

# You don't need for loops



lichtsteinerdev



davidlichtsteiner



lichtsteinerdev



lichtsteinerdev



# David Lichtsteiner

BSc Informatik | david@lichtsteiner.li | Software Engineer  
working@CSSVersicherung



lichtsteinerdev



davidlichtsteiner



lichtsteinerdev



lichtsteinerdev



# Disclaimer



**Learn the principle**



**Try to use it**



**Try to understand it**



**When it makes sense – break it**

## <Exercise 1 />

Write a function that returns an array with the email addresses of all users using a common loop.

Imperative  
Programming

vs

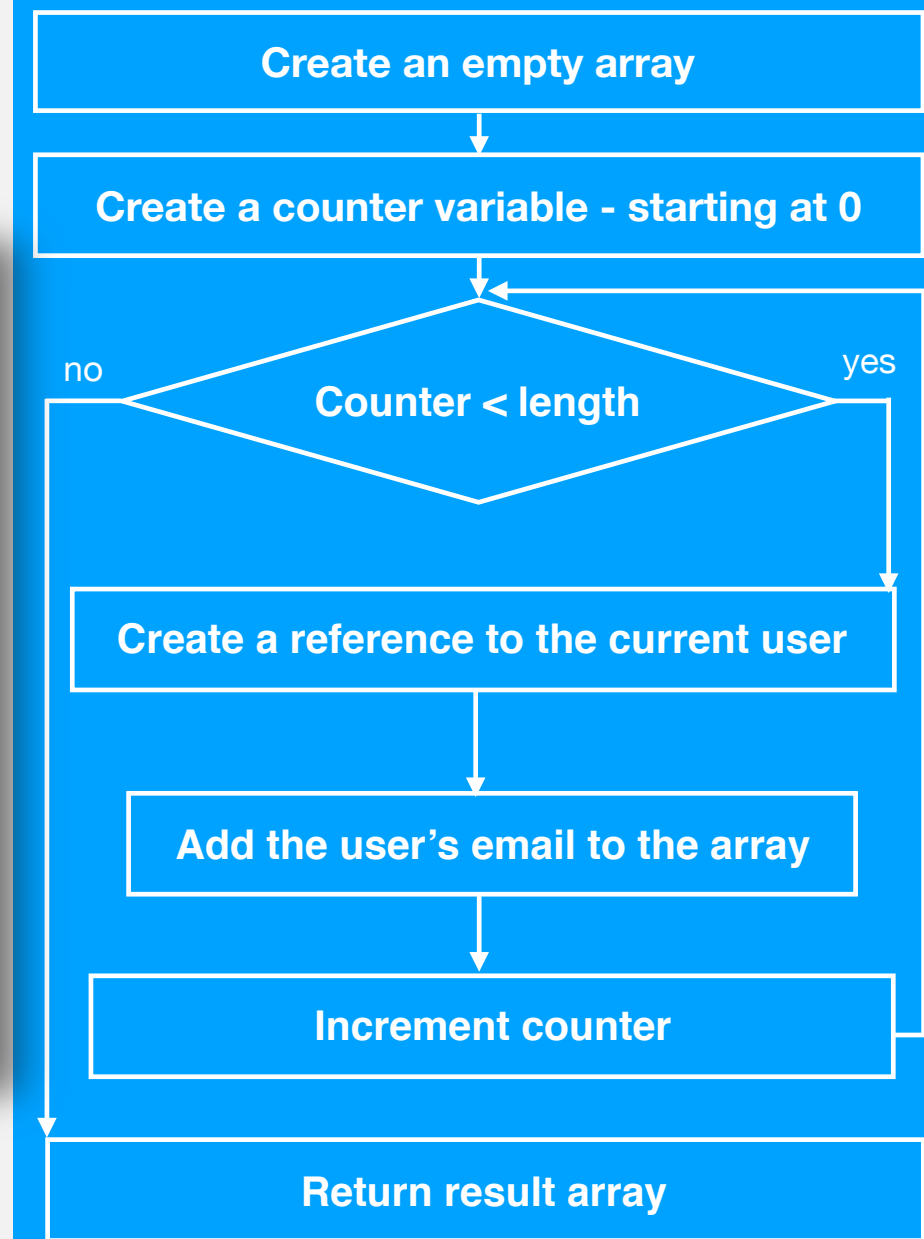
Declarative  
Programming

# Imperative Programming

```
function getClientEmails(clients) {  
  const emails = [];  
  
  for (let i = 0; i < clients.length; i++) {  
    const client = clients[i];  
    emails.push(client.email);  
  }  
  
  return emails;  
}
```

