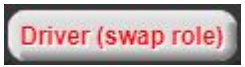
Pair Programming!

Teacher A: Navigate!
Teacher B: Drive!

1. **Teacher A** should see the 

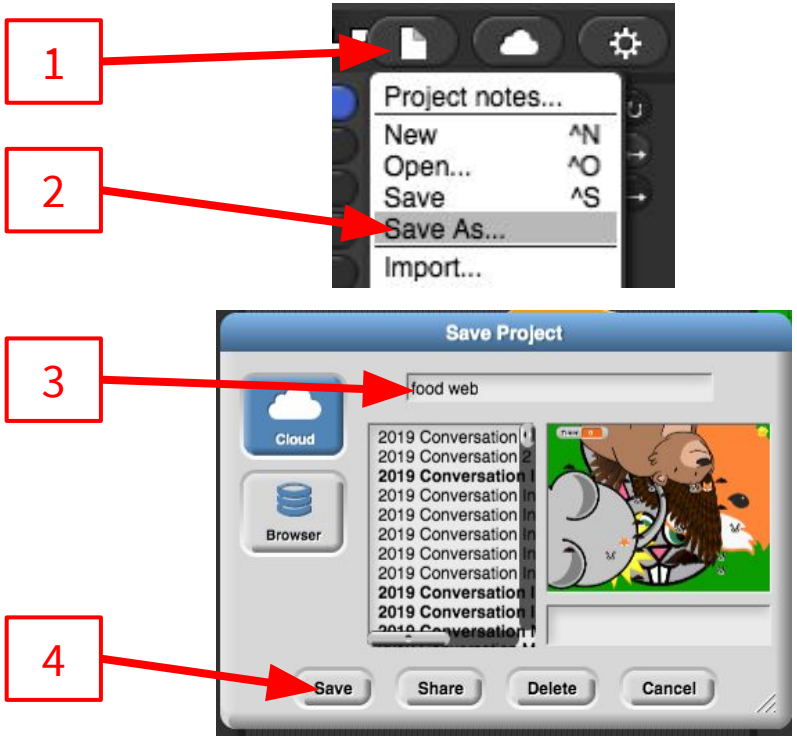
button
2. **Teacher B** should see the 

button
3. **Teacher B** should **share screen** and **perform all the coding actions**

Save Your Code

It's a good practice to Save your code every time you finish something!

1. Click the **File Icon**
2. Click "**Save as...**"
3. Type in "**Foodwebs**"
4. Click on the **Save** button

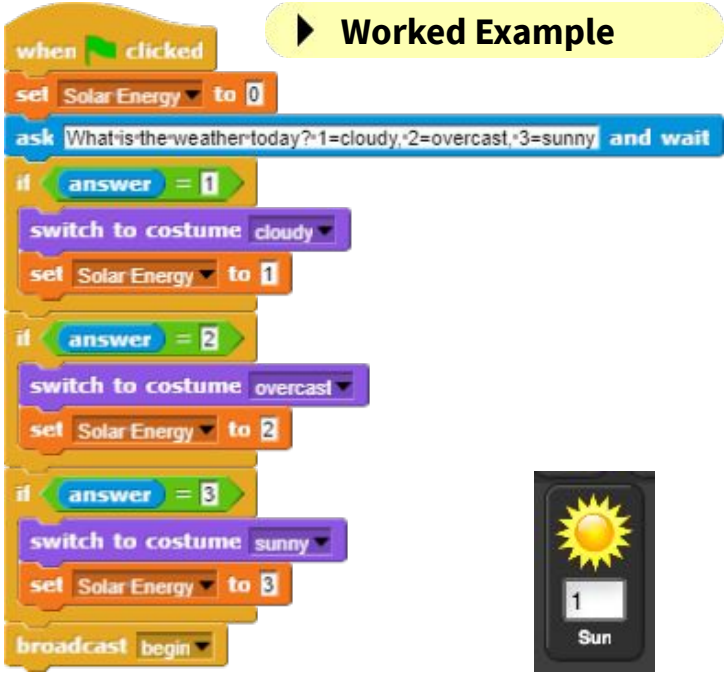


Step 1: Use the Sun!

