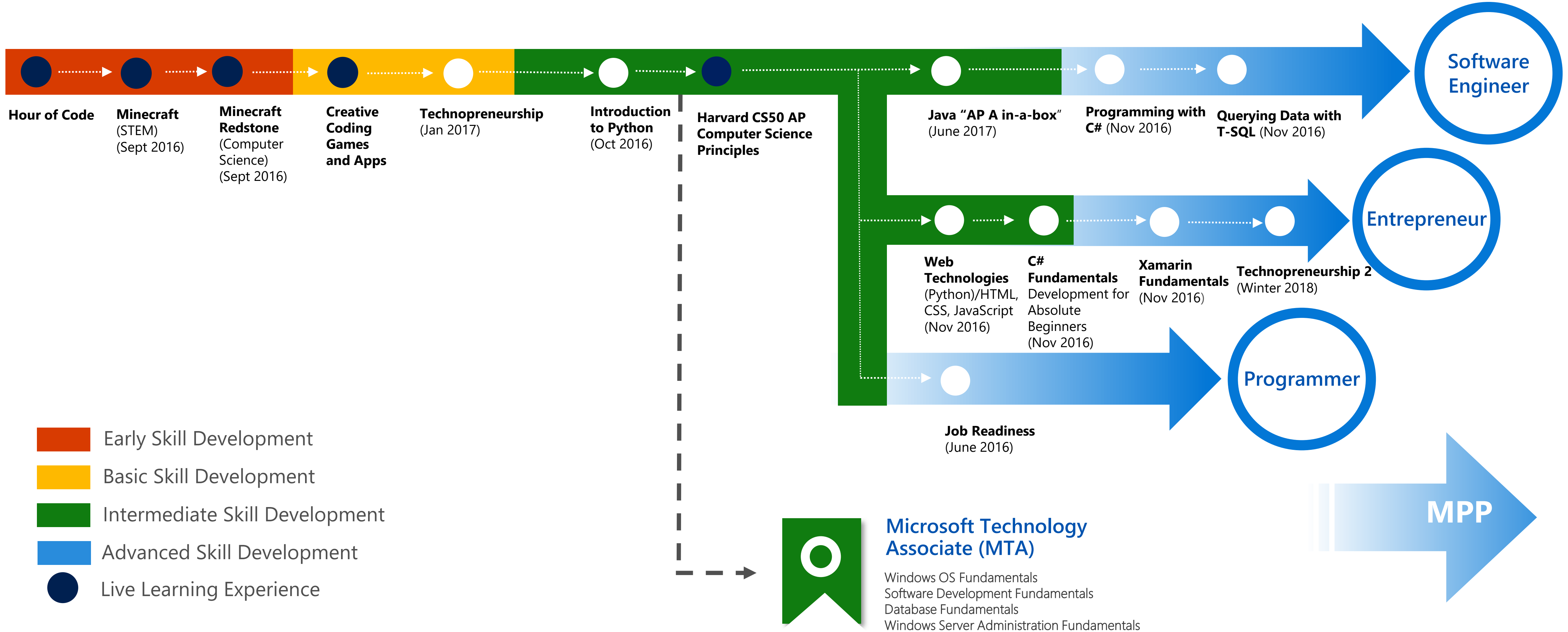
Microsoft Virtual Academy



Flatverse & Kodu Camp Leader Prep

- Review overview document
- Watch TouchDevelop/Kodu Makerspace videos
- Review the Leader Guide for each session
- Review tutorials for each session
- Gather materials
- Print Student Workbooks

Computer Science Pathway with MTA



Hour of Code

Minecraft (STEM) (Sept 2016)

Minecraft Redstone (Computer Science) (Sept 2016)

Creative Coding Games and Apps

Technopreneurship (Jan 2017)

Introduction to Python (Oct 2016)

Harvard CS50 AP Computer Science Principles

Java "AP A in-a-box" (June 2017)

Programming with C# (Nov 2016)

Querying Data with T-SQL (Nov 2016)

Software Engineer

Web Technologies (Python)/HTML, CSS, JavaScript (Nov 2016)

C# Fundamentals Development for Absolute Beginners (Nov 2016)

Xamarin Fundamentals (Nov 2016)

Technopreneurship 2 (Winter 2018)

Entrepreneur

Job Readiness (June 2016)