**HOW**

```
function getClientEmails(clients) {  
  const emails = [];  
  
  for (let i = 0; i < clients.length; i++) {  
    const client = clients[i];  
    emails.push(client.email);  
  }  
  
  return emails;  
}
```





# Declarative Programming

```
SELECT email FROM clients WHERE email IS NOT NULL
```

**WHAT**

# Why?

**Less Bugs** - easier to reason about

**Less Time** - reuse more code

# What?

## Higher Order Function

*/ˈhʌɪə/ awr-der fuhngk-shuh n*

A higher order function is a function that takes another function as a parameter, returns a function, or does both.

# How?

```
const clientEmails = [];  
  
for (let client of clients) {  
  const clientEmail = client.email;  
  
  clientEmails.push(clientEmail);  
}
```

```
const clientEmails = [];  
  
for (let client of clients) {  
    const clientEmail = client.email;  
  
    clientEmails.push(clientEmail);  
}
```

```
const stockTotals = [];  
  
for (let item of inventory.items) {  
    const stockTotal = {  
        item: item.name,  
        value: item.quantity * item.price  
    };  
    stockTotals.push(stockTotal)  
}
```

```
const clientEmails = [];  
  
for (let client of clients) {  
    const clientEmail = client.email;  
  
    clientEmails.push(clientEmail);  
}
```

```
const stockTotals = [];  
  
for (let item of inventory.items) {  
    const stockTotal = {  
        item: item.name,  
        value: item.quantity * item.price  
    };  
    stockTotals.push(stockTotal)  
}
```

```
const clientEmails = [];  
  
for (let client of clients) {  
    const clientEmail = client.email;  
  
    clientEmails.push(clientEmail);  
}
```

```
const stockTotals = [];  
  
for (let item of inventory.items) {  
    const stockTotal = {  
        item: item.name,  
        value: item.quantity * item.price  
    };  
    stockTotals.push(stockTotal)  
}
```



```
const results = [];  
  
for (let item of items) {  
    const result = item.email;  
  
    results.push(result);  
}
```

```
const results = [];  
  
for (let item of items) {  
    const result = {  
        item: item.name,  
        value: item.quantity * item.price  
    };  
    results.push(result)  
}
```

```
const results = [];  
  
for (let item of items) {  
  const result = item.email;  
  
  results.push(result);  
}
```

```
const results = [];  
  
for (let item of items) {  
  const result = {  
    item: item.name,  
    value: item.quantity * item.price  
  };  
  results.push(result)  
}
```

```
const callback = (client) => {  
  return client.email;  
};  
  
const results = [];  
  
for (let item of items) {  
  const result = callback(item);  
  results.push(result);  
}
```

```
const callback = (item) => {  
  return {  
    product: item.name,  
    value: item.quantity * item.price  
  };  
};  
  
const results = [];  
  
for (let item of items) {  
  const result = callback(item);  
  results.push(result)  
}
```

```
const callback = (client) => {  
  return client.email;  
};  
  
const results = [];  
  
for (let item of items) {  
  const result = callback(item);  
  results.push(result);  
}
```

```
const callback = (item) => {  
  return {  
    product: item.name,  
    value: item.quantity * item.price  
  };  
};  
  
const results = [];  
  
for (let item of items) {  
  const result = callback(item);  
  results.push(result)  
}
```

```
const callback = (client) => {  
  return client.email;  
};  
  
const results = [];  
  
for (let item of items) {  
  const result = callback(item);  
  results.push(result);  
}
```


```
const callback = (item) => {  
  return {  
    product: item.name,  
    value: item.quantity * item.price  
  };  
};  
  
const results = [];  
  
for (let item of items) {  
  const result = callback(item);  
  results.push(result)  
}
```

```
const callback = (client) => {  
  return client.email;  
};  
  
const results = clients.map(callback);
```

```
const callback = (item) => {  
  return {  
    product: item.name,  
    value: item.quantity * item.price  
  };  
};  
  
const results = clients.map(callback);
```

## <Exercise 2 />

Write a function that returns an array with the email addresses of all users using a higher order function.



