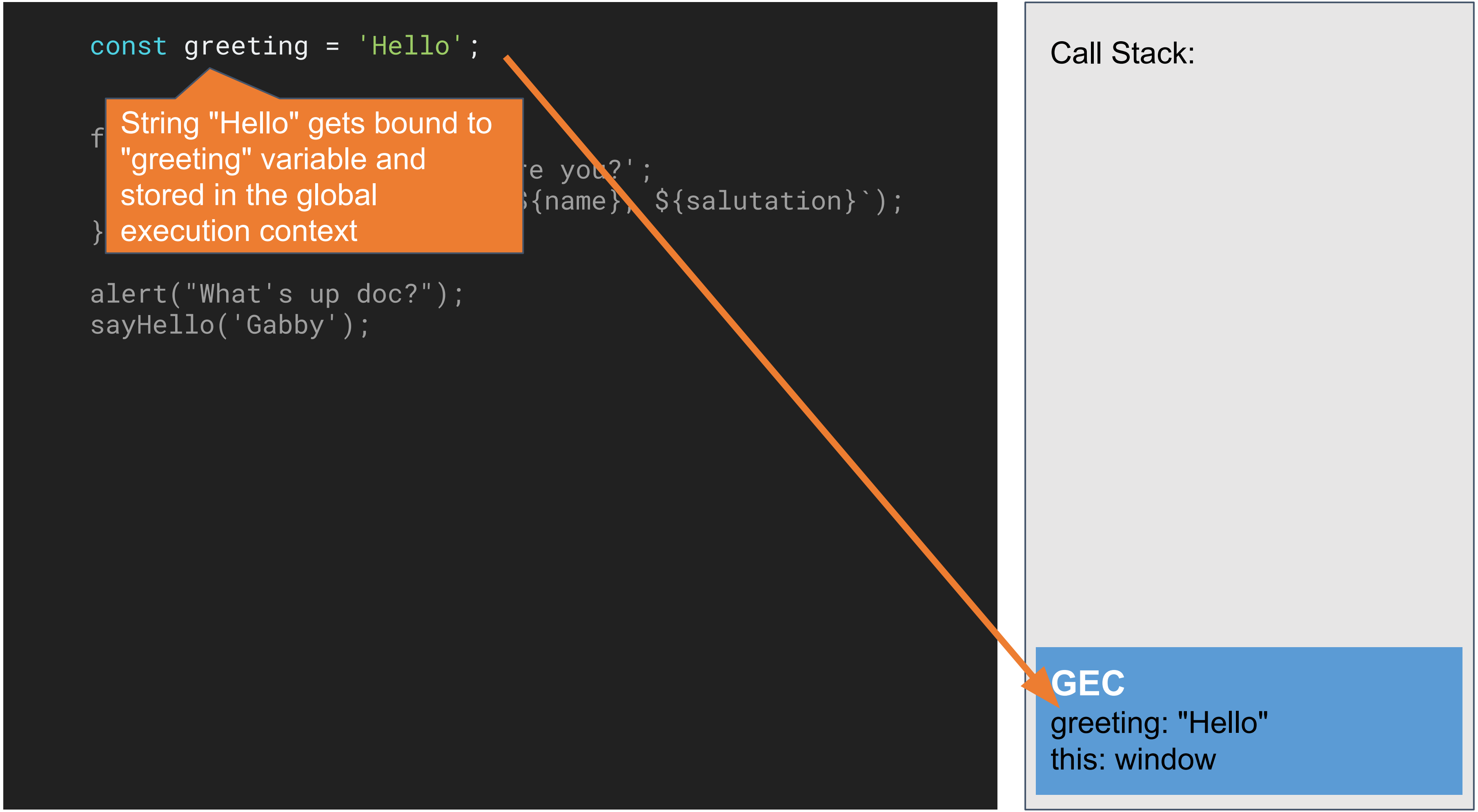
Execution Context

Web Development Boot Camp

Unit 17

```
const greeting = 'Hello';

function sayHello(name) {
  const salutation = 'how are you?';
  console.log(`${greeting} ${name}, ${salutation}`);
}

alert("What's up doc?");
sayHello('Gabby');
```

Call Stack:

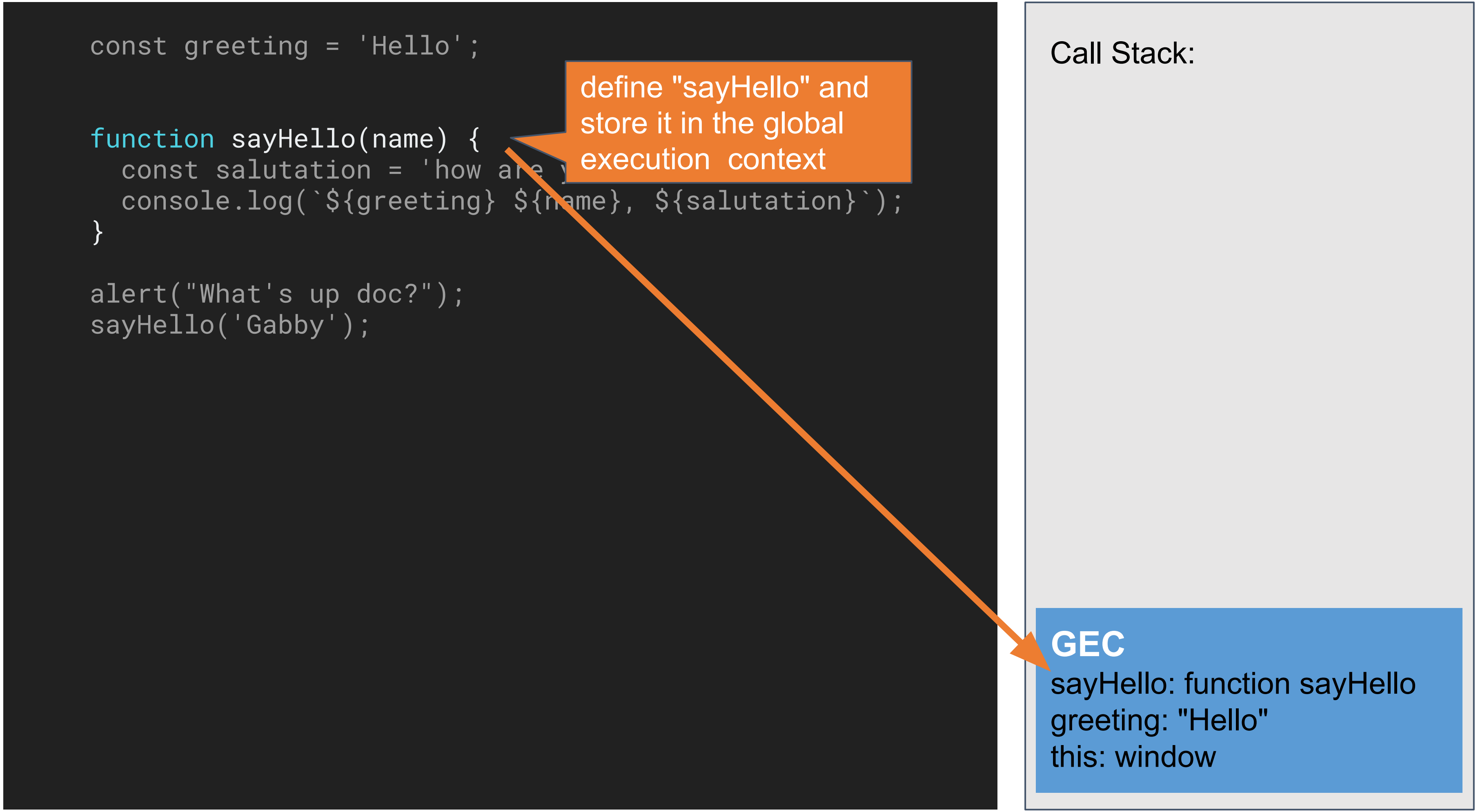


```
const greeting = 'Hello';


f
                                e you?';
                               {name}, ${salutation}`);
}

alert("What's up doc?");
sayHello('Gabby');
```

String "Hello" gets bound to "greeting" variable and stored in the global execution context

Call Stack:

GEC
greeting: "Hello"
this: window

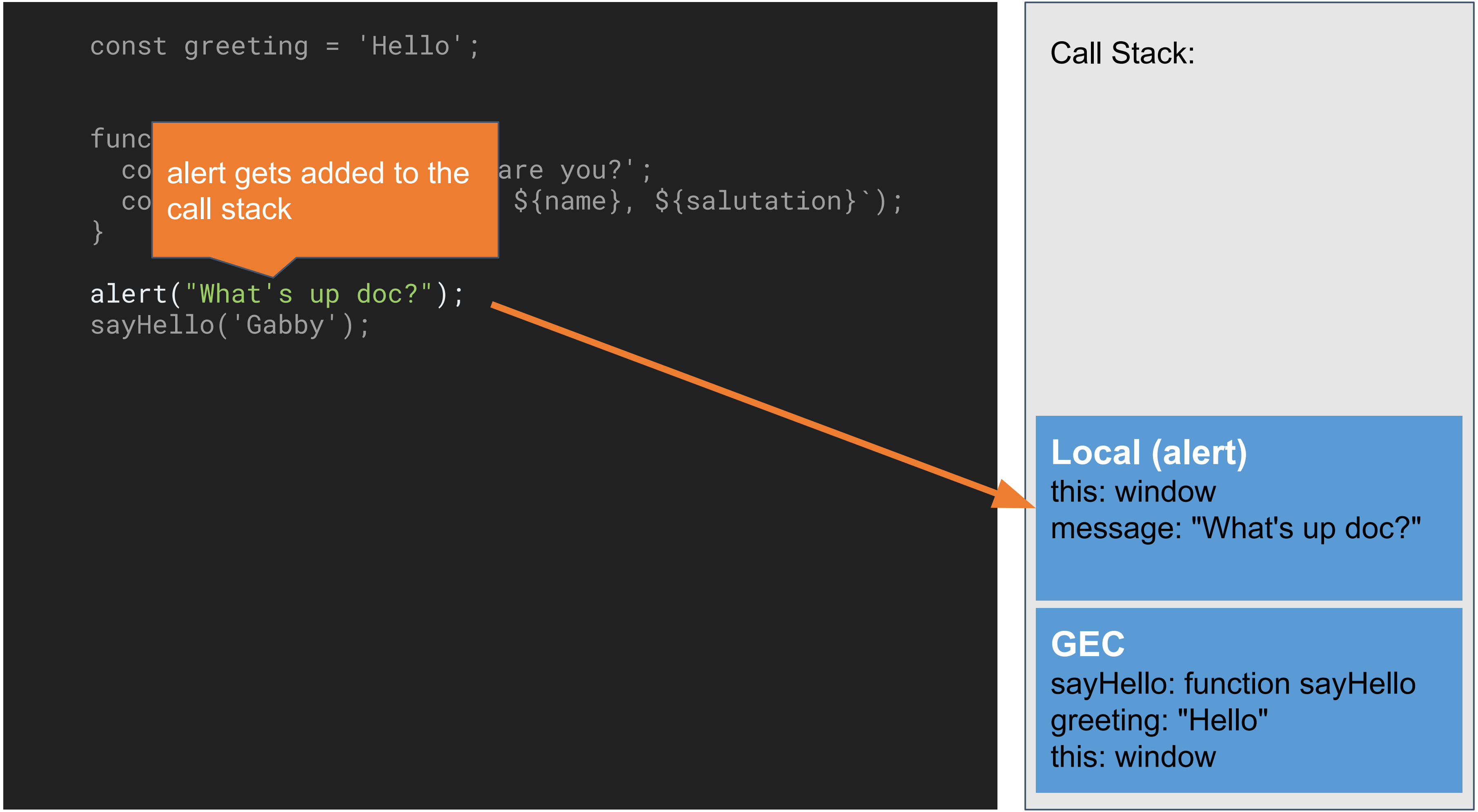


```
const greeting = 'Hello';


function sayHello(name) {
  const salutation = 'how are y
  console.log(`${greeting} ${name}, ${salutation}`);
}

alert("What's up doc?");
sayHello('Gabby');
```

define "sayHello" and store it in the global execution context

Call Stack:

GEC
sayHello: function sayHello
greeting: "Hello"
this: window



```
const greeting = 'Hello';


func
  co                       are you?';
  co                        ${name}, ${salutation}`);
}

alert("What's up doc?");
sayHello('Gabby');
```

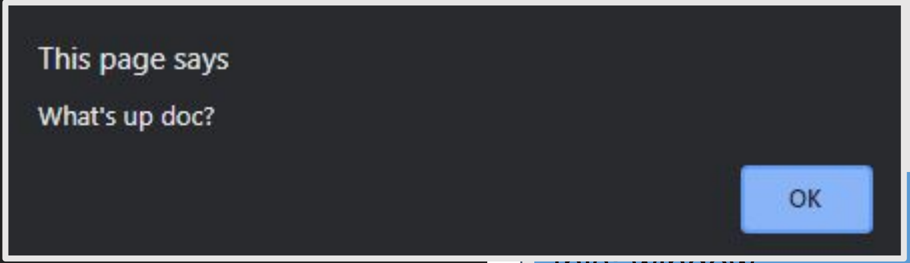
alert gets added to the call stack

Call Stack:

Local (alert)
this: window
message: "What's up doc?"

