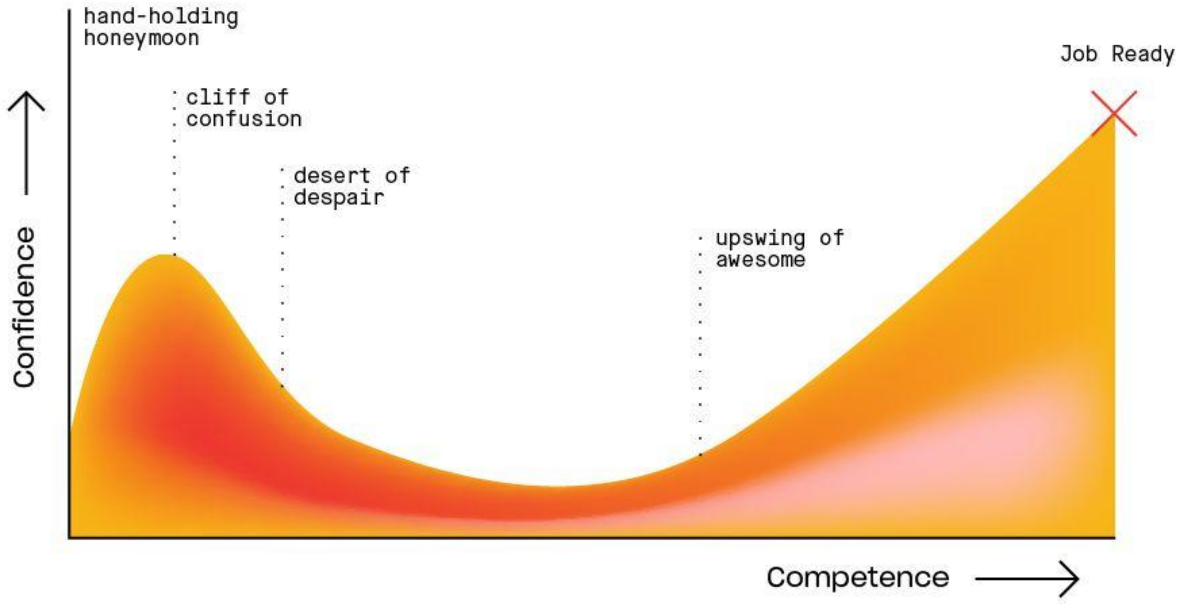
Week 3: Arrays & Functions

Agenda

- Introductions
- Questions?
- Arrays
- Functions
- Objects
- Next Week: ES6 and Intermediate Javascript

Coding Confidenct vs Competence



https://www.thinkful.com/blog/why-learning-to-code-is-so-damn-hard/

What questions do you have?

Arrays

Stores a **list of data**, any type:

```
let arrayOfManyThings = [“hello”, 5, true]
```

You can **add and remove** from an array:

```
arrayOfManyThings.push(“a new thing”)
arrayOfManyThings.pop()
```

You can **access** an element in an array **by its index**:

```
alert(arrayOfManyThings[2])
arrayOfManyThings[0] = “hola”
```

You can get the **length** of an array:

```
arrayOfManyThings.length
```

Arrays are used **very frequently** to store lists of data to display (messages, products, emails, tasks)

Array Methods

Map - creates a new array by calling a function on each element in an array

Reduce - gets a value by calling a function on each element in an array

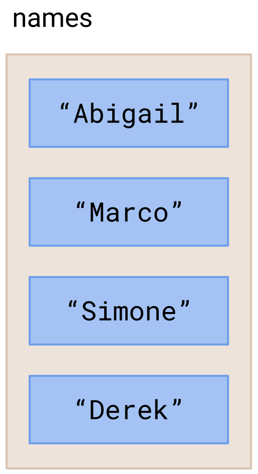
Foreach - calls a function on each element in an array

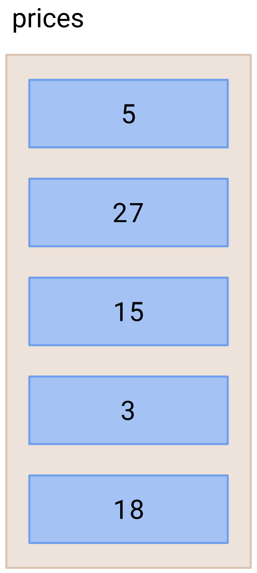
Filter - calls a function on each element in an array and creates a new array with the values that the function returned true on

Splice - a method that can add, remove or replace elements in an array

```
let names = [
  “Abigail”,
  “Marco”,
  “Simone”,
  “Derek”
]
```

```
let prices = [5, 27, 15, 3, 18]
```