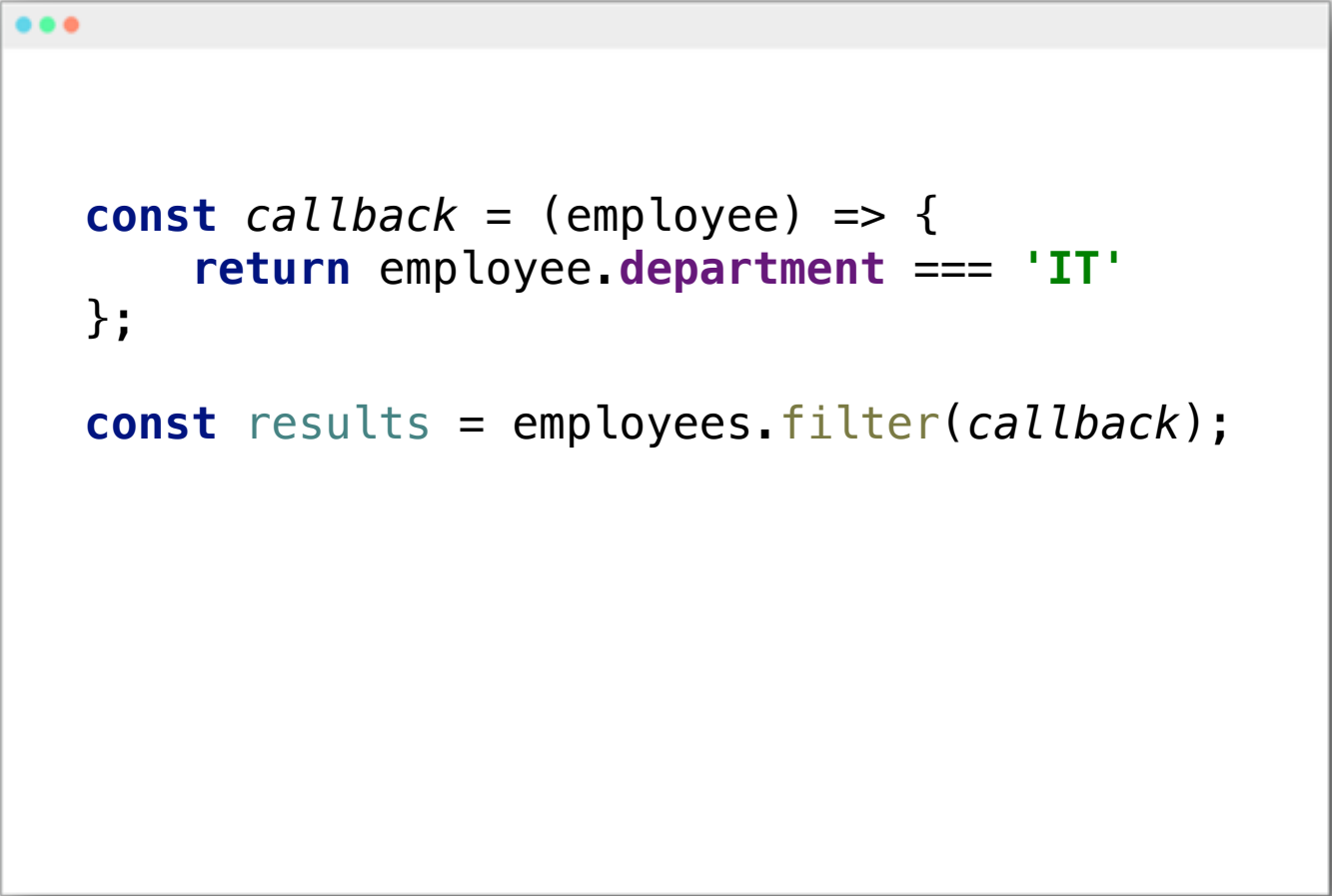
```
const itEmployees = [];  
  
for (let employee of employees) {  
  if(employee.department === 'IT') {  
    itEmployees.push(employee)  
  }  
}
```



```
const results = [];  
  
for (let item of items) {  
  if(item.department === 'IT') {  
    results.push(item)  
  }  
}
```

```
const results = [];  
  
for (let item of items) {  
  if(item.department === 'IT') {  
    results.push(item)  
  }  
}
```