GEC
sayHello: function sayHello
greeting: "Hello"
this: window

```
const greeting = 'Hello';


function sayHello(name) {
  const salutation = 'how are you?';
  console.log(`${greeting} ${name}, ${salutation}`);
}

alert("What's up doc?");
sayHello('Gabby');
```



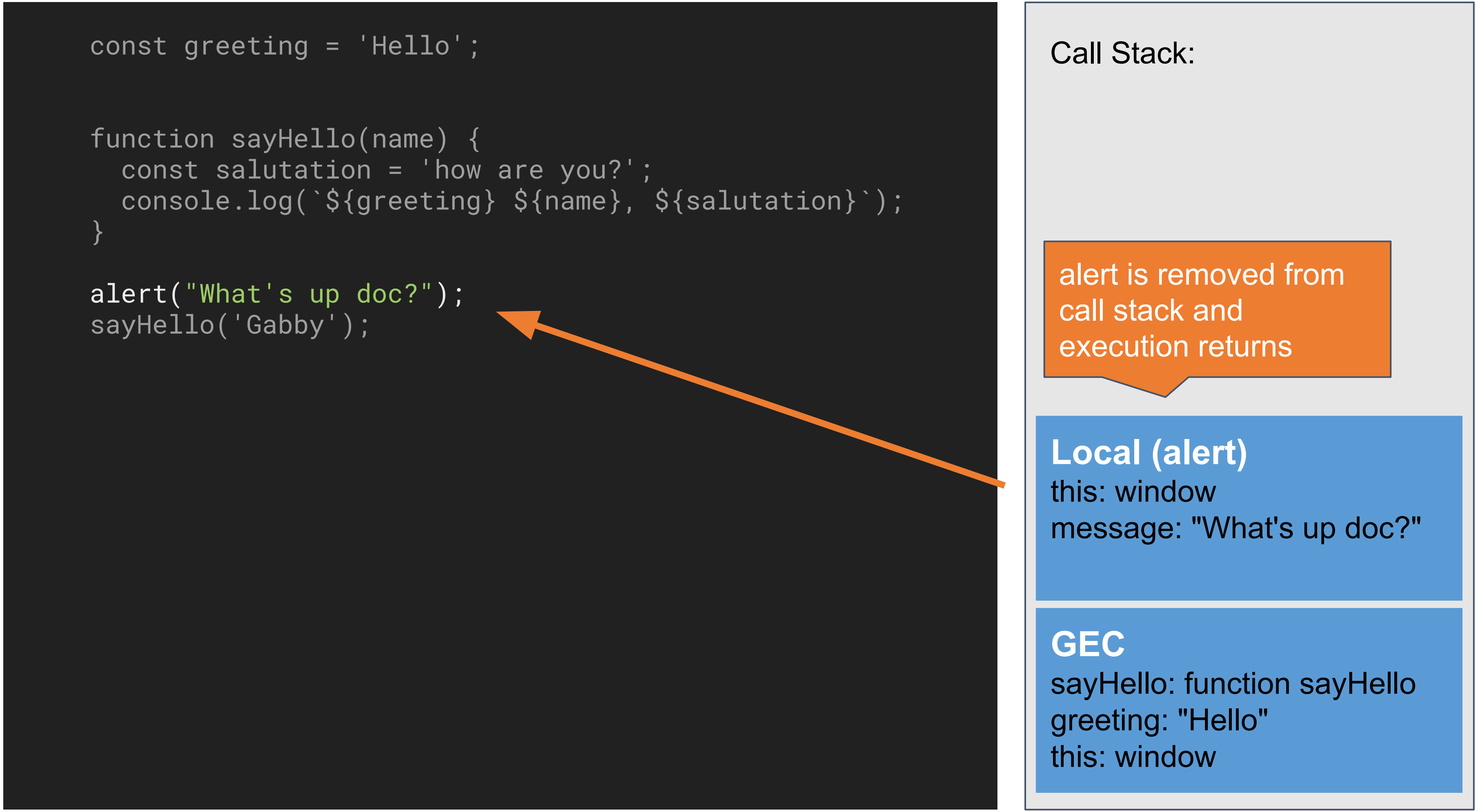
This page says

What's up doc?

Call Stack:

this: window
message: "What's up doc?"