Let's first look at the **Algorithm** for the **Sun.**

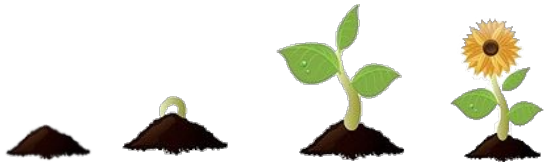
1. Click on the **Sun** sprite
2. Run the code using the Green Flag with **different inputs (1,2,3)**. What does the sun code do?
3. Label **each** if block with a **comment** saying what the if block do.
4. Put a **comment** on the side of the script describing what **Computational Thinking** element is in this script.

Worked Example



Save your code!

Step 2: Finish The Plants!

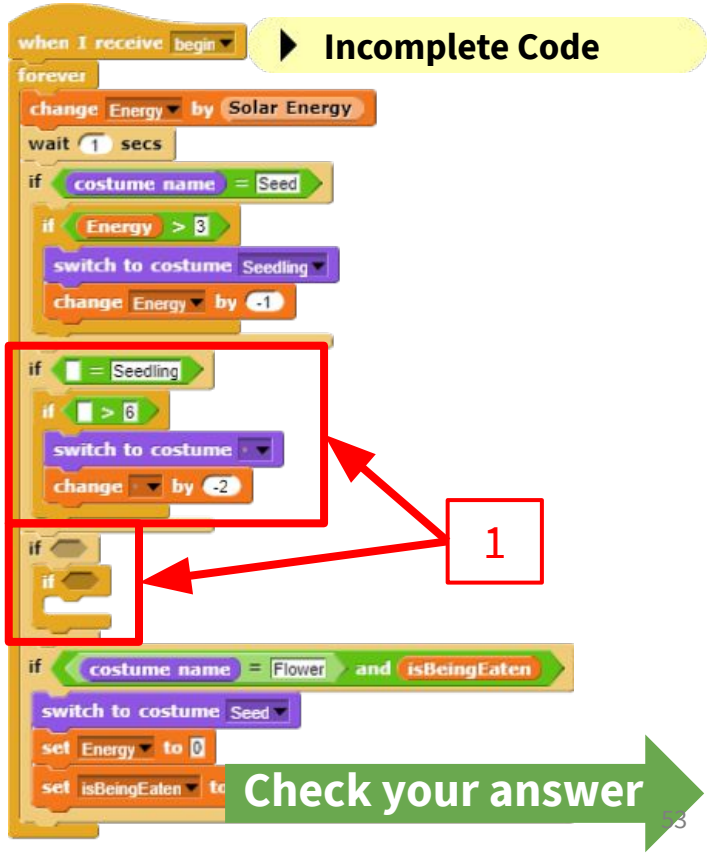


This plant has **incomplete code**.

1. Given the first **if** block, complete the **second and the third if** block.

Hint:

- ➢ The second condition should be: **IF** the costume is a seedling and **IF** Energy is greater than 6, switch to costume sapling and lose two energy
- ➢ The third condition should be: **IF** the costume is a sapling and **IF** Energy is greater than 12, switch to costume Flower and lose 4 energy



Step 2: Finish The Plants!

Is this what your code looks like?

1. Run your code and see how the plant behaves.
2. For each **Computational Thinking** element you see in the script, leave a **comment** on the side explaining the element in the **context** of the code.

Save your code!

Switch for the next activity!

Teacher A: Drive!
Teacher B: Navigate!

1. **Teacher A** Click on the 

button to switch to Driver
2. **After** Teacher A switched to Navigator, **teacher B** click on the 

to switch to Navigator
3. **Teacher A** should **share screen** and **perform all the coding actions**

Step 2: Finish The Plants!

Here are some examples. Yours don't have to be the same.

Pattern Recognition

The **script** uses the **forever** block to represent the **repeating** life cycle.

Abstractions

The **Energy** variable is an **abstraction** of life energy in numbers.

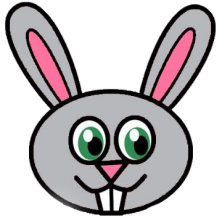
Decomposition

The **if** blocks **decompose** the state of the plant into four phases.

Algorithms

The whole **script** is a **step-by-step** description of the life cycle of a plant.

Step 3: Debug the Bunny!