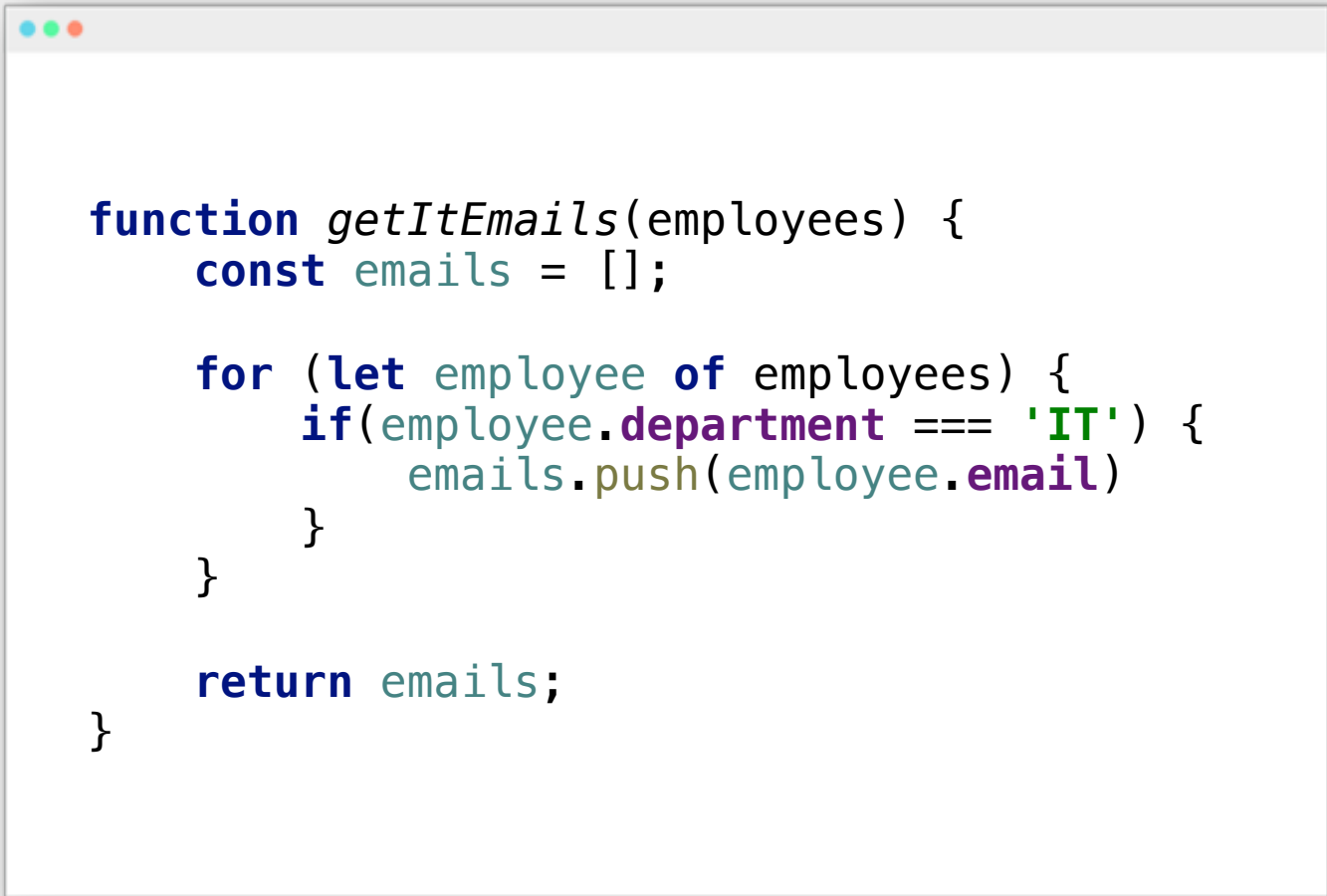
```
const callback = (employee) => {  
    return employee.department === 'IT'  
};  
  
const results = [];  
  
for (let item of items) {  
    if(callback(item)) {  
        results.push(item);  
    }  
}
```



```
const callback = (employee) => {  
    return employee.department === 'IT'  
};  
  
const results = employees.filter(callback);
```

## <Exercise 3 />

Write a function that returns the sales employees using a higher order function.