Objects

A **group** of **data and functionality**

Property = Variable = Data
Method = Function = Functionality

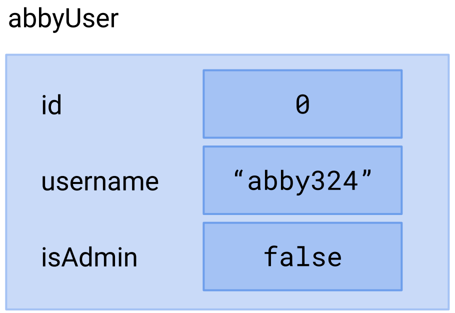
```
let message = {
    from: “Simone”,
    to: “Annapurna”,
    text: “Are you here yet?”,
    toString() {
        return this.text;
    }
}
```

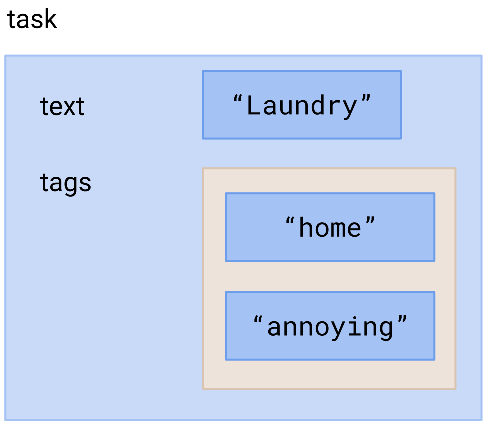
It’s very common to put **objects inside an array**:

```
let friends = [
    {
        name: “Annapurna”,
        age: 39,
        online: false
    },
    {
        name: “Simone”,
        age: 45,
        online: true
    }
]
```

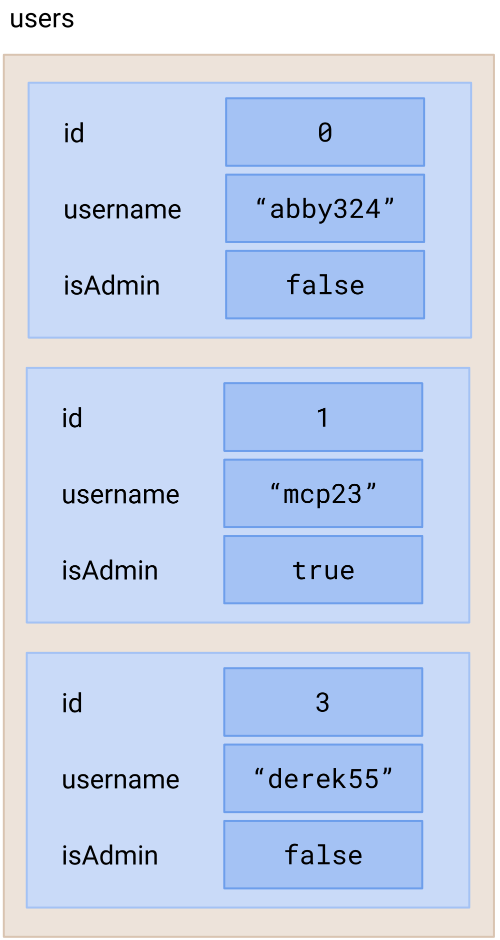
```
let abbyUser = {
  id: 0,
  username: “abby324”,
  isAdmin: false
}
```

```
let task = {
  text: “Laundry”
  tags: [“home”, “annoying”]
}
```

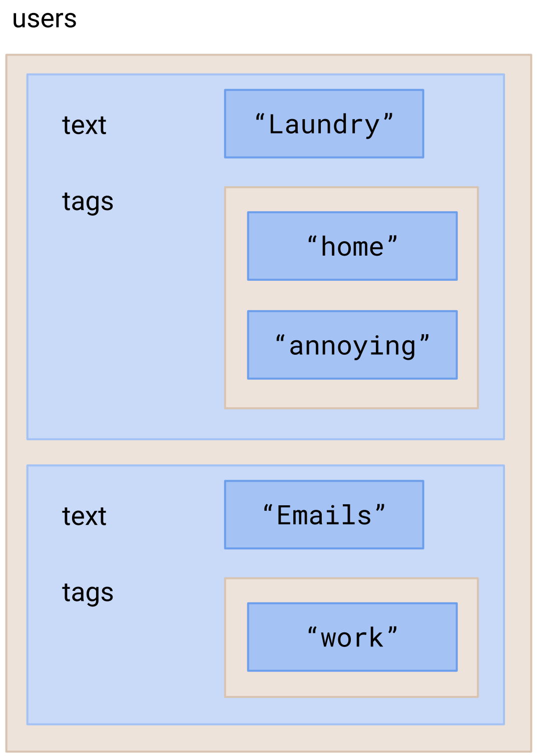




```
let users = [
  {
    id: 0,
    username: “abby324”,
    isAdmin: false
  },
  {
    id: 1,
    username: “mcp23”,
    isAdmin: true
  },
  {
    id: 3,
    username: “derek55”,
    isAdmin: false
  }
]
```



```
let tasks = [
  {
    text: “Laundry”
    tags: [
      “home”,
      “annoying”
    ]
  },
  {
    text: “Emails”
    tags: [
      “work”
    ]
  },
]
```