Programmer

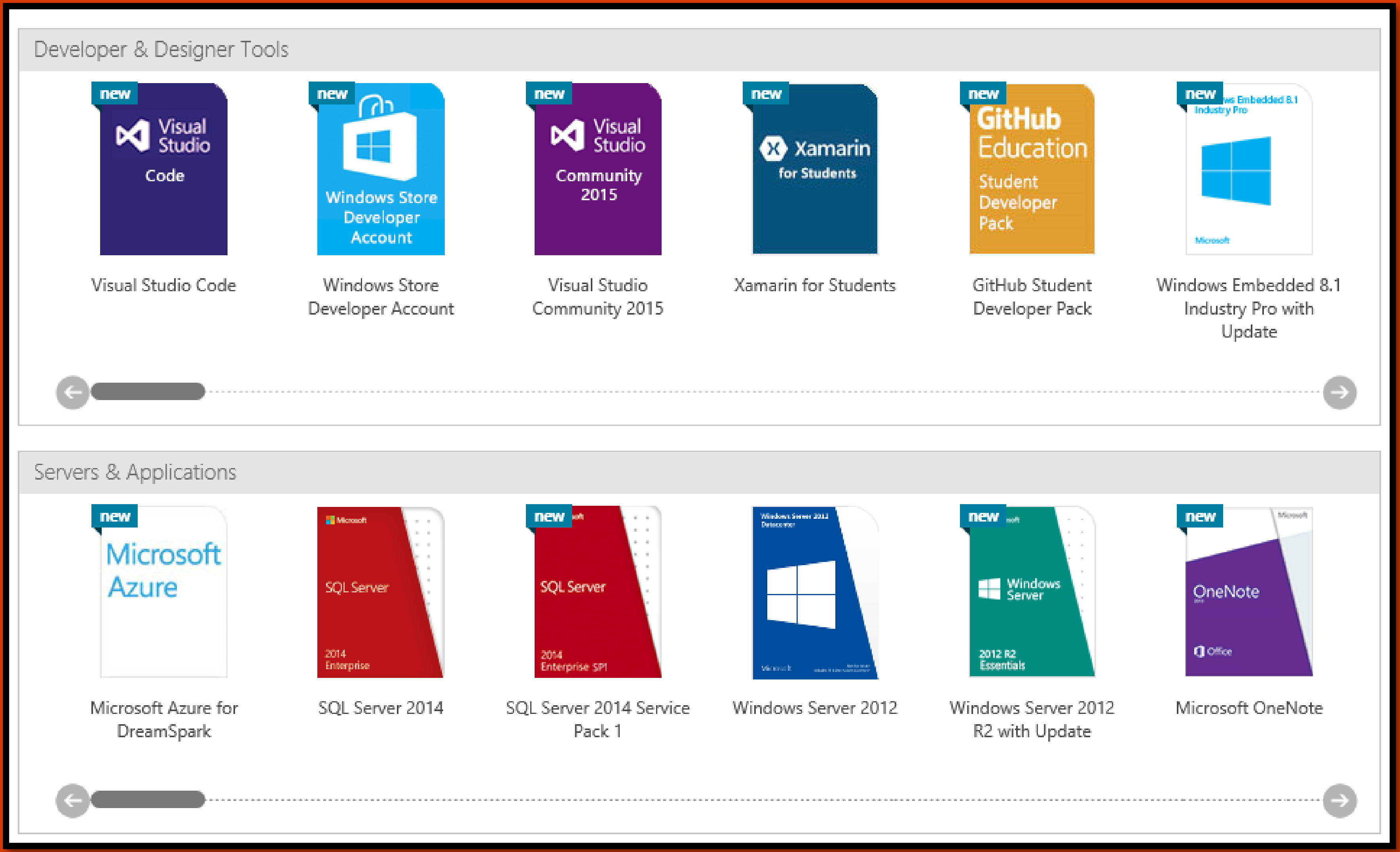
MPP

- 🟥 Early Skill Development
- 🟨 Basic Skill Development
- 🟩 Intermediate Skill Development
- 🟦 Advanced Skill Development
- ● Live Learning Experience

Microsoft Technology Associate (MTA)

Windows OS Fundamentals
Software Development Fundamentals
Database Fundamentals
Windows Server Administration Fundamentals
Networking Fundamentals
Security Fundamentals
Mobility and Devices Fundamentals
Cloud Fundamentals

DreamSpark: Level up Your Coding & Software Development Programming



new
Visual Studio Code
new
Windows Store Developer Account
new
Visual Studio Community 2015
new
Xamarin for Students
new
GitHub Education Student Developer Pack
new
Windows Embedded 8.1 Industry Pro
new
Microsoft Azure
SQL Server 2014 Enterprise
new
SQL Server 2014 Enterprise SP1
Windows Server 2012 Datacenter
new
Windows Server 2012 R2 Essentials
new
OneNote
Office

Developer & Designer Tools

- Visual Studio Code
- Windows Store Developer Account
- Visual Studio Community 2015
- Xamarin for Students
- GitHub Student Developer Pack
- Windows Embedded 8.1 Industry Pro with Update

Servers & Applications

- Microsoft Azure for DreamSpark
- SQL Server 2014
- SQL Server 2014 Service Pack 1
- Windows Server 2012
- Windows Server 2012 R2 with Update
- Microsoft OneNote

Creative Coding through Games and Apps (CCGA)

What is it?