GEC
sayHello: function sayHello
greeting: "Hello"
this: window



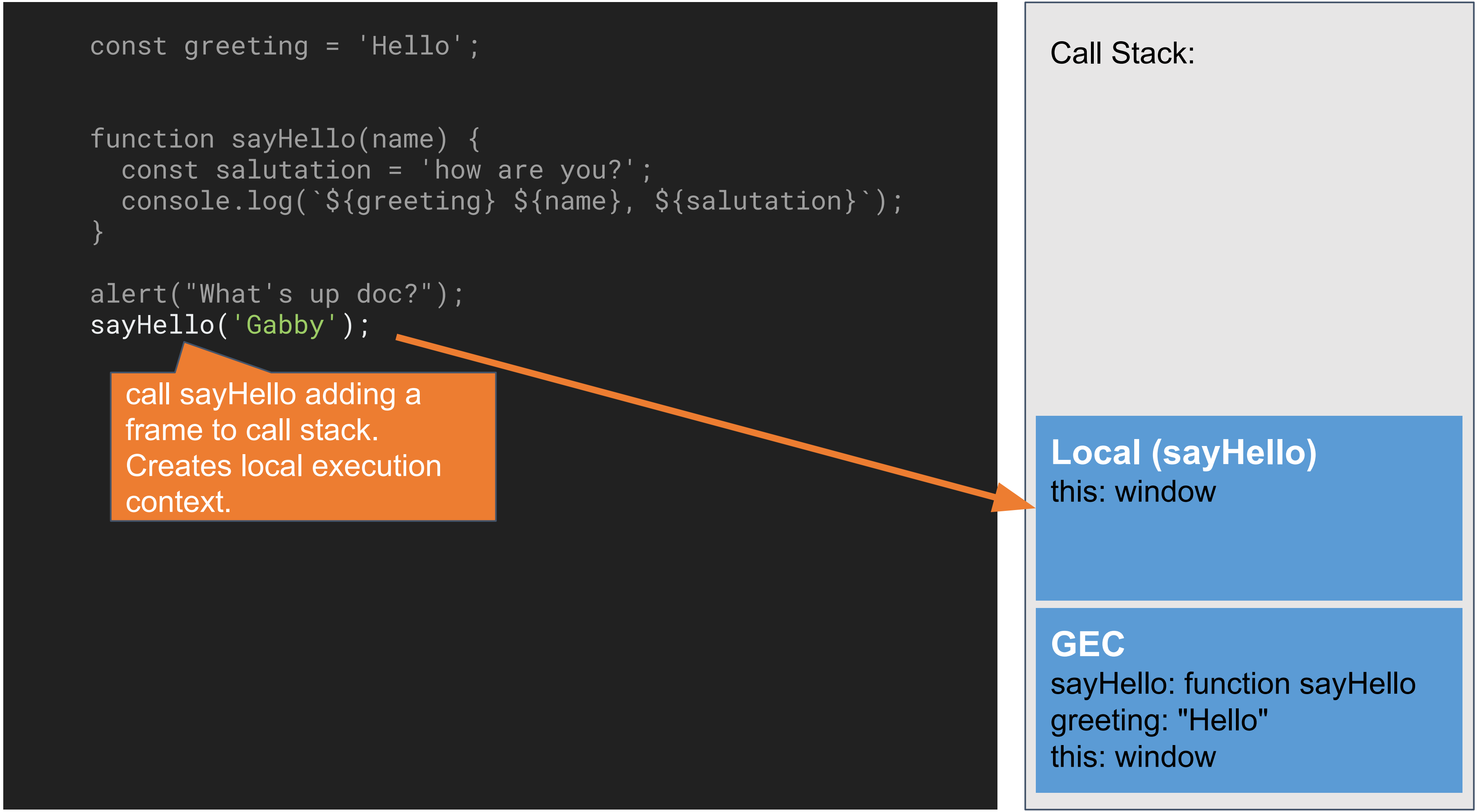
```
const greeting = 'Hello';


function sayHello(name) {
  const salutation = 'how are you?';
  console.log(`${greeting} ${name}, ${salutation}`);
}

alert("What's up doc?");
sayHello('Gabby');
```

Call Stack:

alert is removed from call stack and execution returns

Local (alert)
this: window
message: "What's up doc?"

GEC
sayHello: function sayHello
greeting: "Hello"
this: window



```
const greeting = 'Hello';


function sayHello(name) {
  const salutation = 'how are you?';
  console.log(`${greeting} ${name}, ${salutation}`);
}

alert("What's up doc?");
sayHello('Gabby');
```