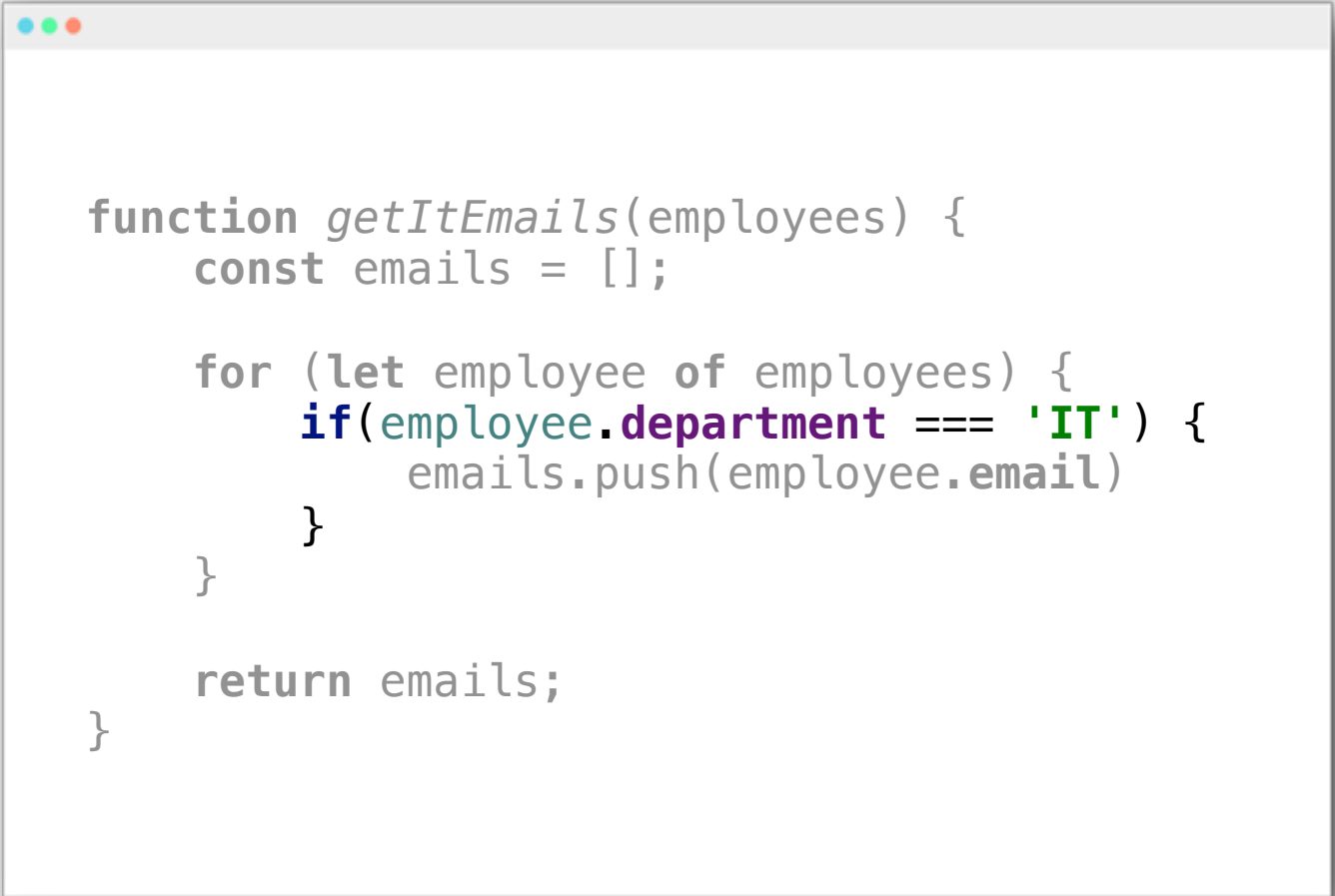


```
function getItEmails(employees) {  
  const emails = [];  
  
  for (let employee of employees) {  
    if(employee.department === 'IT') {  
      emails.push(employee.email)  
    }  
  }  
  
  return emails;  
}
```

```
function getItEmails(employees) {  
  const emails = [];  
  
  for (let employee of employees) {  
    if(employee.department === 'IT') {  
      emails.push(employee.email)  
    }  
  }  
  
  return emails;  
}
```



map..?




```
function getItEmails(employees) {  
  const emails = [];  
  
  for (let employee of employees) {  
    if(employee.department === 'IT') {  
      emails.push(employee.email)  
    }  
  }  
  
  return emails;  
}
```



