JS

High Order Functions

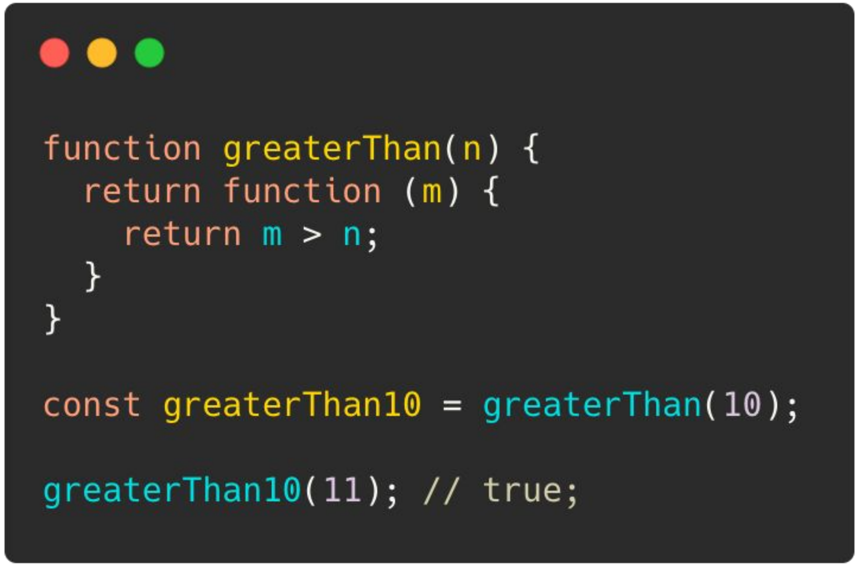
Abraham Berzunza

What is H. O. F ?

They’re functions that **operate on other functions**, either by taking them as arguments or by returning them.

Use case # 1

Create new functions ...



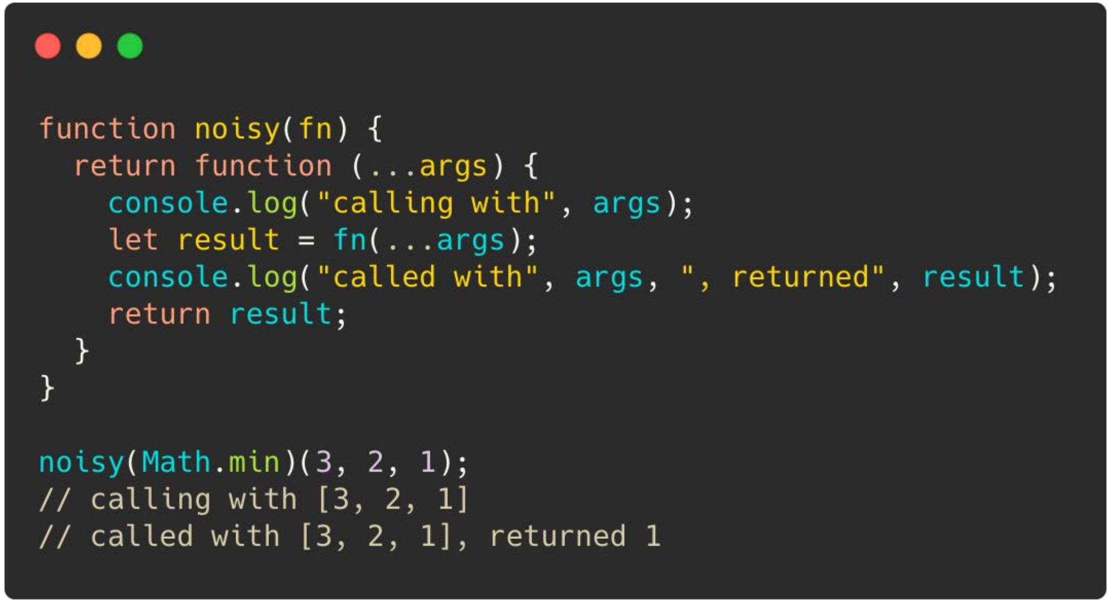
```
function greaterThan(n) {
  return function (m) {
    return m > n;
  }
}

const greaterThan10 = greaterThan(10);

greaterThan10(11); // true;
```

Use case # 2

Change existent functions ...

