

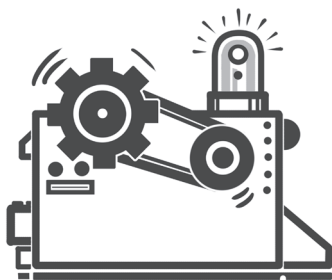
TRAPPED IN A VIDEO GAME

MORE TO EXPLORE

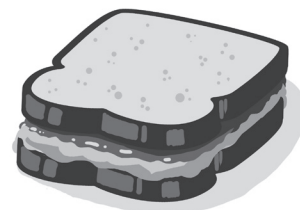
One of the first things you'll realize once you begin programming is that computers aren't smart, but they have great memories. For example, a computer could never figure out how to make a peanut butter and jelly sandwich on its own. But once you teach a computer how to make a PB&J, it'll never forget.

Because computers have such good memories, programmers use little shortcuts called **“functions”** to teach a job once and then repeat it over and over in their code.

The first time you want a computer to make a peanut butter and jelly sandwich, you'll need to teach it to get two slices of bread, spread peanut butter on one slice, spread jelly on the other slice, and then put the two slices together. However, if you want the computer to make lots of PB&Js in the future (and why wouldn't you?), you could just use a function to give that job a name—Peanut Butter and Jelly Sandwich. Now every time you type “Peanut Butter and Jelly Sandwich” into your code, the computer will remember how to build the sandwich and deliver a delicious lunch.