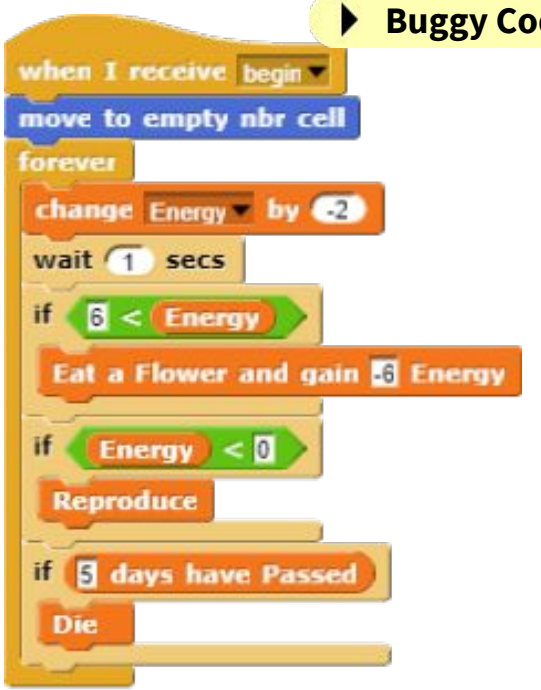
Now, switch to the **Bunny** sprite.

There are some **known bugs** in the bunny's code.

1. Follow the **hints** to **fix the bugs**:

➢ Bunny should move EVERYTIME in the loop
➢ **IF** its energy is LESS than 6, It should eat
➢ It should **GAIN energy** not LOSE IT when it eats
➢ **IF** it has no more energy, It should Die
➢ **IF** 5 days have passed, it should reproduce

Buggy Code



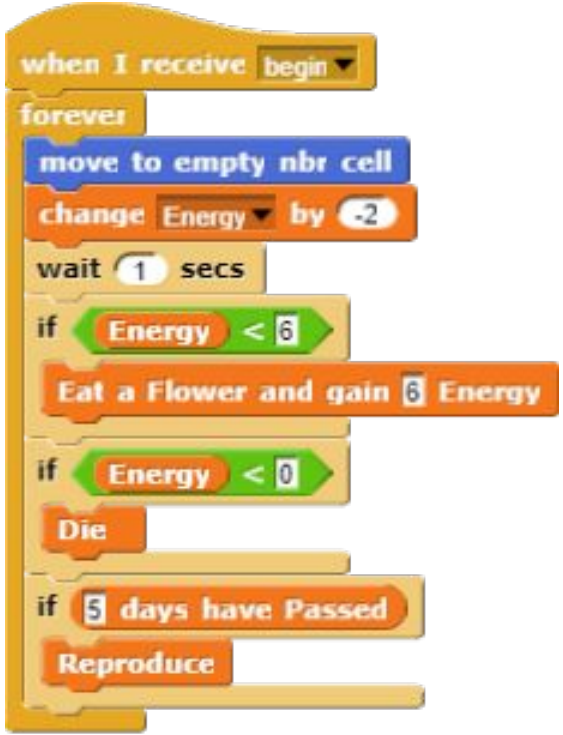
Check your answer

Step 3: Debug the Bunny!

What does your code look like?

It's okay to be different as long as the bugs are fixed!

1. Run your code and see if the bunny work as expected.
2. For each **Computational Thinking** element you see in the script, leave a **comment** on the side explaining the element in the **context** of the code.



Save your code!

Step 3: Debug the Bunny!

Here are examples. Yours don't have to be the same.

Pattern Recognition

The **script** uses the **forever** block to represent the **repeating** life cycle.

Abstractions

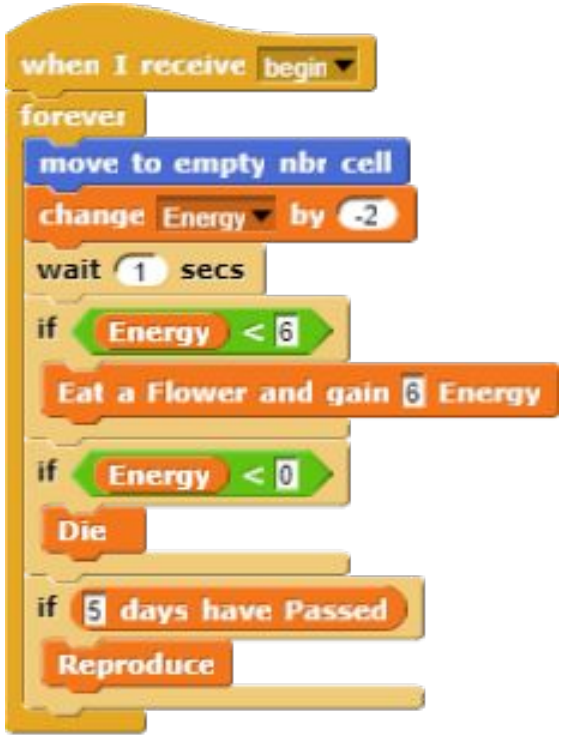
The script uses custom blocks to make **abstraction** of the bunny **behaviour**.

