



THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

DEPARTMENT OF
COMPUTER SCIENCE
Au Reikura



2025

Contributors

J. Turner

J. Kasmara

© 2025 University of Waikato. All rights reserved. No part of this book may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without prior consent of the Department of Computer Science, University of Waikato.

The course material may be used only for the University's educational purposes. It includes extracts of copyright works copied under copyright licences. You may not copy or distribute any part of this material to any other person, and may print from it only for your own use. You may not make a further copy for any other purpose. Failure to comply with the terms of this warning may expose you to legal action for copyright infringement and/or disciplinary action by the University.

LEARNING PYTHON WITH BUDDY	4
High-risk AI	7
Human Oversight	7
Classification	9
Prohibited AI Practices	10
Programming Buddy to learn maths	13
Summary	17
Useful Resources	18

LEARNING PYTHON WITH BUDDY

Welcome to the seventh session of the Artificial Intelligence Series. In today's session, we will introduce you to learning Python with Buddy.

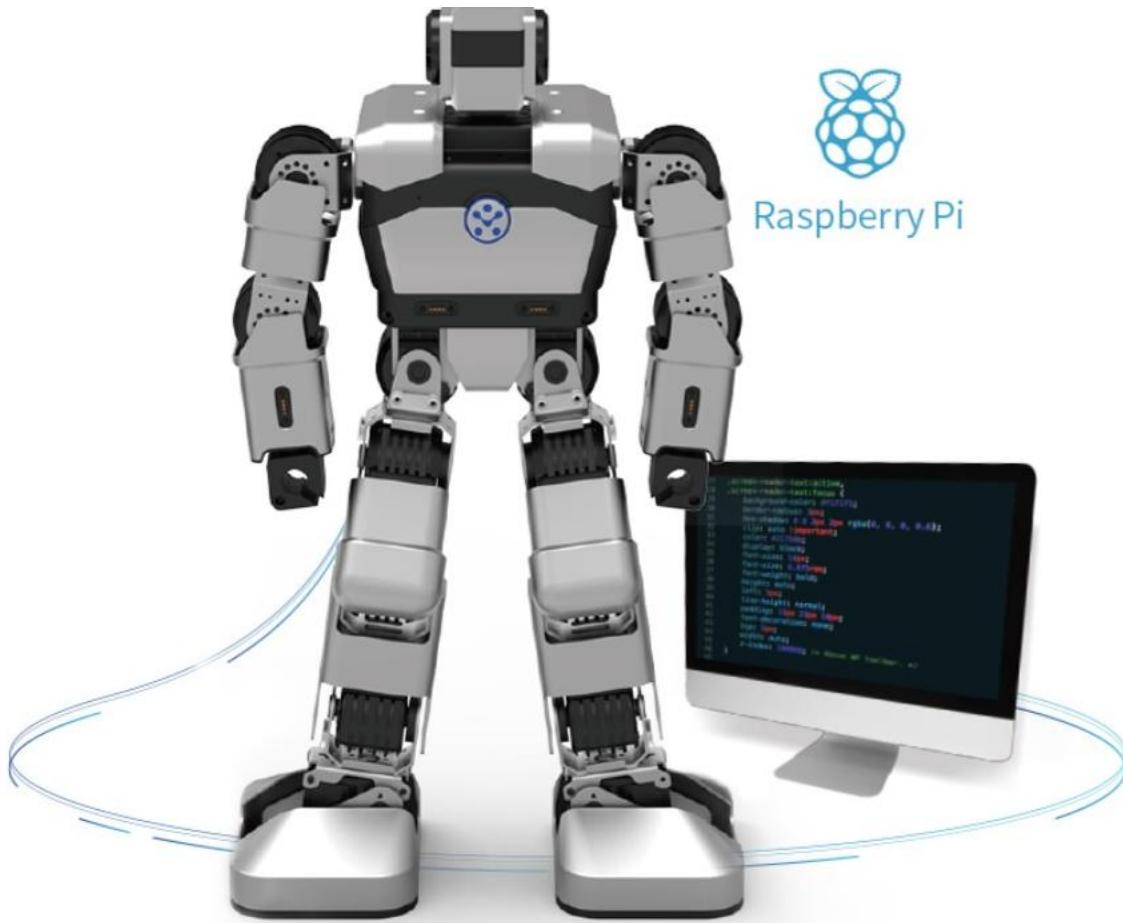


Figure 1: Our humanoid robot, Buddy

In the last session, we introduced you to a web-based interface, Jupyter Lab. This tool will become increasingly important in your third year of university when you work on more exciting projects that you can truly call your own. To quickly recap, in the first session, we trained Buddy to learn how to detect objects based on the objects that Buddy already trained to recognise. If there's one key takeaway so far about artificial intelligence, it's that AI programs run well on Python.