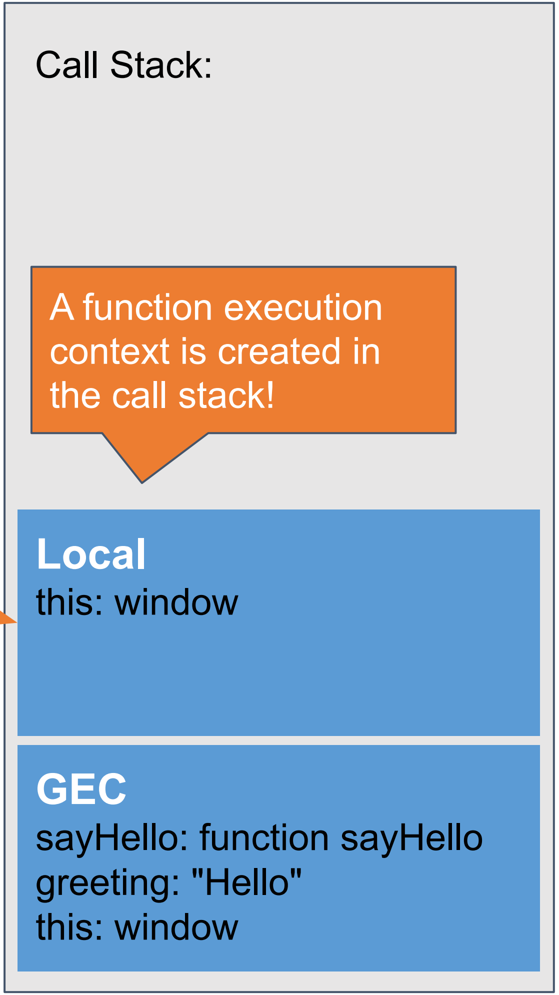
call sayHello adding a frame to call stack. Creates local execution context.

Call Stack:

Local (sayHello)
this: window

GEC
sayHello: function sayHello
greeting: "Hello"
this: window

```
const greeting = 'Hello';


function sayHello(name) {
  const salutation = 'how are you?';
  console.log(`${greeting} ${name}, ${salutation}`);
}

alert("What's up doc?");
sayHello('Gabby');
```