Functions

Why use functions?

1) Reusability

DRY = Don't Repeat Yourself

Having the same code in two (or more) places means you have to **keep them in sync**, a huge headache

2) Organization

Functions are the **building blocks** of Javascript

It's much **easier to read and understand** code that is broken into functions.

Multiple people working on the same app can **work on different functions** (often in different files) without stepping on each other's toes

Input & Output

Input = parameters

```
function myFunction(parameter1, parameter2) {
```

Output = return

```
return "output"
```

Showing something on the screen is NOT the same thing as **returning** something from a function

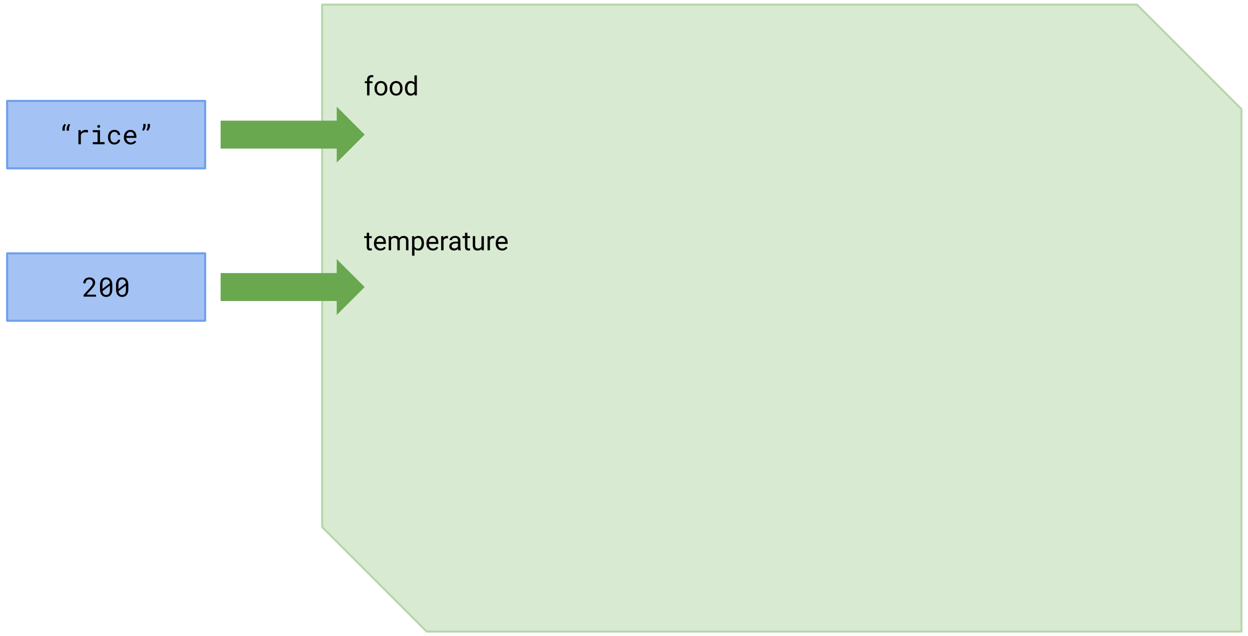
console.log("something") ≠ return "something"

Returning gives the info to another part of the code
Showing on the screen gives the info to the user