Thus, the three fundamental aspects that set Python apart from other programming languages are:

- **Indentation matters** – Stay consistent, or Python won’t play nice.
- **Comment your way** – Use one of Python’s 3 simple ways to add comments.
- **Variables on the fly** – Just assign a value, and boom, it’s created (perfect for rapid prototyping).

Indentation Example

```
01 if 5 > 2:  
02     print("Five is greater than two!")  
03 if 5 > 2:  
04     print("Five is greater than two!")
```

Line 2 shows a one-space indentation to separate its logic from Line 1. However, in Lines 3 and 4, there is no separation between the two statements. In Python, missing indentation is a significant issue because your program won’t run—the interpreter won’t understand the instructions. As a result, you will be presented with an error.

There are three ways to add comments in Python. As shown on the screen, the first method is a one-liner, where your comment is directly after your line of code on the same line. The second method is preferred by some people who like to write their comment first, followed by the code on a new line. Finally, the third method is for multi-line comments, where your comment takes up more than one line. All three methods are perfectly acceptable in Python.

In Python, the process has been streamlined so that the need to write extra lines for declaring a variable has been eliminated.

You just need to assign a value, and boom, the variable is created!

Variables on the fly Example

```
01 x = 5  
02 y = 7  
03 z = x * y
```

No wonder, it’s no surprise that Python is so popular for rapid prototyping, as variables are created the moment you assign a value to them. For example, in Line 1, the variable x is assigned the value of 5. In Line 2, the value 7 is assigned to y. This simplicity makes solving mathematical problems a breeze by asking the computer to compute it for us. In this simple example, Line 3 shows how we assign the result of multiplying x and y to a new variable, z.



However, no matter how advanced computers are becoming at an exponential rate, let's not forget that they are man-made creations. This gives us an ethical obligation to understand and verify the answers ourselves first. Let's give that a try!

What value does z hold?

Ans: If you said that z holds the value of 35, you're absolutely correct! Exercises like this act as a safeguard to ensure that our robots and programs behave as they're supposed to.



High-risk AI

As we continue exploring artificial intelligence, it's crucial to recognise that some AI systems pose higher risks than others. These "high-risk AI" systems have the potential to impact health, safety, and fundamental rights. For example, AI used in medical diagnoses, autonomous vehicles, and recruitment processes must be carefully designed to avoid harm or unfair outcomes.

The EU Artificial Intelligence Act defines strict rules for these systems, ensuring that human oversight is always present. This means AI should not make critical decisions without human intervention. Developers must build safeguards to prevent errors and ensure fairness. For instance, if a high-risk AI system in healthcare predicts an incorrect diagnosis, a doctor must be able to review and correct it.



Figure 2: AI brings precision and insights to computer-assisted surgery.

As we program Buddy, it's important to apply this principle of human oversight. We must ensure that AI systems assist us rather than fully replace human decision-making. This understanding helps us create responsible AI that benefits society while minimising risks.

Human Oversight

As we continue working with Buddy, it's important to remember that AI systems are tools designed to assist humans, not replace them. While AI can analyse data, recognise patterns, and even make predictions, it does not have human judgment, emotions, or ethical reasoning. This is where human oversight comes in—ensuring that AI is used safely, fairly, and responsibly.

The EU Artificial Intelligence Act (Article 14) emphasises that high-risk AI systems must include features that allow humans to oversee and intervene when necessary. This means AI

should not operate without human supervision, especially in areas where mistakes can lead to harm. For example, if an AI system in a self-driving car detects a pedestrian incorrectly, a human should be able to override the AI to prevent an accident.



Figure 3: overriding AI system in a self-driving car

In our case, when programming Buddy, we must apply the same principle. We need to regularly check Buddy's responses, ensure the AI is making correct decisions, and be ready to step in if something goes wrong. This also means designing AI systems that explain their decisions in a way that humans can understand—an essential part of ensuring trust and accountability.

