```
// A function that does something
function saySomething() {
  p('i can only say this')
}
saySomething()

// Tell the function what to say
function sayThis(text) {
  p(text)
}
sayThis('this')

// Say two things
function sayTwoThings(one, two) {
  p(one)
  p(two)
}
sayTwoThings('this', 'that')

// A function that returns something
function returnSomething() {
  return 'I can only this'
}
let a = returnSomething()
p(a)

function returnThis(what) {
  return what
}
let b = returnThis('say what I tell you to')
p(b)

// Then show them how to wrap the function inside of p()
// Put a function inside of a function
p(returnThis('from inside'))


// So you may be wondering, why create a function that returns what you sent it.
Well, you don't. We only do that as a learning exercise. Real world functions change
or respond to what is sent to them. Let's try some functions that do math.
function multiplyTwoNumbers(num1, num2) {
  return num1 * num2
}
multiplyTwoNumbers(2, 3)


function p(text) {
  document.write(`<p>${text}</p>`)
}
```