---

**THINK IN STEPS**

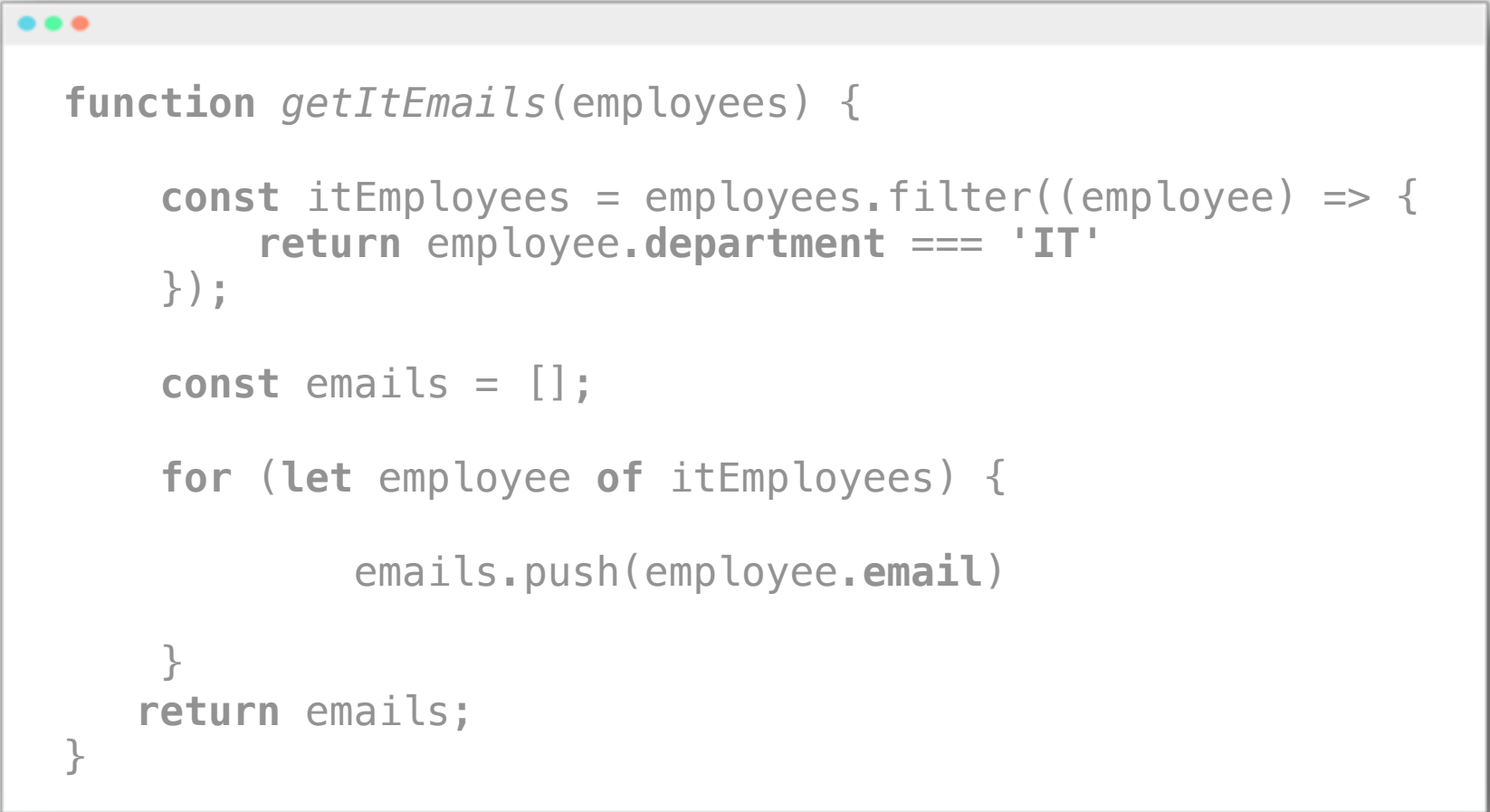


```
function getItEmails(employees) {  
  
    const emails = [];  
  
    for (let employee of employees) {  
        if(employee.department === 'IT') {  
            emails.push(employee.email)  
        }  
    }  
    return emails;  
}
```

```
function getItEmails(employees) {  
  
    const itEmployees = employees.filter((employee) => {  
        return employee.department === 'IT'  
    });  
  
    const emails = [];  
  
    for (let employee of employees) {  
        if(employee.department === 'IT') {  
            emails.push(employee.email)  
        }  
    }  
    return emails;  
}
```

```
function getItEmails(employees) {  
  
    const itEmployees = employees.filter((employee) => {  
        return employee.department === 'IT'  
    });  
  
    const emails = [];  
  
    for (let employee of itEmployees) {  
        if(employee.department === 'IT') {  
            emails.push(employee.email)  
        }  
    }  
    return emails;  
}
```

```
function getItEmails(employees) {  
    const itEmployees = employees.filter((employee) => {  
        return employee.department === 'IT'  
    });  
  
    const emails = [];  
  
    for (let employee of itEmployees) {  
        if(employee.department === 'IT') {  
            emails.push(employee.email)  
        }  
    }  
    return emails;  
}
```



```
function getItEmails(employees) {  
    const itEmployees = employees.filter((employee) => {  
        return employee.department === 'IT'  
    });  
  
    const emails = [];  
  
    for (let employee of itEmployees) {  
        emails.push(employee.email)  
    }  
    return emails;  
}
```

```
function getItEmails(employees) {  
    const itEmployees = employees.filter((employee) => {  
        return employee.department === 'IT'  
    });  
  
    const emails = [];  
  
    for (let employee of itEmployees) {  
        emails.push(employee.email)  
    }  
    return emails;  
}
```

```
function getItEmails(employees) {  
    const itEmployees = employees.filter((employee) => {  
        return employee.department === 'IT'  
    });  
  
    const emails = [];  
  
    for (let employee of itEmployees) {  
        emails.push(employee.email)  
    }  
    return emails;  
}
```