Trust based on Accountability..



Figure 4: cultivating trust through accountability

By practicing human oversight in AI development, we become responsible programmers, ensuring that AI remains a tool for empowerment rather than an uncontrolled force. Keep this in mind as you continue coding and testing Buddy's capabilities!

Classification

In artificial intelligence, classification is the process of teaching AI to recognise patterns and sort information into categories. AI systems use classification in many ways, from identifying spam emails to recognising faces in photos.

The EU Artificial Intelligence Act (Annex III) highlights that certain AI systems used for classification can pose risks, especially in critical areas like employment, education, and law enforcement, <https://artificialintelligenceact.eu/annex/3/>. For example, AI used to sort job applications or assess students' performance must be carefully designed to avoid unfair bias. If the AI is not properly trained, it might favour one group over another without anyone realising it. This is why transparency and accuracy are crucial in classification systems.

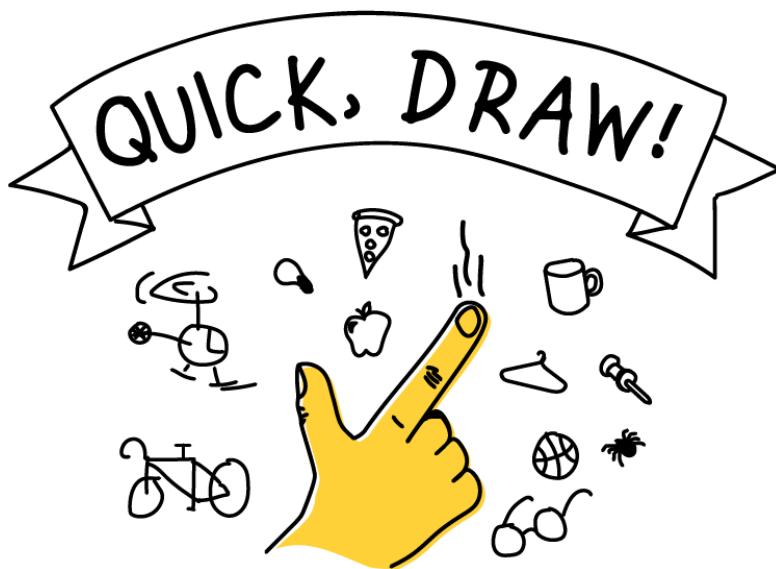


Figure 5: Google Quick, Draw!

When programming Buddy, we must be mindful of how AI classifies objects and people. Buddy should learn from accurate and balanced data to avoid incorrect assumptions. By testing and refining Buddy's classification skills, we ensure that the AI makes reliable and fair decisions. As future AI developers, it's our responsibility to understand classification, apply it ethically, and ensure that AI systems serve everyone fairly and accurately! Wait until Session 8, where you'll

have the chance to help Buddy make his AI systems more reliable and fairer by playing around with Quick, Draw!

Prohibited AI Practices

As we explore AI and its capabilities, it's important to understand that not all AI applications are allowed. Some AI systems can be harmful, unethical, or even dangerous, which is why certain uses of AI are strictly prohibited under the EU Artificial Intelligence Act (Article 5).

AI practices that manipulate people in harmful ways or exploit their vulnerabilities—such as systems that trick users into making decisions they wouldn't normally make—are banned. For example, an AI system designed to control human behaviour without consent, like deceptive advertising that pressures users into buying unnecessary products, is not allowed.



Figure 6: Buying pressures

Another prohibited AI use is social scoring, where people are judged and ranked based on their behaviour, personal characteristics, or beliefs. This could lead to discrimination and unfair treatment, affecting access to jobs, loans, or public services.

When programming Buddy, we must ensure that AI is used ethically. AI should empower people, not control or judge them unfairly. By following these guidelines, we help create AI systems that are responsible, fair, and beneficial to society!

Summary

- **High-risk AI:** Some AI systems, like those in healthcare and self-driving cars, require careful design to prevent harm and ensure fairness.
- **Human Oversight:** AI should remain a tool for empowerment, with humans overseeing and refining its decisions to ensure responsible use.
- **Classification:** AI must classify objects and people accurately and fairly, using balanced data to avoid bias and incorrect assumptions.
- **Prohibited AI Practices:** Some AI uses, like manipulation or social scoring, are banned to protect fairness, privacy, and human rights.



