



Object Oriented Fun

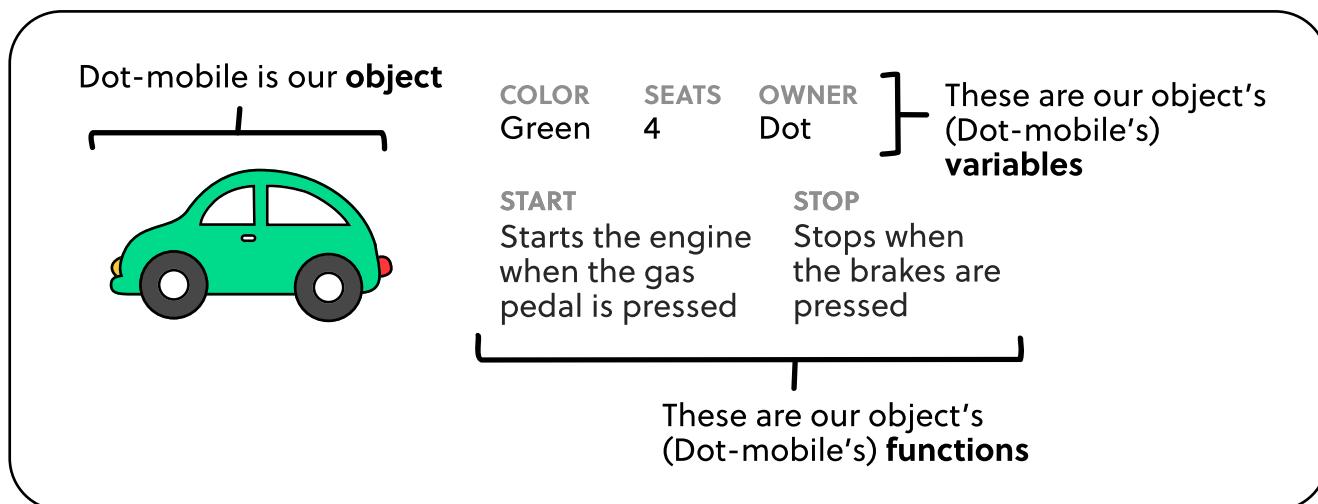
Welcome to Decomosphere!

Ready, set, learn! Dive into the grassy lands of Decomosphere with Dot and learn all about breaking down problems into smaller parts.

Vroom Vroom...

Object-oriented programming can be used to break down objects into their own unique variables and functions. We can make these objects can be just about anything, from cars to dogs to houses!

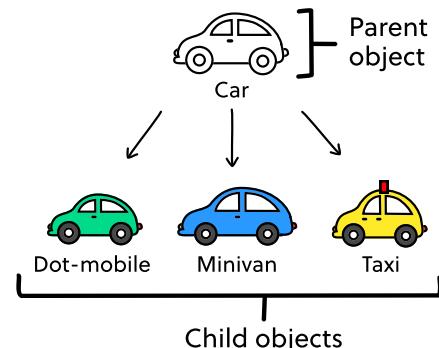
In Decomosphere, Dot and his friends love driving around town and catching the summer breeze. Let's take a deeper look at object oriented programming through Dot's car: the Dot-mobile!



More cars!

On the right, we can see that the Dot-mobile, minivan, and taxi are all different types of cars.

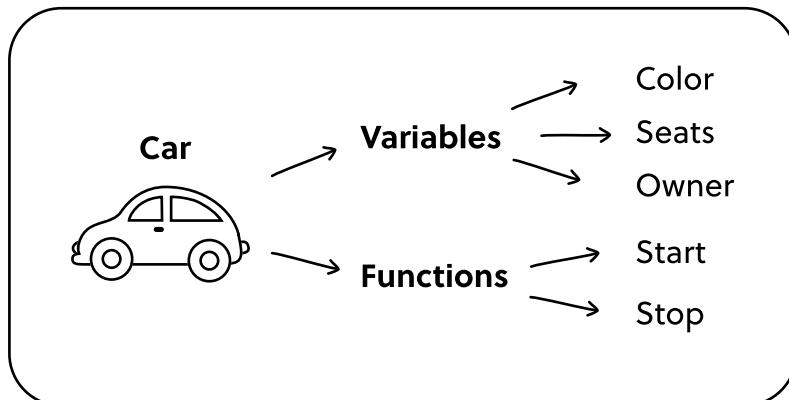
Looking at the complete picture, Dot-mobile belongs to a large **class** of cars. Since all of these cars share **similar properties**, we can create specific types of cars like the



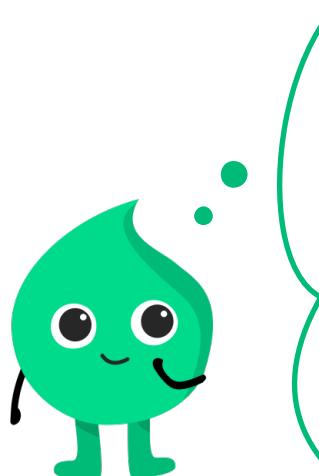
Dot-mobile, Minivan, and Taxi (called child objects) from our general car (called the parent object).

We call the general car our parent object, as it **passes down a set of commonly shared variables and functions** to its child object.

Let's take a closer look at the parent object:



Since all cars have a specific color, number of seats, owner, and can start and stop, **every child car** (the Dot-mobile, Minivan, and Taxi) **can be modeled after the parent car**.