map!



```
function getItEmails(employees) {  
    const itEmployees = employees.filter((employee) => {  
        return employee.department === 'IT'  
    });  
  
    const emails = itEmployees.map((employee) => {  
        return employee.email;  
    });  
  
    return emails;  
}
```

# THINGS THAT I HATE

- 1. For Loops**
- 2. Conditionals**
- 3. Temparial Variables**

```
function getItEmails(employees) {  
    const itEmployees = employees.filter((employee) => {  
        return employee.department === 'IT'  
    });  
  
    const emails = itEmployees.map((employee) => {  
        return employee.email;  
    });  
  
    return emails;  
}
```

```
function getItEmails(employees) {  
    const itEmployees = employees.filter((employee) => {  
        return employee.department === 'IT'  
    });  
  
    const emails = itEmployees.map((employee) => {  
        return employee.email;  
    });  
  
    return emails;  
}
```

```
function getItEmails(employees) {  
    const itEmployees = employees.filter((employee) => {  
        return employee.department === 'IT'  
    });  
  
    const emails = itEmployees.map((employee) => {  
        return employee.email;  
    });  
  
    return emails;  
}
```

```
function getItEmails(employees) {  
    const itEmployees = employees.filter((employee) => {  
        return employee.department === 'IT'  
    });  
  
    return itEmployees.map((employee) => {  
        return employee.email;  
    });  
  
}
```

```
function getItEmails(employees) {  
    return employees.filter((employee) => {  
        return employee.department === 'IT'  
    }).map((employee) => {  
        return employee.email;  
    });  
  
}
```

## <Exercise 4 />

Rewrite the function using  
higher order functions.



# Thanks for listening

<http://bit.ly/feedback4dave>



lichtsteinerdev



davidlichtsteiner



lichtsteinerdev



lichtsteinerdev