Exercises:

- Which article from the EU Artificial Intelligence Act talks about ‘Prohibited AI Practices? **ANS:** Article 5



- List out some of the AI systems that falls under High-risk AI?

ANS: emotion recognition, as safety components in the management and operation of critical digital infrastructure, road traffic, or in the supply of water, gas, heating or electricity.

Have a try to answer the exercises by referring to the website, <https://artificialintelligenceact.eu/>.

Programming Buddy to Learn Maths

Now that you understand some legislation around High-risk AI so we can program AI applications responsibly, let's introduce **Blockly**, which is an open-source software to visually program Buddy by drag-and-drop blocks. Before we make Buddy come to life with Blockly, make sure you follow the “Programming Buddy Method 2” section in the Configuring Buddy.pdf guide.

On the Yanshee operating system, you should see the **Blockly** Desktop icon (Figure 7).

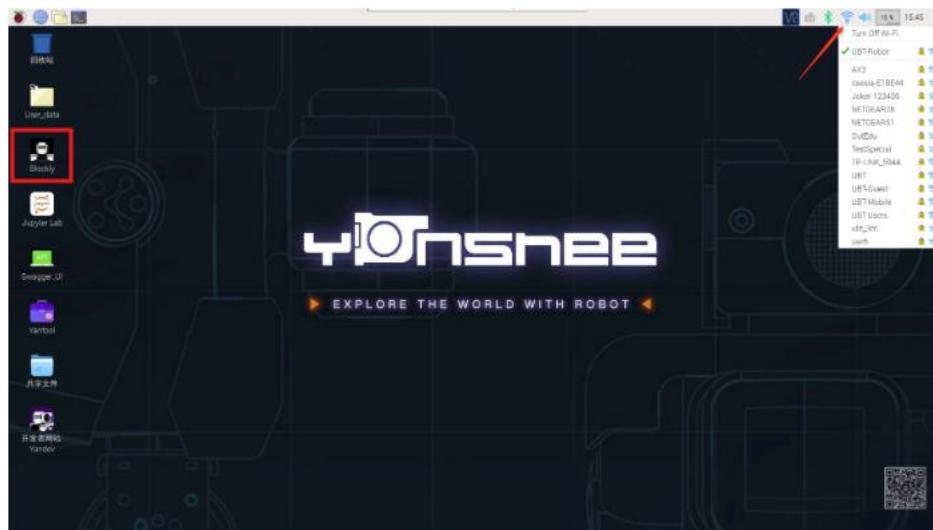


Figure 7: Yanshee operating system

Double-click the icon to open **Blockly** software and once opened, your screen should look like what is being shown in Figure 8.

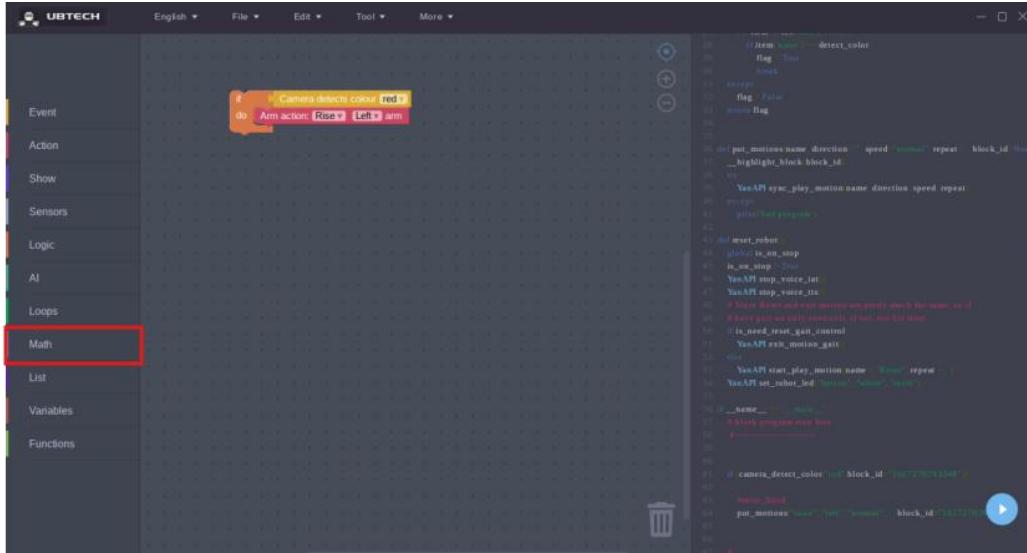


Figure 8: Blockly Interface

Create a new file called `basic_maths.ipynb` and save it your folder. This is where we will write our code in Python for today's session. Feel free to ask staff for assistance if you get stuck.