Why is Object-Oriented Programming useful?

Programs often contain lots of code, which can be **messy and complicated**. When we have similar objects that share similar qualities but are not exactly the same, we can use inheritance in object-oriented programming to take the properties of a parent object and apply it to its child objects.

Following our car example, we can use our parent car object to create other cars like limousines or electric cars.

Let's Review Functions!

A **function** includes a **set of actions** for a certain task. Here's a quick refresher on how we can create our own functions in computer science:

Use **def** at the start to define our function

def [Your descriptive function title]:
Add your commands here!

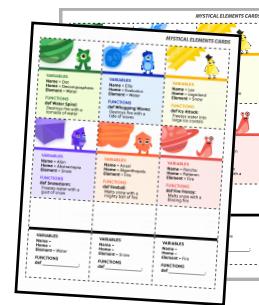
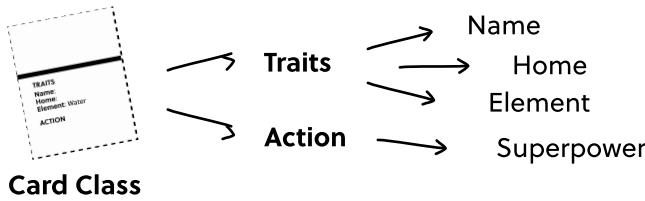
Begin the body of your function with a **colon**

Mystical Elements: The Game

Use your knowledge of object oriented programming to play a fun game of Mystical Elements with a friend! On your Card sheet, the first 6 cards each represent an object (a character from our 6 planets) with their unique variables and functions. Each function is a superpower that relates to water, snow, or fire.

Setup

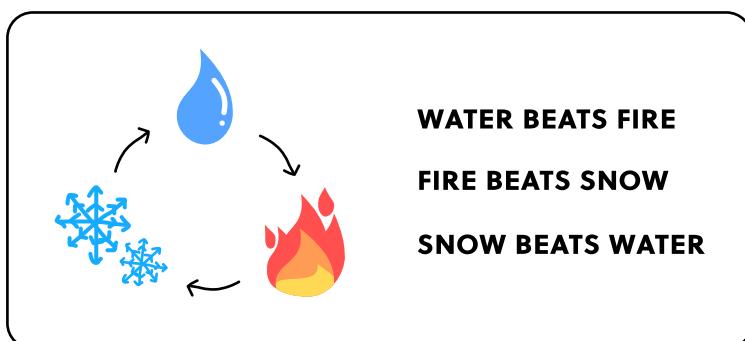
- Customize 3 different child objects on your Card sheet from the parent Card object. Assign your child objects variables (name and home) and a superpower function!



- Once you are finished customizing your child object cards, cut all of the cards out along the dashed lines.

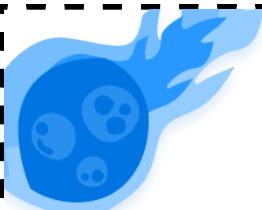
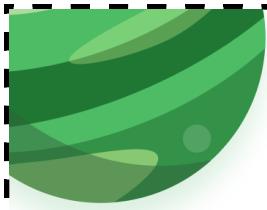
How to Play

- From your 9 cards, pick 1 card to play and place it face down.
- Once both of you have placed your cards down, flip them face up and compare your object's **element** variable to decide the winner! Water puts out fire, fire melts snow, and snow freezes water.
- The player who wins the round takes both cards.
- The person with the most cards wins!



Ex. Water beats fire





VARIABLES

Name = Dot
Home = Decomposphere
Element = Water

FUNCTIONS

def Water Spiral:
 Destroys fire with a tornado of water

VARIABLES

Name = Ellis
Home = Evaluatus
Element = Water

FUNCTIONS

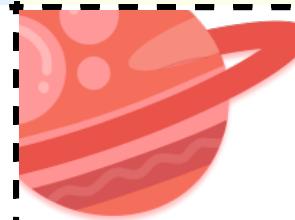
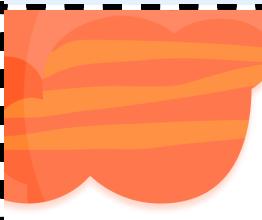
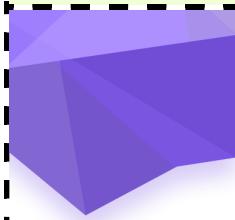
def Whopping Waves:
 Destroys fire with a tide of waves

VARIABLES

Name = Lex
Home = Logicland
Element = Snow

FUNCTIONS

def Icy Attack:
 Freezes water into large ice crystals



VARIABLES

Name = Alon
Home = Abstractopia
Element = Snow

FUNCTIONS

def Snowstorm:
 Freezes water with a gust of snow

VARIABLES

Name = Ansel
Home = Algorithopoly
Element = Fire

FUNCTIONS

def Fireball:
 Melts snow with a mighty ball of fire

VARIABLES

Name = Pancho
Home = Patteron
Element = Fire

FUNCTIONS

def Fire Frenzy:
 Melts snow with a blazing fire

VARIABLES

Name =
Home =
Element = Water

FUNCTIONS

def _____:

VARIABLES

Name =
Home =
Element = Snow

FUNCTIONS

def _____:

VARIABLES

Name =
Home =
Element = Fire

FUNCTIONS

def _____: