Do you speak Python?

In this lesson, students are going to explore the role of the English language in programming languages. Based on their knowledge of English they write their own coding dictionary and use it to understand a basic program in Python. Duration: 1 hour.

STRUCTURE

- Introduction: Programming languages
- Next step: The role of the English language in programming languages
- In practice: Write your own coding dictionary and understand a basic program
- Evaluation: The importance of understanding code.

PREPARATION

You can prepare by:

- Reading this teacher's guide and the students' worksheet.
- Opening the presentation and reading through the slides.

PERSONALISED LEARNING

No programming knowledge is required for this lesson. You can challenge students who do have programming knowledge by giving them alternative assignments. For assignment 1 you can let them choose their own words for their dictionary instead of the prewritten words of the assignment. For assignment 2 you can let them write their own basic program instead of translating the given program.

WORK AND CAREER

Computer programmers, software developers and coders use their knowledge of programming languages in order to write software, build websites, apps and more.

ETHICS AND TECHNOLOGY

Most programming languages are based on English, is this fair to non-English speakers? And is it necessary for yourself to be able to read code in order to be able to work with programs?

LEARNING TARGETS

Domein curriculum 2021	Leerdoelen digitale vaardigheden	Kerndoelen	21st century skills
1 De werking en het (creatieve) gebruik van digitale technologie DG3.2 Aansturen van en creatie met digitale technologie.	**1 Computational thinking** De leerling kent een aantal functies in minstens één programmeertaal en kan daar de meest effectieve oplossing uit kiezen.	**1 Engels** De leerling leert welke rol het Engels speelt in verschillende soorten internationale contacten.	**1 Communiceren**
2 Digitale communicatie & samenwerking DG4.2 Digitale communicatie.	**2 Computational thinking** De leerling kan een bestaande reeks instructies lezen en begrijpen.	**2 Engels** De leerling leert strategieën te gebruiken voor het uitbreiden van zijn Engelse woordenschat.	**2 Samenwerken**

INTRODUCTION

Opening slide

Slide 1, Group discussion

Ask your students whether they have ever programmed. What programming languages did they use? Do they think it's hard to learn a programming language? Why (not)?

NEXT STEP

Slide 2, Group discussion

Watch the following video together: https://www.youtube.com/watch?v=Y_9t3eQFmU4. As you can see in the video, there are many different programming languages. Most programming languages are based on English, why do you think that is?

Possible answers:

- English is the International language of business
- Programmers need to communicate about their code with people worldwide
- It makes it easier to share code with people worldwide.

IN PRACTICE

Slide 3, Group work

Explain: You are going to make your own coding dictionary for the programming language Python. It is okay if you have never programmed before. Just use your knowledge of English to translate the code. Work together with your classmate on assignment 1 on the worksheet.

Slide 4, Group discussion

Discuss the students' answers using this slide. You can use the following questions: What stands out about this programming language? What is typical about a programming language? What is the difference between English and a programming language? For example: Programming languages are short and concise with abbreviations. Punctuation marks like colons, semicolons and brackets have a special meaning in programming. For example, a semicolon means the end of a statement in Python.

Slide 5, Instruction

Now use your coding dictionary to read this code.

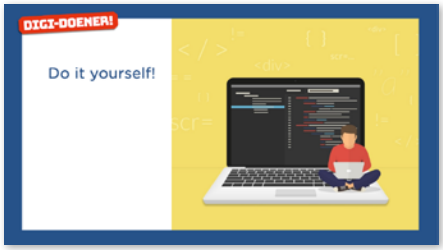
What do you think the program does?

Example of a correct answer:

The program generates a random number, it prints: "I'm thinking of a number between 1 and 10, can you guess what it is?". Then it prints "Have a guess". The user can enter a guess. While the guess is not equal to the random generated number, the program prints "Too high!" if the guess is too high and "Too low!" if the guess is too low. Then the user enters a guess again. Once the guess is equal to the random generated number, the program prints "Got it!".

Slide 6, Group work

Work together on assignment 2 on the worksheet.
Discuss the students' answers afterwards. Example of a correct answer:
This is a piece of code behind a game. The code checks whether the player can fire a bullet, if bullet state is ready, it fires a bullet from the place (coordinates) where the player is located. Then it calculates how the bullet moves. If the distance is under 15, there will be a collision. The part below '#Create Keyboard Bindings' allows the player to control his character, to move left (left arrow), right (right arrow) and to fire a bullet (space). The part below '#Main Game Loop' makes sure that the enemy moves around.



WORK AND CAREER

Slide 7, Group discussion

Computer programmers, software developers and coders use their knowledge of programming languages in order to write software, build websites, apps and more.

EVALUATION

Slide 8, Group discussion

Discuss with your students:

- Do you think it's hard or easy to learn a programming language?
- Does it help to be able to speak English when learning a programming language?
- Most programming languages are based on English, do you think this is fair to non-English speakers?
- Do you think that everybody should learn a programming language? Why (not)?
- Is it important for yourself to be able to read code in order to understand programs? Why (not)?