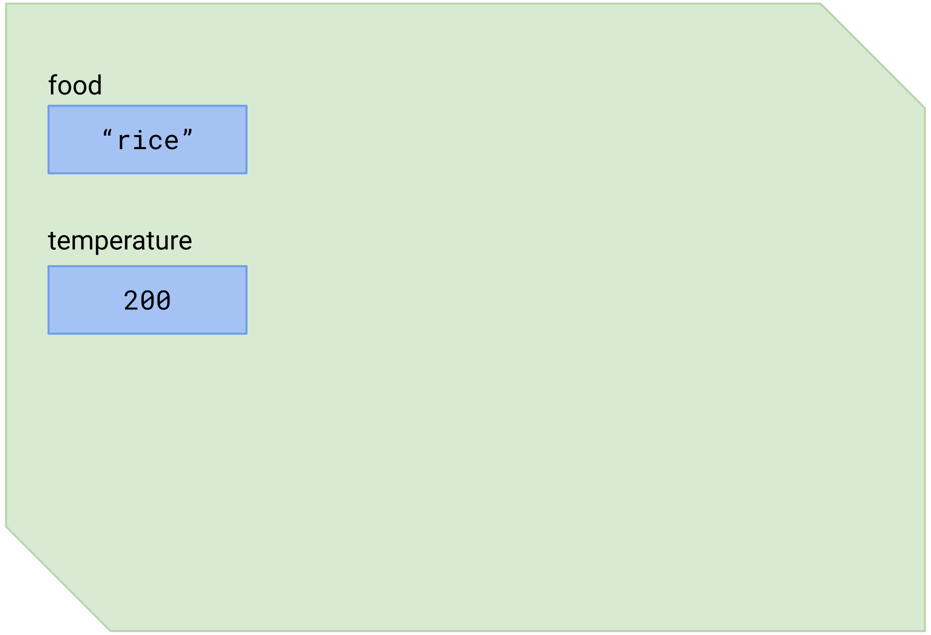
```
function cookInOven(food, temperature) {
  let cookedFood = “Cooked “ + food + “ at “ + temperature;
  return cookedFood;
}

let cookedRice = cookInOven(“rice”, 200)
console.log(cookedRice)
```

```
function cookInOven(food, temperature) {
  let cookedFood = “Cooked “ + food + “ at “ + temperature;
  return cookedFood;
}

let cookedRice = cookInOven(“rice”, 200)
console.log(cookedRice)
```

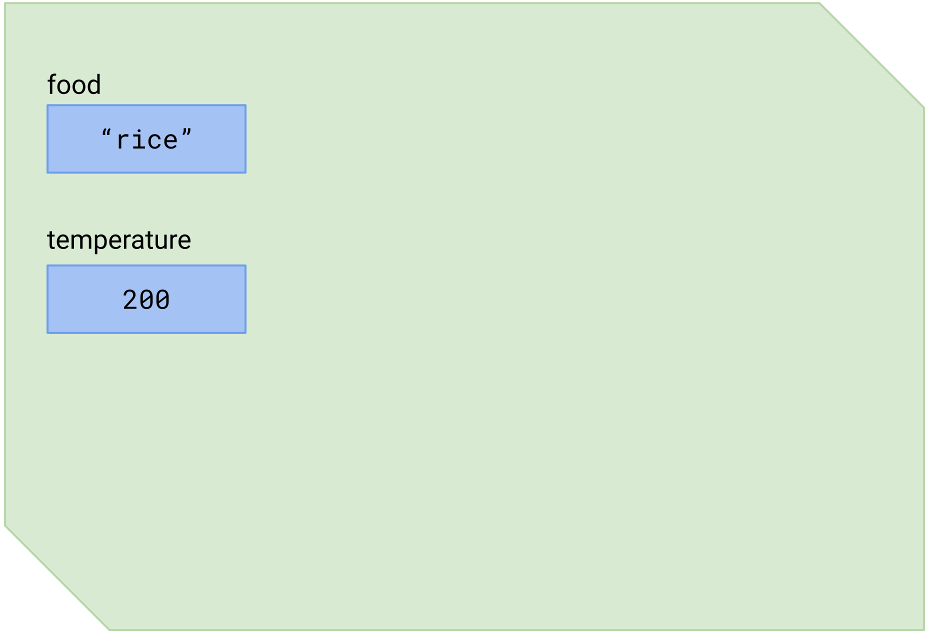
```
function cookInOven(food, temperature) {
  let cookedFood = “Cooked “ + food + “ at “ + temperature;
  return cookedFood;
}

let cookedRice = cookInOven(“rice”, 200)
console.log(cookedRice)
```