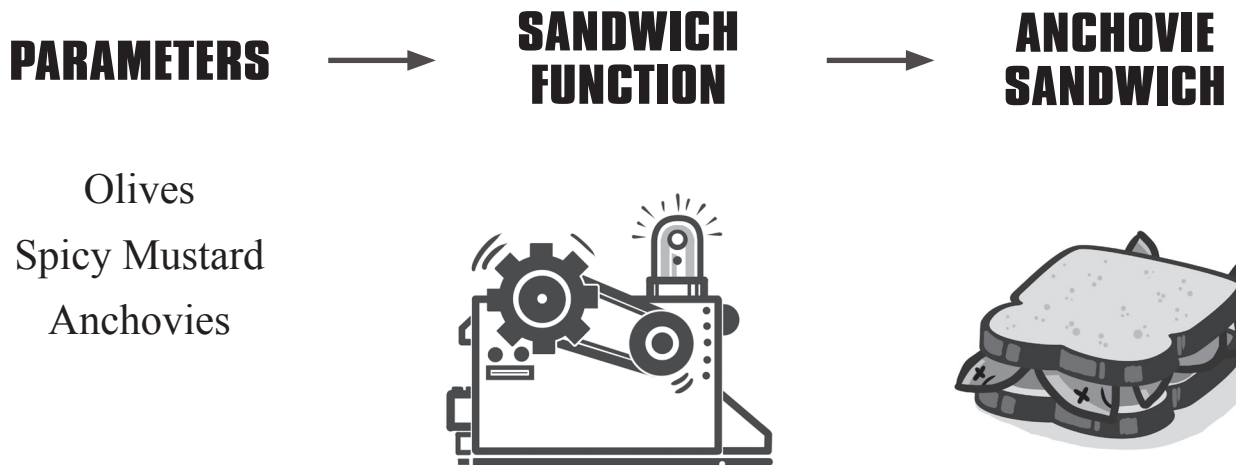
PB&J FUNCTION



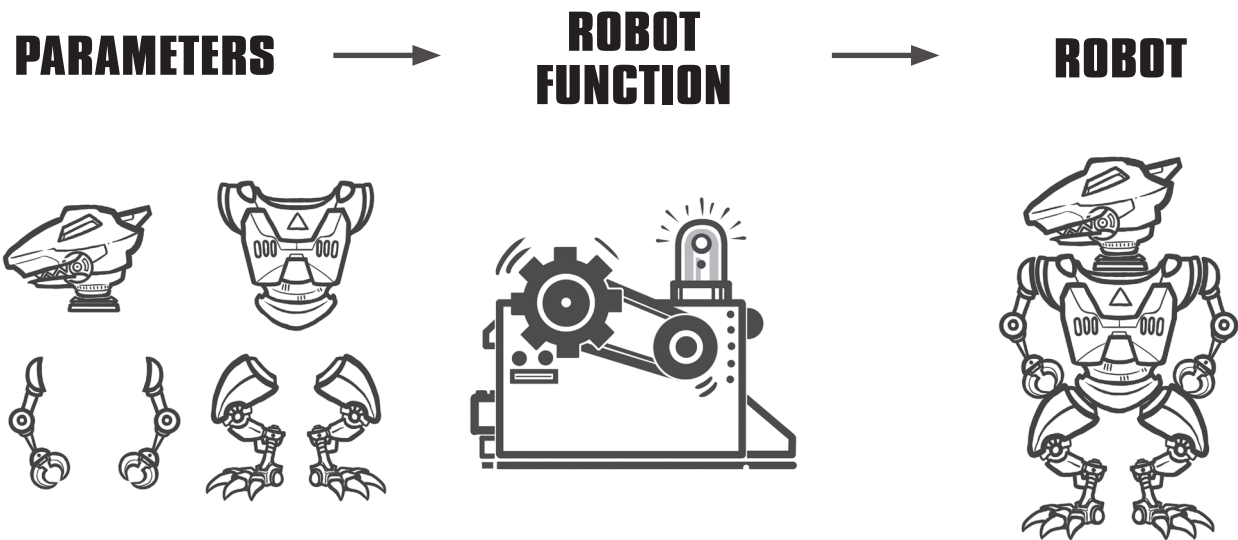
PB&J

That's pretty cool, but what's even cooler is that you can train functions to do a bunch of different jobs. For example, instead of naming the function Peanut Butter and Jelly Sandwich, you could just name it Sandwich. This function could then make peanut butter and jelly sandwiches, as well as ham and cheese sandwiches, BLT sandwiches, and those gross grown-up sandwiches with olives and spicy mustard.

You'd build this tasty function by telling the computer that a sandwich is a list of ingredients smoothed between two slices of bread. Then you can swap out ingredients to build whatever sandwich you want. Programmers call these ingredients "parameters."

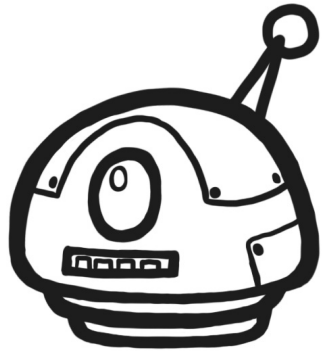


Trapped in a Video Game: Robots Revolt has a real-life function in it. If you've read the book already, you'll remember the walking robot factory Jesse sees in the sewer. The factory could assemble all kinds of robots—it just needed to know what parts to use.



In this section, you'll use functions and parameters to build custom robots. Check out the robot parts over the next four pages, then use the functions on the last two pages to put them together.

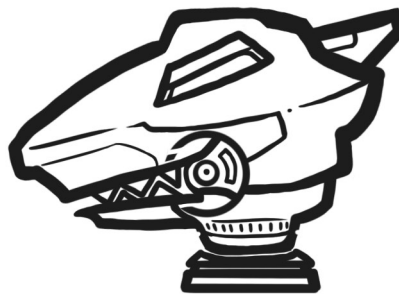
Heads



SPACE DROID



CYBORG GENIUS

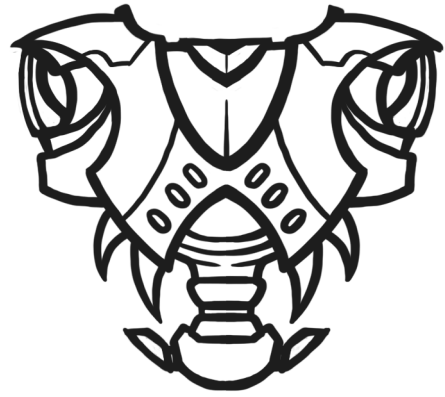


MECH SHARK

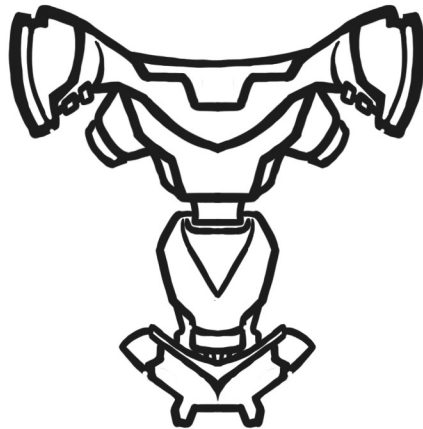
TORSOS



EXPLORATION SUIT



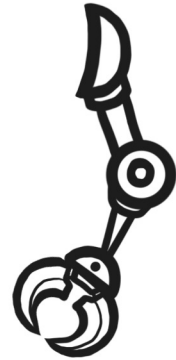
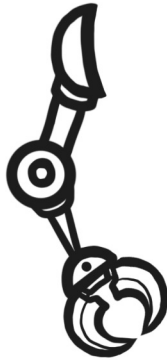
BATTLE SUIT



SPEED SUIT

Arms

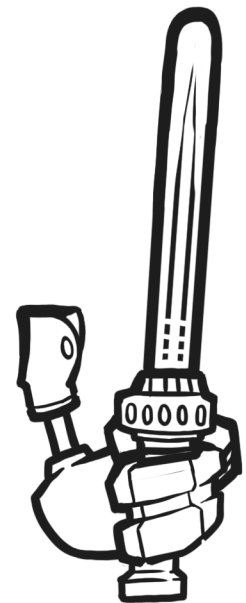
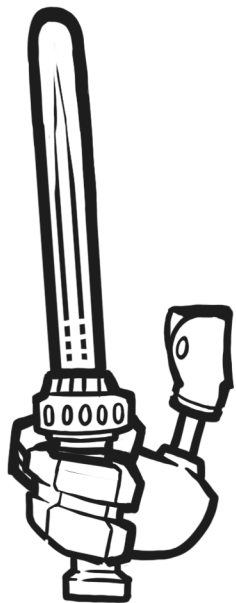
**CLAW
CONTROL**



