ERHVERVSAKADEMI AARHUS

Computer languages. Local State variables in JS and Scheme.

Sila.

More programs from Alyssa P. Hacker … (A person familiar to readers of “Structure and Interpretation of Computer Programs” by Abelson and Sussman).

A function for withdrawing money:

Chapter 3 in SICP, p. 222 compared to how it looked in JS much later.

The straight-forward declaration:

```
(define balance 100)

(define (withdraw amount)
   (if (>= balance amount)
     (begin (set! balance (- balance amount))
            balance)
     "Insufficient funds"))
```

```
let balance = 100;

function withdraw(amount) {
   if (balance >= amount) {
      balance = balance - amount;
      return balance;
   } else {
      return "Insufficient funds";
   }
}
```

With a local state variable:

```
(define new-withdraw
  (let ((balance 100))
    (lambda (amount)
      (if (>= balance amount)
        (begin (set! balance (- balance
amount))
               balance)
        "Insufficient funds"))))
```

```
function make_withdraw_100() {
    let balance = 100;
    return amount => {
        if (balance >= amount) {
            balance = balance - amount;
            return balance;
        } else {
            return "Insufficient funds";
        }
    };
}
const new_withdraw = make_withdraw_100();
```

With initialization of the balance:

```
(define (make-withdraw balance)
   (lambda (amount)
      (if (>= balance amount)
         (begin (set! balance (- balance
amount))
                balance)
         "Insufficient funds")))
```

```
function make_withdraw(balance) {
    return amount => {
        if (balance >= amount) {
            balance = balance - amount;
            return balance;
        } else {
            return "Insufficient funds";
        }
    };
}
```

Use:

```
(define W1 (make-withdraw 100))
```

```
const W1 = make_withdraw(100);
```

In Scheme:

First, test the new-withdraw function. Does it work as expected?

Then try the more sophisticated make-withdraw.

Make 2 accounts, based on make-withdraw, initialized to 100, and withdraw 60 from both of them, then 30 from one and 50 from the other. What happens? Try it.