

# JAVASCRIPT FUNDAMENTALS – PART 2



JONAS.IO  
SCHMEDTMANN

# THE COMPLETE JAVASCRIPT COURSE

FROM ZERO TO EXPERT!



@JONASSCHMEDTMAN

SECTION

JAVASCRIPT FUNDAMENTALS - PART 2

LECTURE

FUNCTIONS CALLING OTHER  
FUNCTIONS





# CALLING A FUNCTION INSIDE A FUNCTION: DATA FLOW

```
const cutPieces = function (fruit) {  
  return fruit * 4;  
};  
  
const fruitProcessor = function (apples, oranges) {  
  
  const applePieces = cutPieces(apples);  
  const orangePieces = cutPieces(oranges);  
  
  const juice = `Juice with ${applePieces} pieces of  
apple and ${orangePieces} pieces of orange.`;  
  return juice;  
};  
  
console.log(fruitProcessor(2, 3));
```

The diagram illustrates the data flow in the provided JavaScript code. Red arrows represent the flow of data from the console log call to the fruitProcessor function, then to the cutPieces function, and back to the fruitProcessor function. A yellow arrow shows the flow of data from the console log call to the fruitProcessor function.

- The console log call `console.log(fruitProcessor(2, 3));` has two arguments: `2` (highlighted in a red box) and `3` (highlighted in a yellow box).
- The `fruitProcessor` function is called with `apples = 2` and `oranges = 3`. The value `2` is passed to the `cutPieces` function to calculate `applePieces`, and the value `3` is passed to the `cutPieces` function to calculate `orangePieces`.
- The `cutPieces` function calculates `applePieces = 2 * 4 = 8` and `orangePieces = 3 * 4 = 12`.
- The `fruitProcessor` function then constructs the `juice` string using these values: `const juice = `Juice with 8 pieces of apple and 12 pieces of orange.`;`





JONAS.IO  
SCHMEDTMANN

# THE COMPLETE JAVASCRIPT COURSE

FROM ZERO TO EXPERT!



@JONASSCHMEDTMAN

SECTION

JAVASCRIPT FUNDAMENTALS - PART 2

LECTURE

REVIEWING FUNCTIONS



# FUNCTIONS REVIEW: 3 DIFFERENT FUNCTION TYPES

## 👉 Function declaration

Function that can be used before it's declared



```
function calcAge(birthYear) {  
  return 2037 - birthYear;  
}
```

## 👉 Function expression

Essentially a function *value* stored in a variable



```
const calcAge = function (birthYear) {  
  return 2037 - birthYear;  
};
```

## 👉 Arrow function

Great for a quick one-line functions. Has no `this` keyword (more later...)



```
const calcAge = birthYear => 2037 - birthYear;
```

👉 Three different ways of writing functions, but they all work in a similar way: receive **input** data, **transform** data, and then **output** data.



# FUNCTIONS REVIEW: ANATOMY OF A FUNCTION

