EXAMPLE 39: NAMED AND ANONYMOUS FUNCTION LITERALS

EXAMPLE 39: NAMED AND ANONYMOUS FUNCTION LITERALS

WE HAVE ALREADY ENCOUNTERED ANONYMOUS FUNCTION LITERALS

(TWICE)

WE HAVE ALREADY ENCOUNTERED ANONYMOUS FUNCTION LITERALS (TWICE)

```
var anonFunction = function() {
    console.log("I am an anonymous function 
literal!");
  }
```

```
function Rectangle(l,b) {
    this.length = l;
    this.breadth = b;
    this.area = function() {
          console.log("I am an anonymous functio
  literal!");
          return this.length * this.breadth;
    };
}
```

WE HAVE ALREADY ENCOUNTERED ANONYMOUS FUNCTION LITERALS

(TWICE)

```
var anonFunction = function() {
    console.log("I am an anonymous function literal!");
}
```

```
function Rectangle(l,b) {
    this.length = l;
    this.breadth = b;
    this.area = function() {
        console.log("I am an anonymous function literal!");
        return this.length * this.breadth;
    };
}
```

WE HAVE ALREADY ENCOUNTERED ANONYMOUS FUNCTION LITERALS

(TWICE)

```
var anonFunction = function() {
  console.log("I am an anonymous function 
literal!");
}
```

```
function Rectangle(l,b) {
  this.length = l;
  this.breadth = b;
  this.area = function() {
     console.log("I am an anonymous function 
literal!");
     return this.length * this.breadth;
  };
}
```

WE HAVE ALREADY ENCOUNTERED ANONYMOUS FUNCTION LITERALS

(TWICE)

```
var anonFunction = function() {
  console.log("I am an anonymous function literal!");
}
```

```
function Rectangle(l,b) {
  this.length = l;
  this.breadth = b;
  this.area = function() {
     console.log("I am an anonymous function literal!");
     return this.length * this.breadth;
  };
}
```

OBJECT PROPERTIES ARE ALMOST

WE HAVE ALREADY ENCOUNTERED ANONYMOUS FUNCTION LITERALS

(TWICE)

BUT NOT ALL FUNCTION LITERALS ARE ANONYMOUS!

BUT NOT ALL FUNCTION LITERALS ARE ANONYMOUS!

```
var namedFunction = function foo(x) {
  console.log("I am a named function!" -
  if (x == 1){
   foo(2);
  }
}
```

BUT NOT ALL FUNCTION LITERALS ARE ANONYMOUS!

```
var namedFunction = function foo(x) {
  console.log("I am a named function!"
  if (x == 1) {
    foo(2);
  }
}
```

HERE THE FUNCTION LITERAL IS NAMED **`foo`**, BUT ASSIGNED TO A VARIABLE NAMED **`namedFunction`**

BUT NOT ALL FUNCTION LITERALS ARE ANONYMOUS!

```
var namedFunction = function foo(x) {
  console.log("I am a named function!"
  if (x == 1) {
   foo(2);
  }
}
```

HERE THE FUNCTION LITERAL IS NAMED **foo**, BUT ASSIGNED TO A VARIABLE NAMED

namedFunction

BUT NOT ALL FUNCTION LITERALS ARE ANONYMOUS!

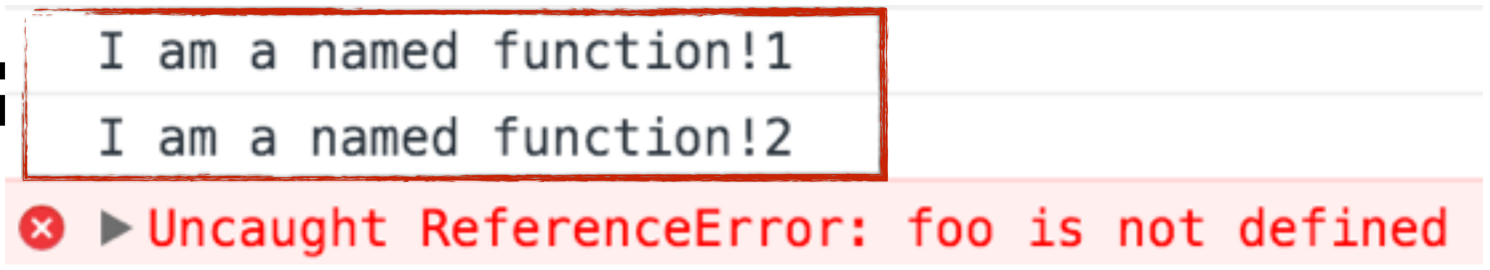
```
var namedFunction = function foo(x) {
    console.log("I am a named function!"
    if (x == 1) {
        foo(2);
    }
}
```