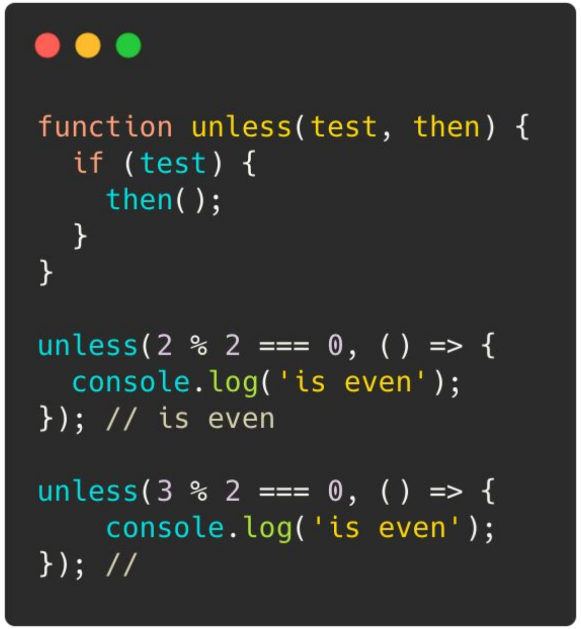


```
function noisy(fn) {
  return function (...args) {
    console.log("calling with", args);
    let result = fn(...args);
    console.log("called with", args, ", returned", result);
    return result;
  }
}

noisy(Math.min)(3, 2, 1);
// calling with [3, 2, 1]
// called with [3, 2, 1], returned 1
```

Use case # 3

create functions that provide new types of control flow ...



```
function unless(test, then) {
  if (test) {
    then();
  }
}

unless(2 % 2 === 0, () => {
  console.log('is even');
}); // is even

unless(3 % 2 === 0, () => {
    console.log('is even');
}); //
```

H.O.F Built in

- forEach
- map
- filter
- reduce

JS

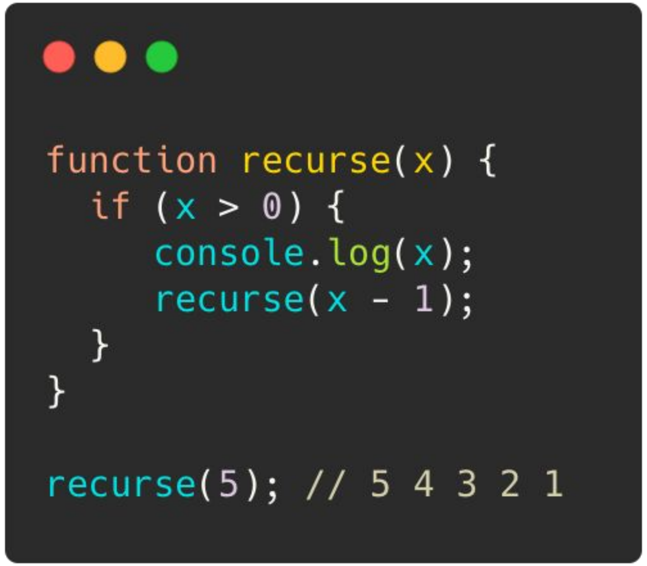
Recursion

Abraham Berzunza

What is it?

The act of a function **calling itself.**

Example



```
function recurse(x) {
  if (x > 0) {
     console.log(x);
     recurse(x - 1);
  }
}

recurse(5); // 5 4 3 2 1
```

Activity

Solve exercise 3 included in the following link:

https://github.com/abrahamBerzunza/js-training-program/blob/master/exercises/exercise-3.js

Make sure all test cases pass and send your **pull request.**

MDN

https://developer.mozilla.org/en-US/docs/Glossary/Recursion

Eloquent JS

https://eloquentjavascript.net/05_higher_order.html

JS