Decomposition

The if blocks **decompose** the energy level of the bunny into 2 conditions.

Algorithms

The **script** is a **step-by-step** description of a peaceful life cycle of a bunny.

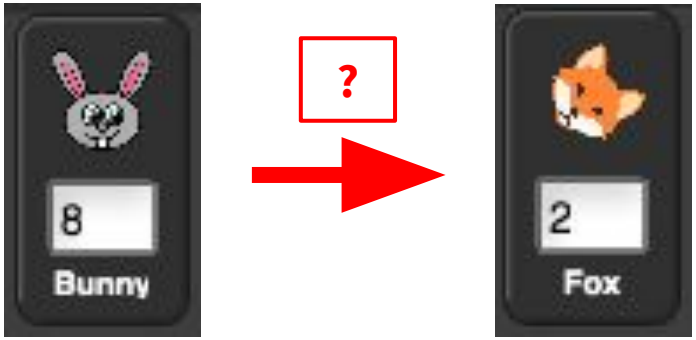


Step 4: Create the Fox!

We are going to use the **Patterns** in the Bunny Code to create the **Algorithm** for the Fox!

Worked Example **Create**

What behaviors are **similar and different** between a Bunny and a Fox?



Step 4: Create the Fox!

The Fox should have the following attributes:

- Set its Energy variable to 16 before it begins
- Move, lose 2 energy, and wait 1 second each time in a forever loop
- If the fox is low on energy (less than 6), it will eat a bunny and gain 8 energy
- If the fox has no more energy, it should die
- If 5 days have passed, the fox should reproduce!

1. Create the **code** for the **Fox** in the food web.

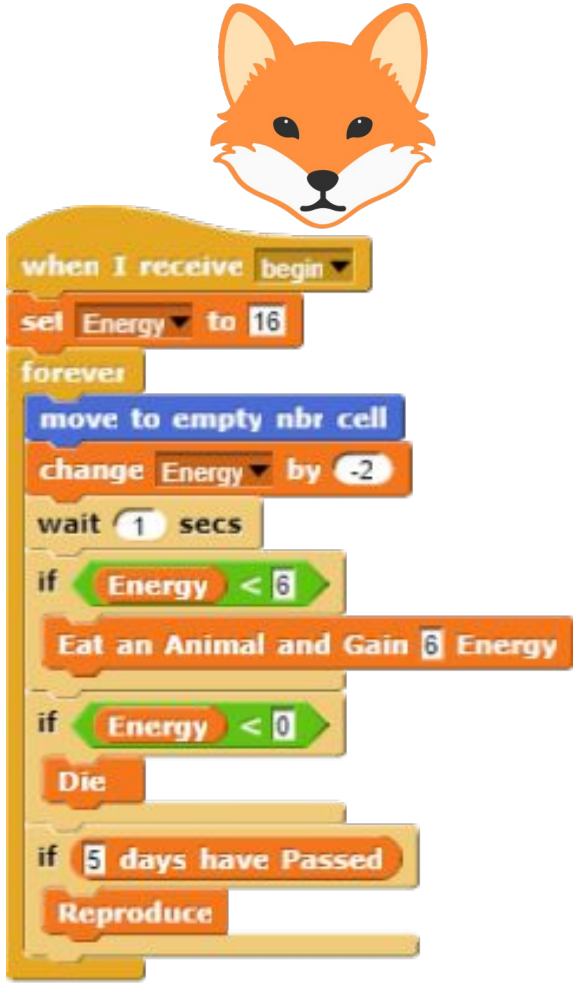
Check your answer

Step 4: Create the Fox!

Is this what your code looks like?

You might have different looking code, but as long as it has the same functions, it's correct!

Remember, a Fox is a ***Carnivore***. Make sure you are using the **Eat an Animal** Block.



Save your code!

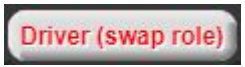
If there is time.....

Do you see any **Computational Thinking** PRADA elements in the above sequence?

Switch for the next activity!

Teacher A: Navigate!
Teacher B: Drive!