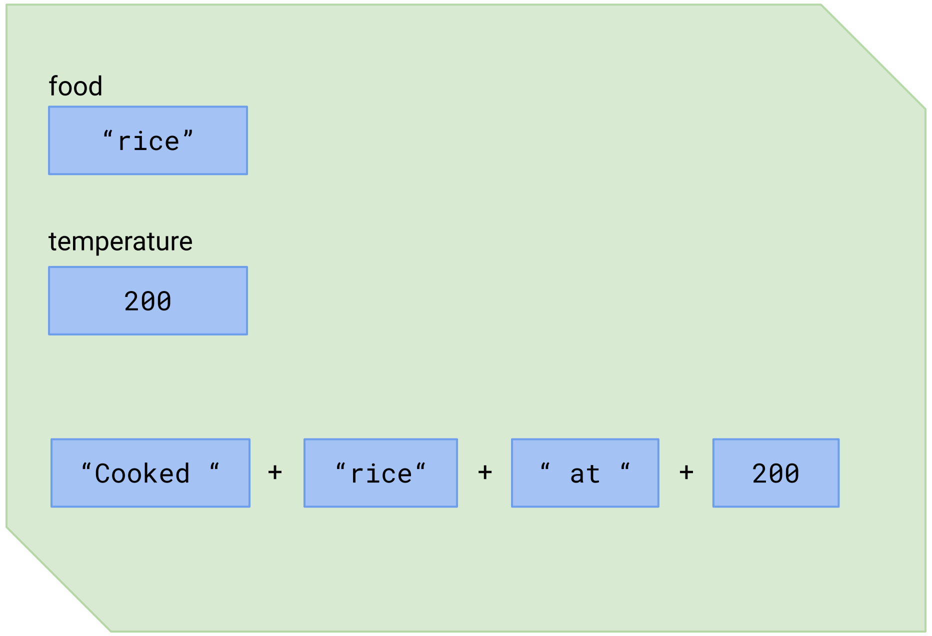
```
function cookInOven(food, temperature) {
  let cookedFood = “Cooked “ + food + “ at “ + temperature;
  return cookedFood;
}

let cookedRice = cookInOven(“rice”, 200)
console.log(cookedRice)
```



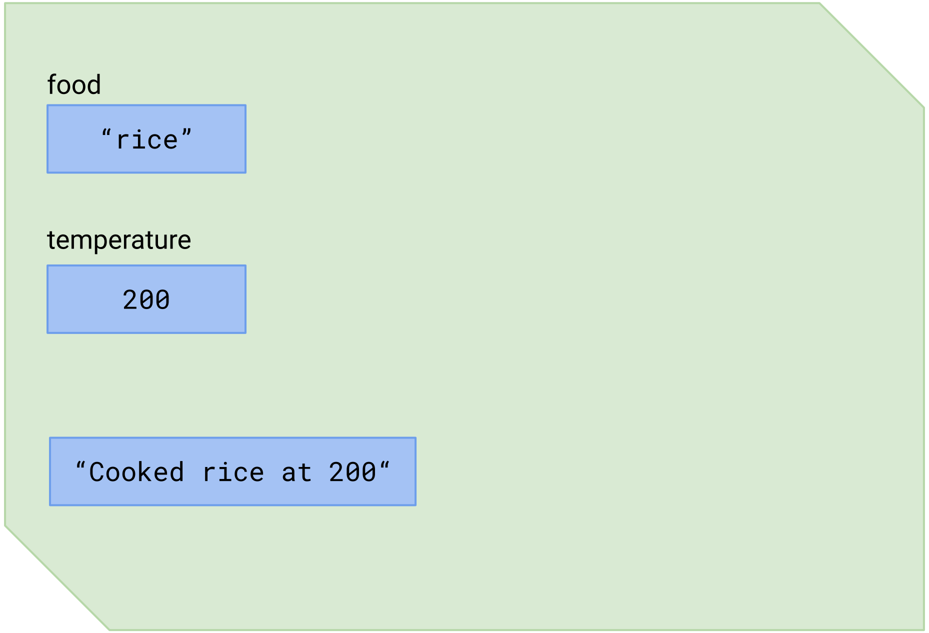
```
function cookInOven(food, temperature) {
  let cookedFood = "Cooked " + food + " at " + temperature;
  return cookedFood;
}

let cookedRice = cookInOven("rice", 200)
console.log(cookedRice)
```