- Semester-long interactive course using TouchDevelop
- Taught in 6, 9, 12 or 18 weeks
- Maps to CSTA standards
- Teaches fundamental programming concepts, real-world problem solving and product design skills
- Combines online and in-class resources for learning flexibility

Who is it for?

- Learners of any age with no previous programming experience
- Optimal for lower secondary grades
- Teachers with little or no computer science background
- Teachers looking to engage and excite students with hands-on game and app development

Harvard’s CS50 Computer Science Principles

Harvard course focusing on fundamentals of computing, problem solving, data, the Internet, cybersecurity, and programming. For use with or without AP designation/exam.

Start with…
Student: motivated with strong study skills (Algebra I recommended)
Teacher: previous experience teaching computer science and/or programming

End with….
Students learn to think algorithmically and solve problems that prepare them in each of the CS Principles Framework Learning Objectives.
Hands-on experience with multiple programming languages inspires confidence and the desire to continue their computer science journey

GET STARTED: Computer science basics introduced bit-by-bit by fixing fun apps and games

SELF-STUDY: content designed to be consumed directly by learners			
Name	Level	Length	Description
Hour of Code using TouchDevelop (aka “Jetpack Jumper”)	Beginner	1-3 hours	Fix a game through a simple set of challenges so a crazy robot can make it through a maze of wacky obstacles.
Birth of Bot	Intermediate	8 hours	Build from scratch a game that controls a crazy robot trying to make it through a maze of wacky obstacles.
Bot Levels Up	Intermediate	3 hours	Learn to make gameplay more challenging, add graphic effects and animations, use variables and cloud data, provide player instructions, and publish and share a game.
Hour of Code Event Readiness Center	Beginner	1 hour	Facilitator training and event materials to prepare and lead a group of students through an Hour of Code™.
INSTRUCTOR-LED: content designed for use in schools, camps and clubs			
Jetpack Jumper for Classroom, Camps and Clubs	Beginner	6-8 hours	Extend students’ computer science learning from Jetpack Jumper in your classroom, camp or club with presentation materials and all necessary student learning resources.
Dream it, Make it for Classroom, Camps and Clubs	Beginner	4-8 hours	Extend your students’ Hour of Code by adding real-world skills that developers use via your classroom, camp or club.

BUILD SKILLS: Hands-on learning covering computer science fundamentals

SELF-STUDY: content designed to be consumed directly by learners			
Name	Level	Length	Description
Introduction to Programming with Python	Intermediate	8 hours	Learn universally-applicable programming fundamentals by solving authentic problems using Python and Visual Studio.
Creating Games with Project Spark	Intermediate	5 hours	Learn to play your first adventure with Crossroads, control character behavior with Kode, and publishing your worlds.
Creative Coding through Games and Apps Implementation Training	Beginner	10-12 hours	On-line training to help teachers prepare to teach Creative Coding through Games and Apps.
INSTRUCTOR-LED: content designed for use in schools, camps and clubs			
Creative Coding through Games and Apps	Beginner	90 hours	First-semester course for introduction to programming designed for the early secondary grades.
Creative Coding Academy	Beginner	20 Hours	The club and camp version of the popular semester-length Creative Coding through Games and Apps course.

PURSUE A PATH: Full courses covering computer science basics in depth for college & careers

SELF-STUDY or INSTRUCTOR-LED: flexible content to be consumed directly by learners or in an instructor-led setting

Name	Level	Length	Description
Software Development Fundamentals	Advanced	8 hours	Learn key fundamental concepts of software development programs that all software developers need to know.
HTML5 App Dev Fundamentals	Advanced	8 hours	Explore introductory concepts for Hyper-Text Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript.
INSTRUCTOR-LED: content designed for use in schools, camps and clubs			
Java Programming Fundamentals	Advanced	6-12 hours	Introduce or review Java programming topics to prepare for the AP CS exam by having students write Java code to complete Office Mix lessons with interactive Code Hunt challenges.
CS 50 for AP Computer Science Principles	Advanced	180 hours	AP-level course designed not just to teach how to program but also how to think more methodically and how to solve problems more effectively.

Questions?