HERE THE FUNCTION LITERAL IS NAMED `foo`, BUT ASSIGNED TO A VARIABLE NAMED `namedFunction`

NOW YOU CAN'T CALL THE FUNCTION AS `foo(x)`, ONLY AS `namedFunction(x)`

NOW YOU CAN'T CALL THE FUNCTION AS `foo(x)`, ONLY AS `namedFunction(x)`

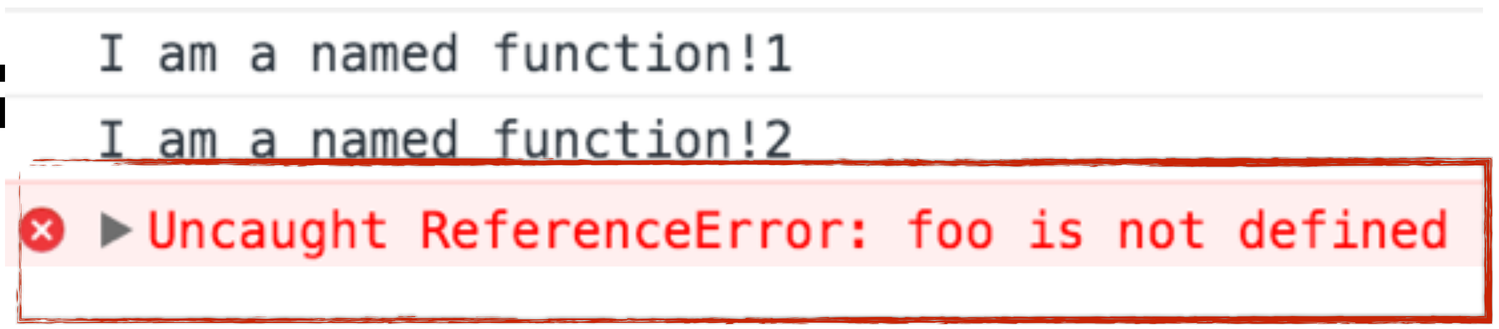
```
namedFunction(1);
foo();
```



```
I am a named function!1
I am a named function!2
▶ Uncaught ReferenceError: foo is not defined
```

NOW YOU CAN'T CALL THE FUNCTION AS `foo(x)`, ONLY AS `namedFunction(x)`

```
namedFunction(1);
foo();
```



```
I am a named function!1
I am a named function!2
▶ Uncaught ReferenceError: foo is not defined
```

ERRM..OK..WHAT'S THE POINT OF HAVING A NAME IF YOU CAN'T USE IT THEN?

ERRM..OK..WHAT'S THE POINT OF HAVING A NAME IF YOU CAN'T USE IT THEN?

RECURSION!

YOU CAN USE THE NAME OF THE FUNCTION EXPRESSION TO REFER TO ITSELF INSIDE THE BODY OF THE FUNCTION

RECURSION!

YOU CAN USE THE NAME OF THE FUNCTION EXPRESSION TO REFER TO ITSELF INSIDE THE BODY OF THE FUNCTION

```
var namedFunction = function foo(x) {
  console.log("I am a named function!"
  if (x == 1){
    foo(2);
  }
}
```

RECURSION!

YOU CAN USE THE NAME OF THE FUNCTION EXPRESSION TO REFER TO ITSELF INSIDE THE BODY OF THE FUNCTION

```
var namedFunction = function foo(x) {
  console.log("I am a named function!"
  if (x == 1){
    foo(2);
  }
}
```

WE HAVE ALREADY ENCOUNTERED ANONYMOUS FUNCTION LITERALS

(TWICE)

BUT NOT ALL FUNCTION LITERALS ARE ANONYMOUS!

RECURSION!