

Python for Kids: Algorithms, Pets and Turtle Drawing

FA-0805 · 1 hour

Software Development

Introduction

Discover step-by-step algorithms through a physical activity, then run and change small Python examples. A virtual pet introduces input, variables and decisions; a colourful Turtle square introduces repetition and drawing commands.

Learning outcomes

- Describe an algorithm as a sequence of instructions.
- Predict how a change in input affects a simple pet program.
- Recognise a variable, a decision and a repeated drawing action.

Prerequisites

Curiosity and willingness to explore. Use a prepared computer with Python 3 and working Tk/Turtle graphics; internet access is needed only for selected resources or remote participation. No previous coding is required.

Training outline

1. Algorithm obstacle course — 25 minutes

Follow a short sequence of steps through a simple supervised course using markers, cones or hoops. Compare a person following instructions with a computer executing an algorithm.

2. Python pet adventure — 15 minutes

Run a prepared program that asks for a pet name and food. It starts with hunger at 5, reduces it by 2 when carrot is selected, and prints the result. Change the input and discuss variables, input/output and if/else decisions.

3. Turtle drawing challenge — 20 minutes

Run a prepared Turtle program that repeats four forward moves and right-angle turns. Choose colours from a list to draw a colourful square, then experiment with colours and drawing parameters.

Coding examples to explore

Pet example: `pet_name = input("Give your pet a name: "); hunger = 5`. Use print with an f-string to greet the pet, then obtain food with input. The if `food == "carrot"` branch prints a response and runs `hunger -= 2`; else prints the alternative response. Display hunger after the decision.

Drawing example: `import turtle; create screen = turtle.Screen() and pen = turtle.Turtle()`. Set `colors = ["red", "blue", "green", "purple"]`. In for `i in range(4)`, use `pen.color(colors[i])`, `pen.forward(100)` and `pen.right(90)`. Finish with `turtle.done()`.