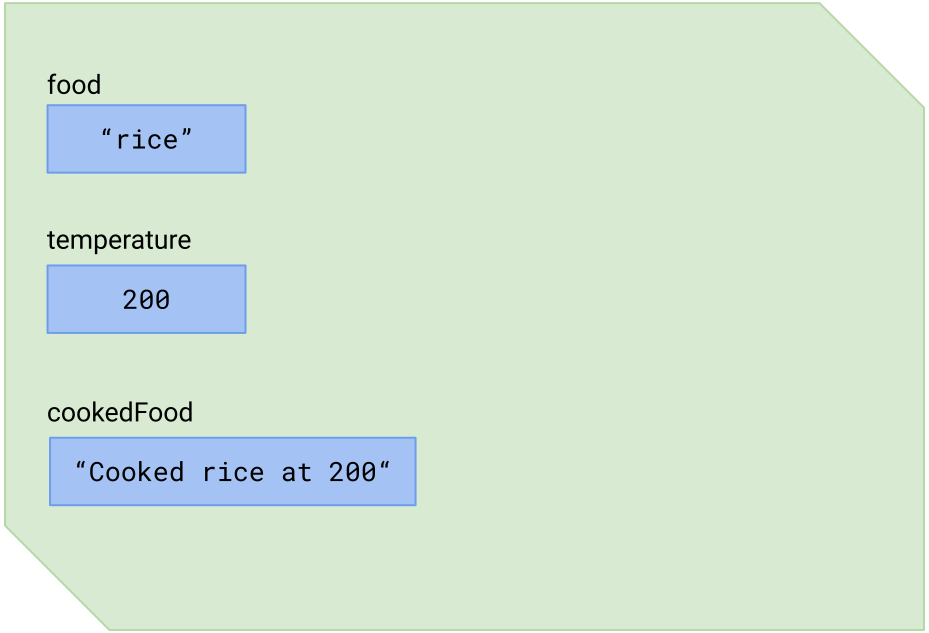
```
function cookInOven(food, temperature) {
  let cookedFood = “Cooked “ + food + “ at “ + temperature;
  return cookedFood;
}

let cookedRice = cookInOven(“rice”, 200)
console.log(cookedRice)
```



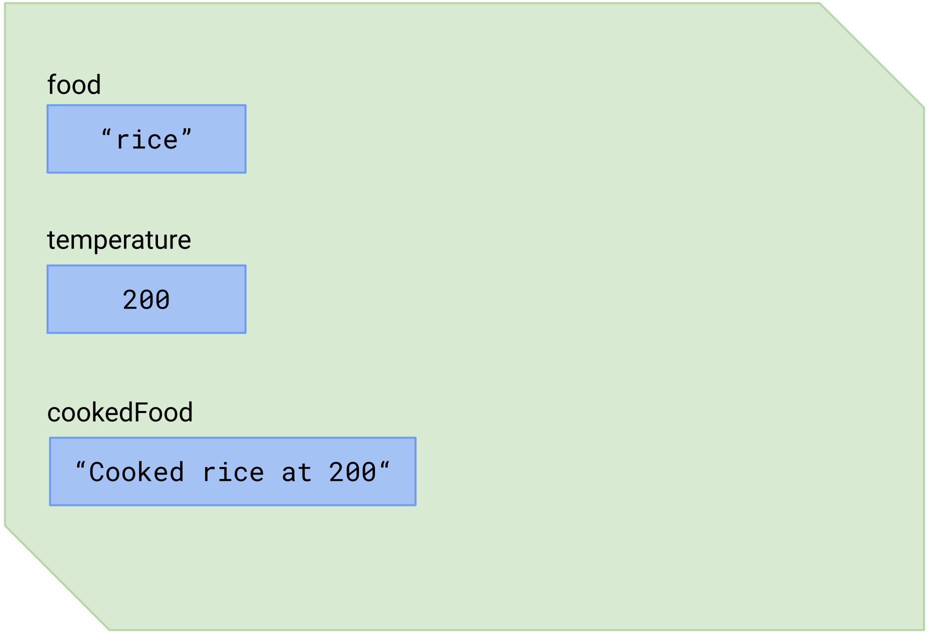
```
function cookInOven(food, temperature) {
  let cookedFood = “Cooked “ + food + “ at “ + temperature;
  return cookedFood;
}

let cookedRice = cookInOven(“rice”, 200)
console.log(cookedRice)
```