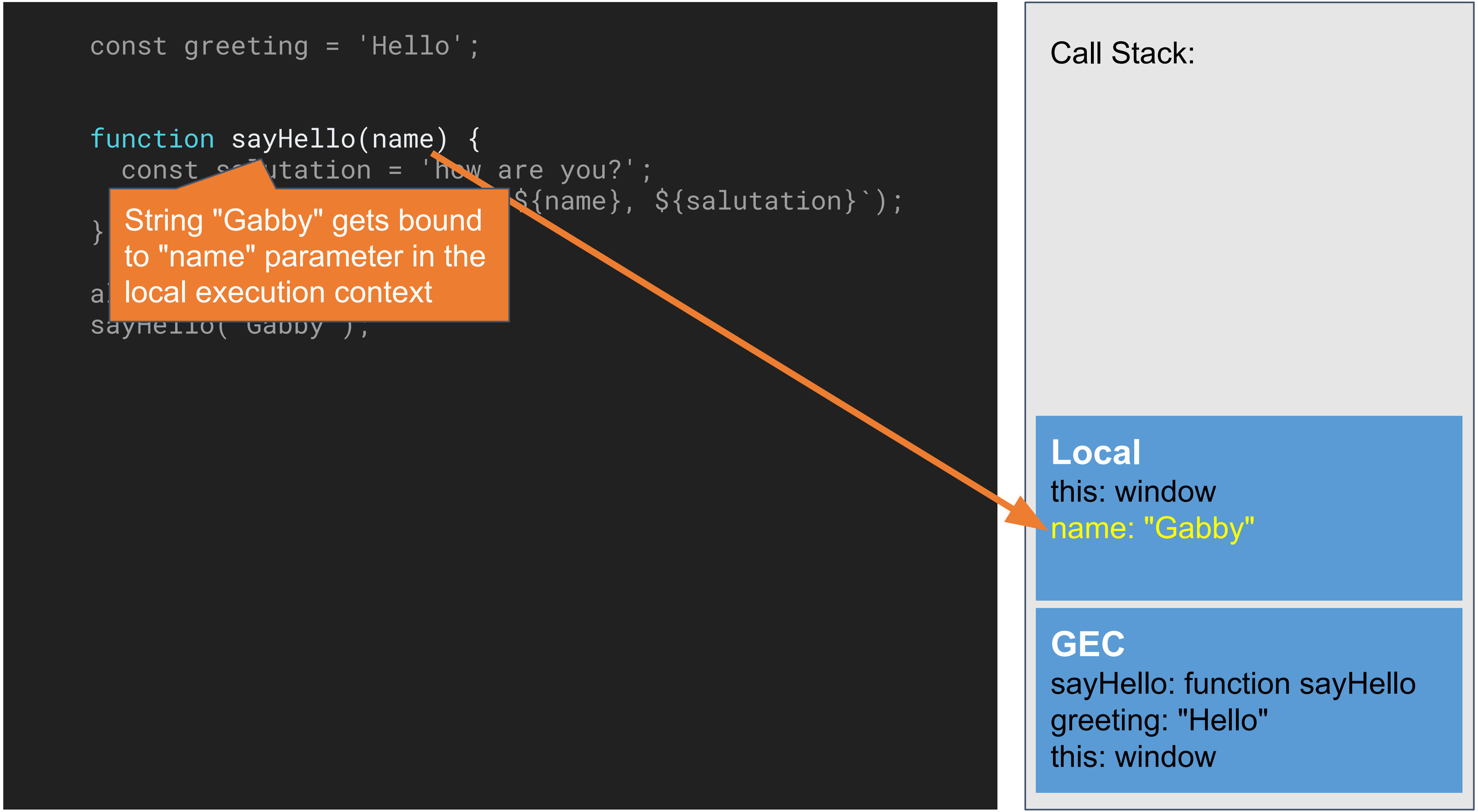


Call Stack:

A function execution context is created in the call stack!

Local
this: window

GEC
sayHello: function sayHello
greeting: "Hello"
this: window



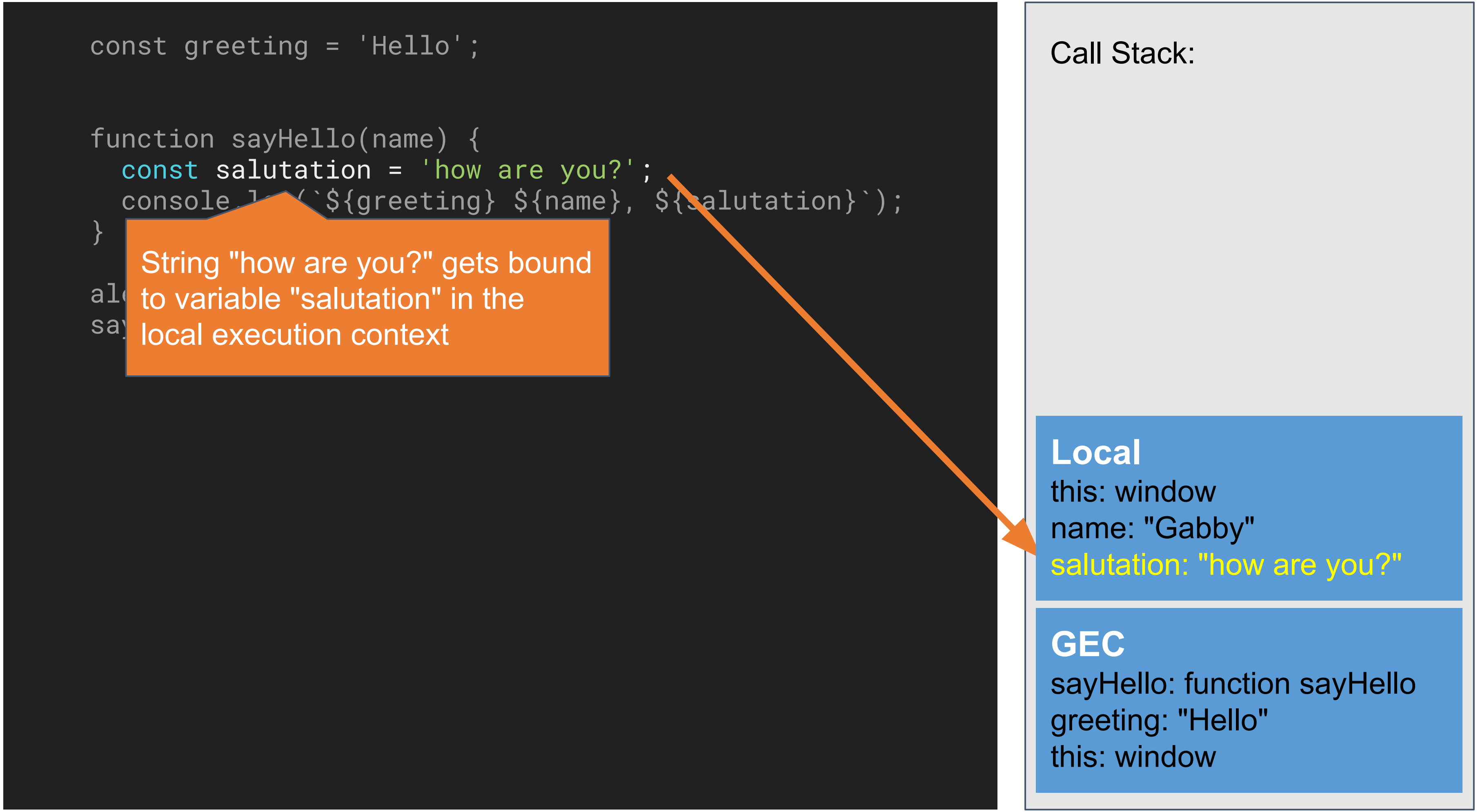
```
const greeting = 'Hello';


function sayHello(name) {
  const salutation = 'how are you?';
                            ${name}, ${salutation}`);
}

a
sayHello('Gabby');
```