Now we can start writing code to program Buddy to have the skillset in mathematics. Buddy is ready to learn new things from us and to show off his new skills.

Firstly, we must include this code:

```
01  #block program start here
02  #-----
03  x = 3 * 5
04  y = 64 % 64
05  z = 2 - 1
06
07
```

Line 1 is commented to show where the location of your block programming will start.

Line 3 defines a variable, x. Variable, x, is being assigned a value of three times five which equates to fifteen.

Line 4 is defining the variable, y and at the same time, assigning a value of the remainder of 64 being divided by 64.

Line 5 defines a variable, z. Where variable z is being assigned a value of the subtraction of 1, being subtracted from two.

Quiz

What is value does the variable, y holds?

Ans: 1



Secondly, we can try assigning a value to variable other than x, or y, or z:

```
07  set apple to 1 + 1
08  set BowlingBall to 33
09
10  If BowlingBall > apple
11  do:
12      else:
```

Line 7 defines a variable, apple and apple is being assigned a value of the addition of two 1's.

Line 8 is defining a variable, BowlingBall to a value of 33.

Line 10 states the conditional statement how the two variables related to each other

Show your Blockly to the student ambassador and answer this last quiz before we get to **Today's Exercises**.

Quiz

What is a likely scenario where apple is heavier than BowlingBall ?

Answer: when apple is assigned a value bigger than BowlingBall



Today's Exercises:

1. See if you can put another conditional statement after the variables x,y,z
2. Select the if and do logic block for Buddy and make Buddy do an action
3. Now select the if, do and else logic block for the variable apple and BowlingBall
4. Show your complete coding for Buddy by trying the Show module and show this to the student ambassadors.
5. See if you can do a mathematical formula using Blockly and once completed show this to the student ambassador.

Note: Use Yanshee mobile app to try this advanced exercise: Buddy's Talent Show!

Advanced Exercise: Buddy's Talent Show! 🎤🎭

Scenario:

Buddy has been invited to a robotics talent show! The audience is waiting for Buddy to perform, and he must showcase two of his best talents: dancing and speaking. Your challenge is to program Buddy to:

- Greet the audience in an engaging way.
- Perform a short dance routine.
- Announce the winner of the talent show with enthusiasm.

Think about how you can make Buddy's performance exciting! Use a combination of Blockly and Python programming to control Buddy's movements, speech, and visual effects. Experiment with conditional statements, loops, and timing to synchronize Buddy's actions.

Once you're ready, perform Buddy's talent show for the student ambassadors and see if they think Buddy deserves to win!

Summary

In this session, we focused on learning Python with Buddy and programming Buddy to learn maths, building essential AI and robotics skills. We explored Python fundamentals such as indentation, comments, and variables, understanding why Python is a preferred language for AI development. Using Blockly, we programmed Buddy to perform mathematical calculations, apply logical conditions, and enhance its decision-making abilities. Alongside hands-on programming, we examined key AI concepts, including high-risk AI applications, the importance of human oversight, and how AI classifies data to make decisions. We also discussed ethical



considerations, such as prohibited AI practices, ensuring that AI systems like Buddy are developed fairly and responsibly. By combining coding skills with an awareness of AI risks and ethics, we are laying the groundwork for creating intelligent and ethical AI systems. Keep practicing and exploring new ways to enhance Buddy's capabilities!

Useful Resources

- € Yanshee manual: <https://yandev.ubtrobot.com/#/en>
- € Text-to-Speech Guide: <https://www.readingrockets.org/topics/assistive-technology/articles/text-speech-tts>
- € Python Quiz https://www.w3schools.com/python/python_quiz.asp
- € EU Artificial Intelligence Act <https://artificialintelligenceact.eu/>
- € Yanshee mobile app on App Store: <https://apps.apple.com/us/app/yanshee/id1290088340>
- € Yanshee mobile app on Google Play:
https://play.google.com/store/apps/details?id=com.ubtedu.alpha1x&hl=en_US