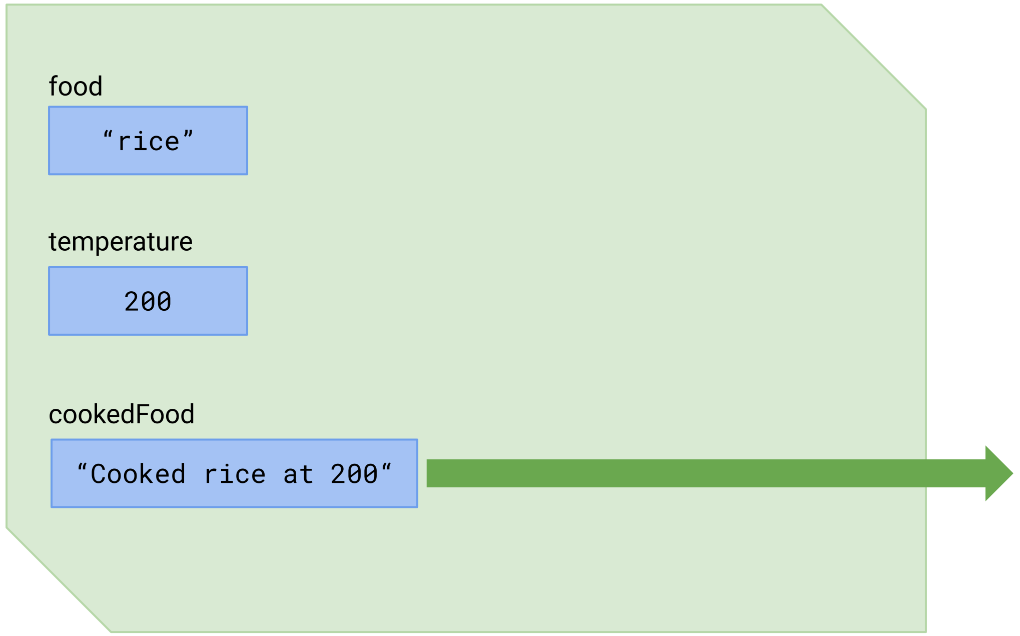
```
function cookInOven(food, temperature) {
  let cookedFood = “Cooked “ + food + “ at “ + temperature;
  return cookedFood;
}

let cookedRice = cookInOven(“rice”, 200)
console.log(cookedRice)
```



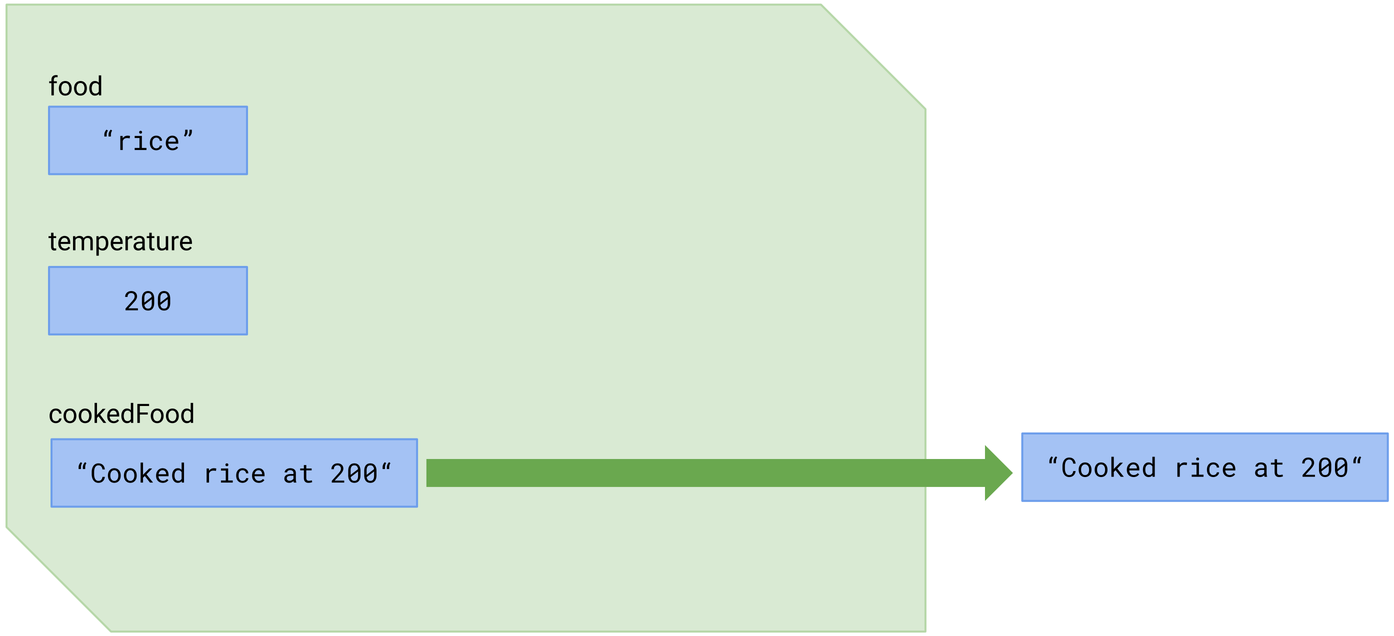
```
function cookInOven(food, temperature) {
  let cookedFood = “Cooked “ + food + “ at “ + temperature;
  return cookedFood;
}

let cookedRice = cookInOven(“rice”, 200)
console.log(cookedRice)
```



```
function cookInOven(food, temperature) {
  let cookedFood = “Cooked “ + food + “ at “ + temperature;
  return cookedFood;
}

let cookedRice = cookInOven(“rice”, 200)
console.log(cookedRice)
```



```
function cookInOven(food, temperature) {
  let cookedFood = “Cooked “ + food + “ at “ + temperature;
  return cookedFood;
}

let cookedRice = cookInOven(“rice”, 200)
console.log(cookedRice)
```



```
function cookInOven(food, temperature) {
  let cookedFood = “Cooked “ + food + “ at “ + temperature;
  return cookedFood;
}

let cookedRice = cookInOven(“rice”, 200)
console.log(cookedRice)
```