String "Gabby" gets bound to "name" parameter in the local execution context

Call Stack:

Local
this: window
name: "Gabby"

GEC
sayHello: function sayHello
greeting: "Hello"
this: window



```
const greeting = 'Hello';


function sayHello(name) {
  const salutation = 'how are you?';
  console.log(`${greeting} ${name}, ${salutation}`);
}

al
say
```

String "how are you?" gets bound to variable "salutation" in the local execution context

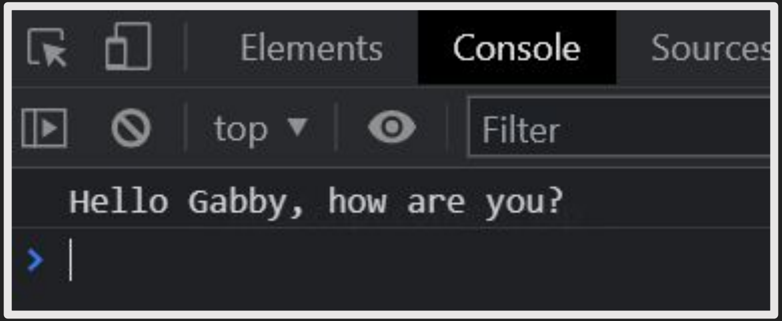
Call Stack:

Local
this: window
name: "Gabby"
salutation: "how are you?"

GEC
sayHello: function sayHello
greeting: "Hello"
this: window

```
const greeting = 'Hello';


function sayHello(name) {
  const salutation = 'how are you?';
  console.log(`${greeting} ${name}, ${salutation}`);
}

alert("What's up
sayHello('Gabby'
```

"greeting" is accessible from the global execution context.



```
Hello Gabby, how are you?
>
```

Call Stack:

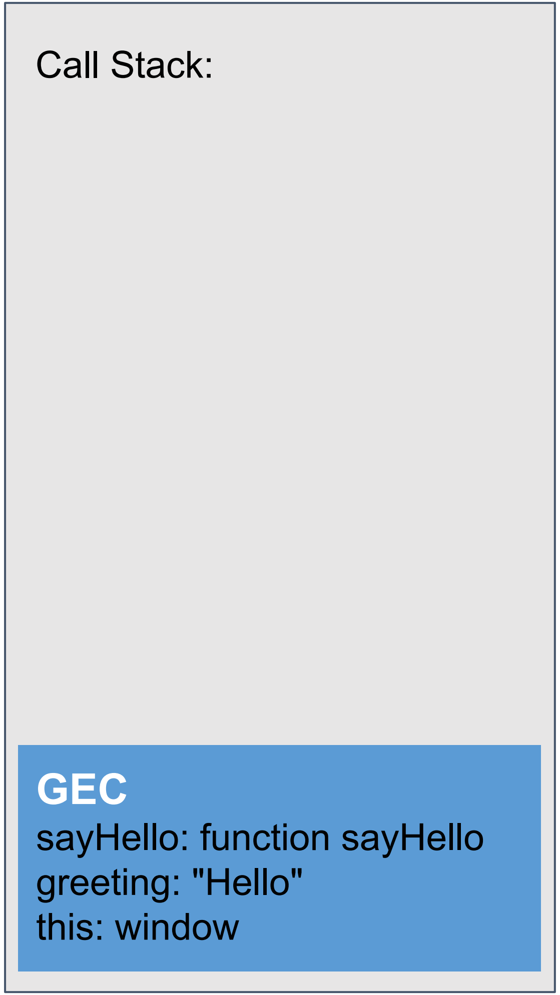
Local
this: window
name: "Gabby"
salutation: "how are you?"

GEC
sayHello: function sayHello
greeting: "Hello"
this: window

```
const greeting = 'Hello';


function sayHello(name) {
  const salutation = 'how are you?';
  console.log(`${greeting} ${name}, ${salutation}`);
}

alert("What's up doc?");
sayHello('Gabby');
```

return from sayHello

no more statements



Call Stack:

GEC
sayHello: function sayHello
greeting: "Hello"
this: window