**ROBO
REVENGE**

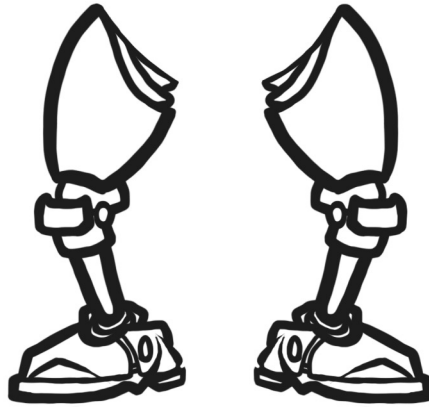


**SAMURAI
SLASHER**

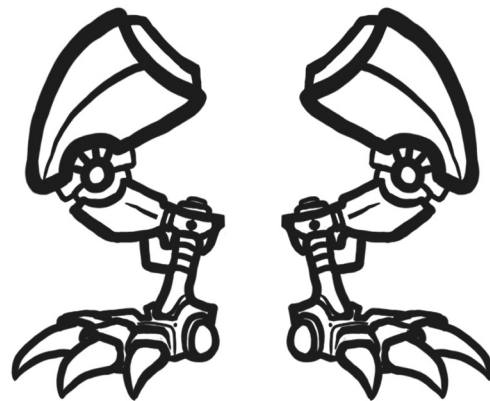


Legs

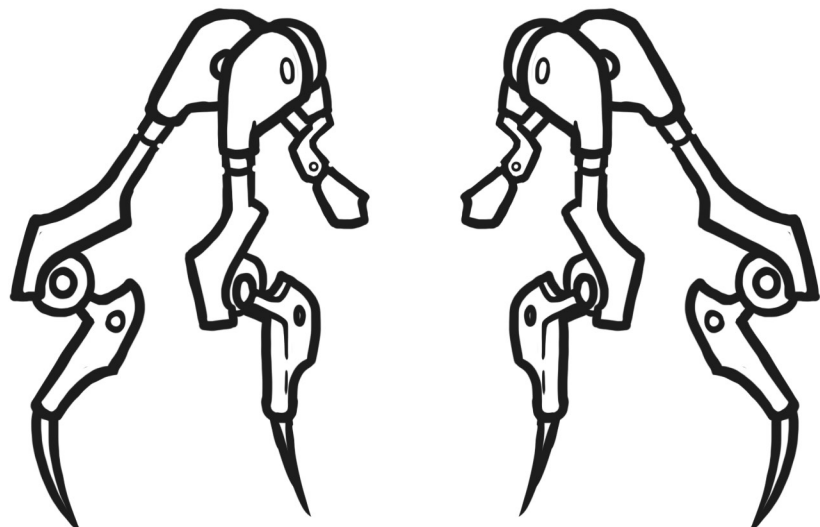
**FUTURE
KICKS**



**JURASSIC
STEEL**



**SPIDEY
SLICERS**



Robot Function Factory

Now for the fun part! In this exercise, you'll act as the robot function factory by assembling parts from the previous pages into complete robots. Simply draw or trace each robot part listed in the parameters below.

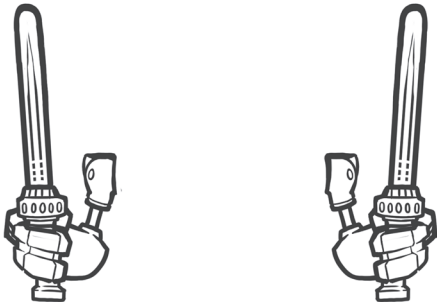
PARAMETERS



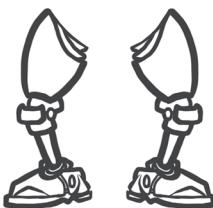
Mech Shark



Battle Suit



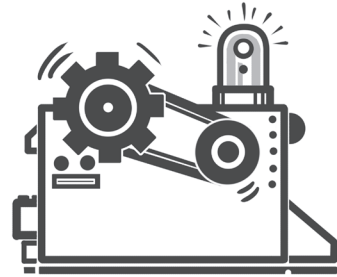
Samurai Slasher



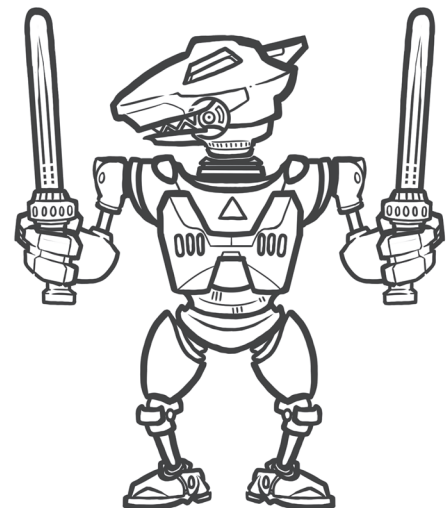
Future Kicks



ROBOT FUNCTION



ROBOT



Great White Sea Pirate

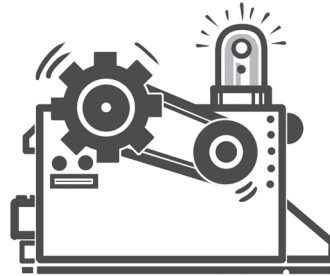


PARAMETERS

Cybor Genius
Speed Suit
Robo Revenge
Spidey Slicers



**ROBOT
FUNCTION**



ROBOT



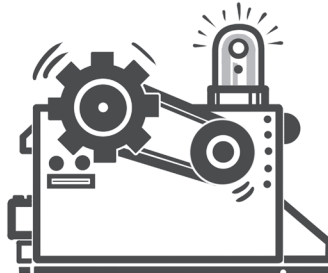
Tarantutron

PARAMETERS

Space Droid
Battle Suit
Claw Control
Jurassic Steel



**ROBOT
FUNCTION**



ROBOT



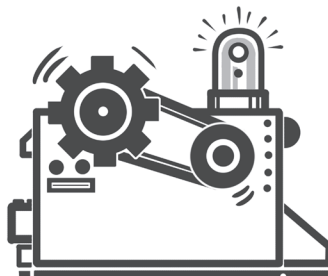
Velocibot 3000

PARAMETERS

Mech Shark
Exploration Suit
Robo Revenge
Spidey Slicers



**ROBOT
FUNCTION**



ROBOT



**Deep-Sea
Metal Monster**