



ARROW FUNCTIONS SYNTAX

```
...  
  
// Traditional function  
function add(a, b) {  
  return a + b;  
}  
  
// Arrow function  
const add = (a, b) => a + b; // Implicit return for single-line body
```





FUNCTIONS CALLING OTHER FUNCTIONS

In JavaScript, functions can call other functions, creating modular and reusable code. This allows you to break down complex tasks into smaller, more manageable units. There are several ways to call functions in JavaScript, each with its own purpose and best practices.



```
const cutPieces = function (fruit) {  
  return fruit * 4;  
}  
  
const fruitProcessor = function (apples, oranges) {  
  const applePieces = cutPieces(apples)  
  const orangesPieces = cutPieces(oranges)  
  
  const juice = `Juice with ${applePieces} pieces of apple and ${orangesPieces}  
pieces of orange`;  
  return juice;  
}  
  
const message = fruitProcessor(2, 4)
```




REVIEWING FUNCTION

Function name

Parameters: placeholders to receive input values. Like local variables of a function

Function body: block of code that we want to reuse. Processes the function's input data

```
function calcAge(birthYear, firstName) {  
  const age = 2037 - birthYear;  
  console.log(`${firstName} is ${age} years old`);  
  return age;  
}
```

return statement to output a value from the function and terminate execution

Calling, running or invoking the function, using ()

```
const age = calcAge(1991, 'Jonas');
```

Variable to save returned value (function output)

Arguments: actual values of function parameters, to input data