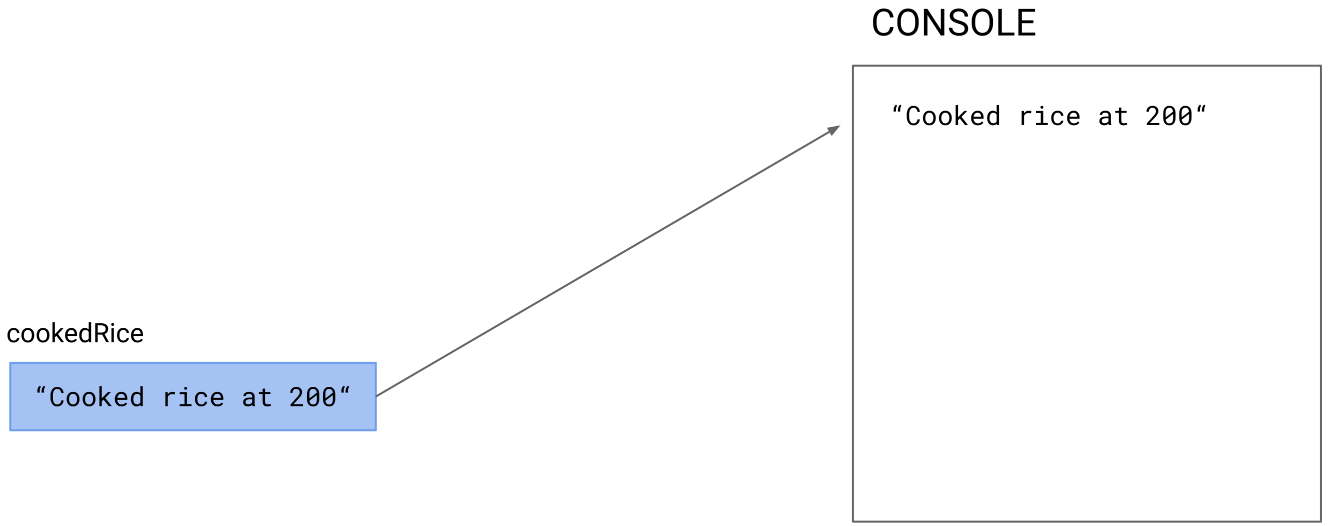
```
“Cooked rice at 200“
```

```
function cookInOven(food, temperature) {
  let cookedFood = “Cooked “ + food + “ at “ + temperature;
  return cookedFood;
}

let cookedRice = cookInOven(“rice”, 200)
console.log(cookedRice)
```

cookedRice

```
“Cooked rice at 200“
```

```
function cookInOven(food, temperature) {
  let cookedFood = “Cooked “ + food + “ at “ + temperature;
  return cookedFood;
}

let cookedRice = cookInOven(“rice”, 200)
console.log(cookedRice)
```



```
“Cooked rice at 200“
```

Tip: Calling Functions from Buttons

If you'd like a little more flexibility with your Javascript, you can set up **buttons that call your Javascript functions** when they are clicked.

1) Add this to your HTML body **before your <script>**:

```
<body>
    <button onclick="myFunction()">Button</button>
    <script src="yourjsfile.js"></script>
</body>
```

2) And make sure you have a function in your Javascript **with a matching name**:

```
function myFunction() {
    alert("You clicked the button!");
}
```

Additional Resources

Mosh Video on Functions

https://www.youtube.com/watch?v=N8ap4k_1QEQ&t=148s

The Net Ninja Video on Functions

https://www.youtube.com/watch?v=xUI5Tsl2JpY

(includes some arrow functions and array methods that we'll dig into next week)