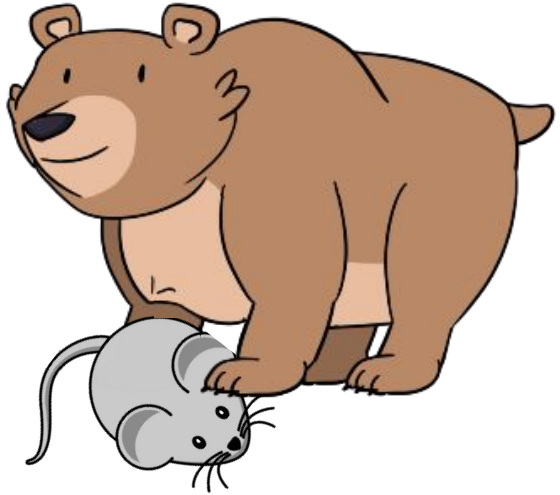
1. **Teacher A** should see the 

button
2. **Teacher B** should see the 

button
3. **Teacher B** should **share screen** and **perform all the coding actions**

Step 5: Choose what's next!

Make the Simulation Your Own!



Step 5: Choose what's next!

There are a bunch of different other animals available in this environment including:

- Mice
- Bears
- Eagles

Extend your code how you'd like **adding features** to the new animals!

You also can import your own animals and add those as well!

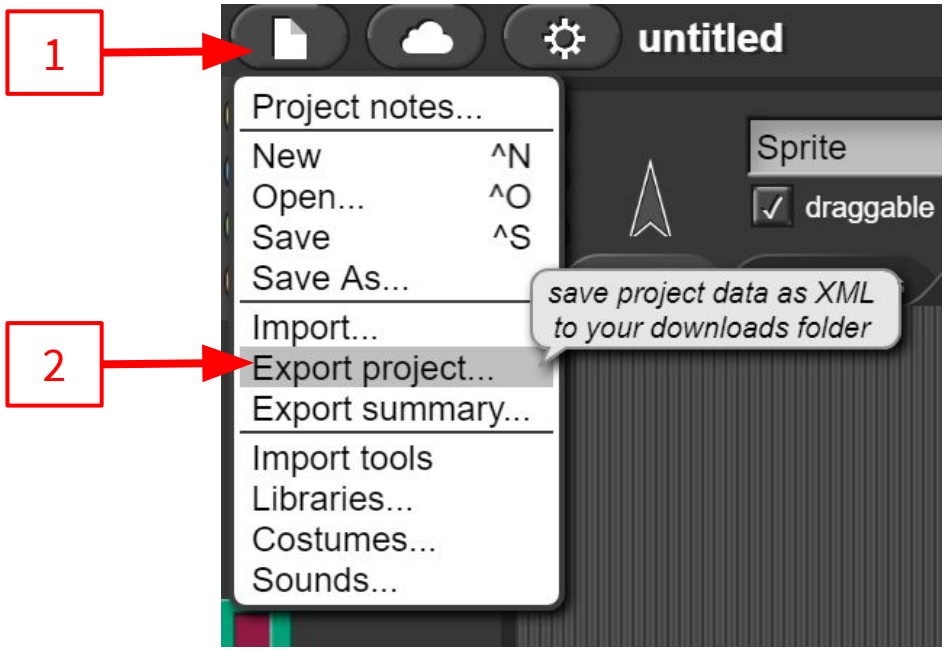
Make sure you add the appropriate **EAT custom block** for each animal type:

A mouse in this case is a ***herbivore***, eagles are ***carnivores***, and bears are ***omnivores***!

You've Completed Activity 2!

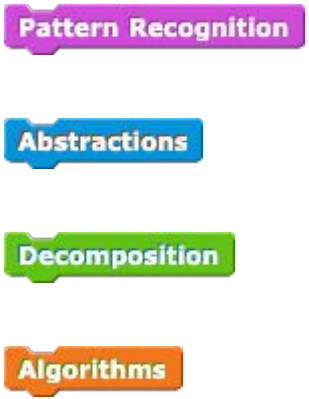
If you want to export and download the project to your computer, here is how:

1. Click on the File Menu
2. Select Export project
3. Choose where you want to save the project file and click save



Reflect

We saw **PRADA Concepts** in action



We also learned **Cellular! Concepts**

- How to do agent-based modeling
- How to make a simulation
- How to read, edit, and create code
- Different activity types - (Parsons problems, worked examples, extensions)

Congratulations on your coding conquest!

3C