

## Personal Project

---

### Textbook

---

## Personal Project



Imagine you are a wildlife scientist or an environmental engineer! You need to design a safe and healthy habitat for an animal. How can you use computer coding to model what that habitat needs to stay balanced? This project will challenge you to **solve a real-life problem in science and engineering** using the computational thinking skills you've learned!

You will **create a digital model or simulation** of an animal habitat. Your program will show how an animal needs certain things (like food, water, and shelter) to survive and thrive.

Protecting animals and ensuring they have healthy places to live is a huge real-world challenge for scientists (like biologists who study animals) and engineers (who design wildlife crossings or preserve habitats). Your project is a simplified version of this important work!

- Consider the computational steps below as you plan your project.
  - **Decomposition:** What are the key parts of an animal's habitat? (e.g., the animal, food, water, shelter). What rules govern how the animal interacts with these things?
  - **Pattern Recognition:** Animals need food and water *repeatedly*. How can we show this pattern in our code?
  - **Abstraction:** How can we represent complex ideas like "hunger" or "thirst" with simple numbers or visual cues in our program?
- **Algorithms:** What are the exact step-by-step instructions your animal needs to follow to find food or shelter?

## Build Your Digital Animal Habitat!

Your program should simulate a simple habitat, showing how your chosen animal interacts with its environment to stay healthy.

**Here are some ideas for what your Digital Animal Habitat could include:**

- **Your Animal Sprite:** This is your main character!
- **Habitat Elements:** Sprites for food (e.g., berries, grass), water (e.g., pond, water dish), and shelter (e.g., cave, tree).
- **Animal Needs (Variables):** You could use variables (like a "hunger level" or "thirst level") that go up over time, and go down when the animal finds food or water.
- **Rules for Survival (Conditionals):**
  - **IF** hunger level is very high, **THEN** the animal looks for food.
  - **IF** the animal touches food, **THEN** hunger goes down, and the food disappears (or shrinks).
  - **IF** thirst level is very high, **THEN** the animal looks for water.
  - **IF** the animal touches water, **THEN** thirst goes down.
  - **IF** hunger or thirst gets too high for too long, **THEN** the animal might "get tired" or "go to sleep" (a simplified way to show it's not healthy).

Before building your project, **create a design document** that shows what your program will do. Choose one method:

- Write out the steps in plain language
- Use pseudocode
- Make a flowchart or labeled diagram

Once you've built your program, test it out. Get feedback from others and make revisions. Update and edit anything in your code to improve any inaccuracies!

## Blocks Pro Challenges (1)

### 1. Animal Habitat

Challenge

Textbook

Animal Habitat

Choose one animal sprite and create a habitat setting (background and other sprites for food, water, shelter).

**Use Variables:** Use at least two variables to track an animal's needs (e.g., "hunger," "thirst," "energy").

**Use Loops:** Your animal's needs should change over time (e.g., hunger goes up slowly in a **forever** loop).

**Use Conditionals (if/else):** Your animal must make decisions based on its needs (e.g., IF hungry THEN go to food).

**Use Operators:** You might use operators to compare needs (e.g., **`hunger > 80`**).

Requirements

0/7

2

Motion

2

Looks

1

Event

Code

Costumes

Sounds

Motion

move 10 steps

turn 15 degrees

turn 15 degrees

go to random position

go to x: 0 y: 0

glide 1 secs to random position

glide 1 secs to x: 0 y: 0

point in direction 90

point towards mouse-pointer

change x by 10

set x to 0

change y by 10

set y to 0

if on edge, bounce

set rotation style left-right

☐ x position

Sprite1

x 0 y 0

Size 100 Direction 90

Sprite1

Backdrops 1

Scratch

Scratch