



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 and circles represent the execution of the `fruitProcessor` function and its internal calls to `cutPieces`. A yellow arrow and circle represent the return value of `fruitProcessor` being passed to `console.log`.

- The `console.log` call passes arguments `2` and `3` to `fruitProcessor`.
- `fruitProcessor` receives `apples = 2` and `oranges = 3`.
- `fruitProcessor` calls `cutPieces(apples)`, which returns `8` (2×4).
- `fruitProcessor` calls `cutPieces(oranges)`, which returns `12` (3×4).
- `fruitProcessor` constructs the `juice` string and returns it.
- The returned string is passed to `console.log`.



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

