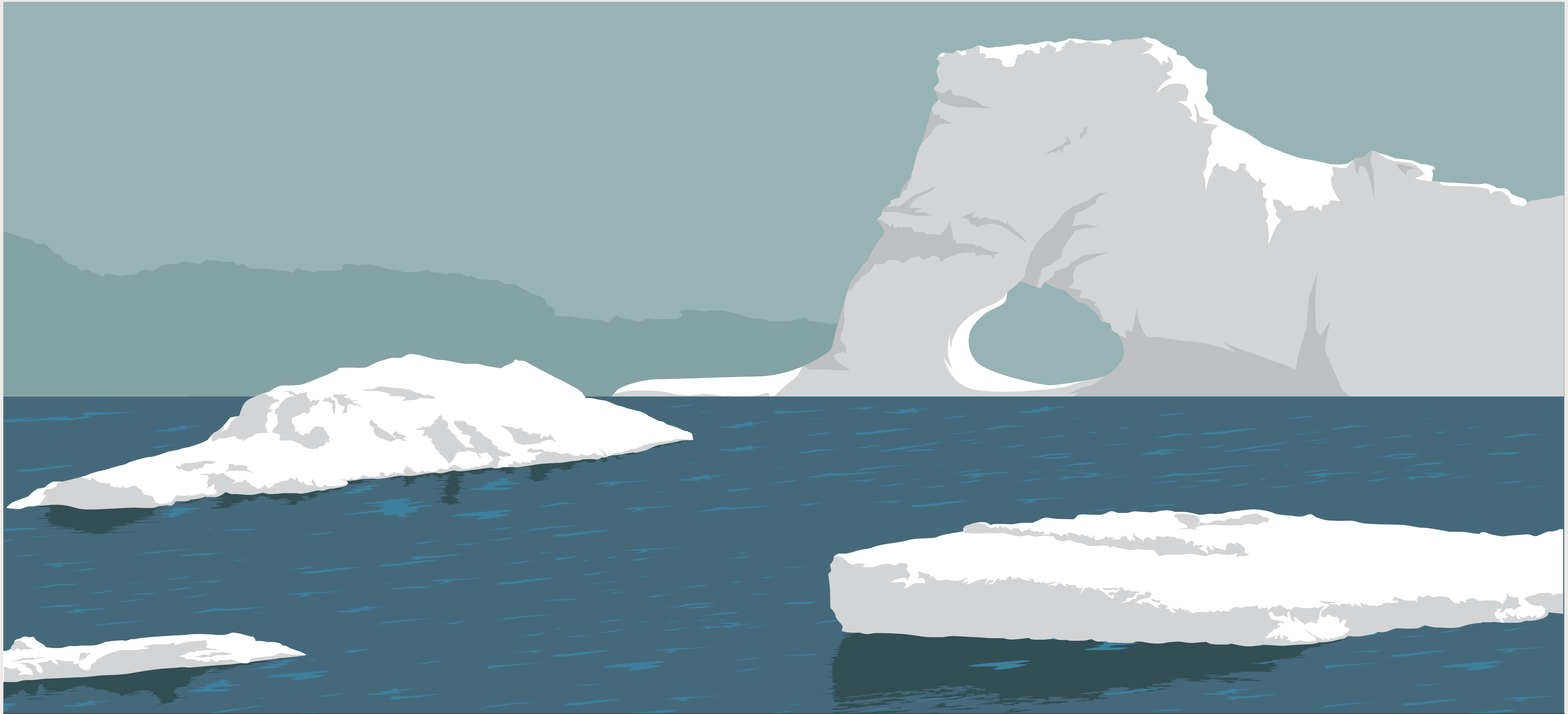


Explore

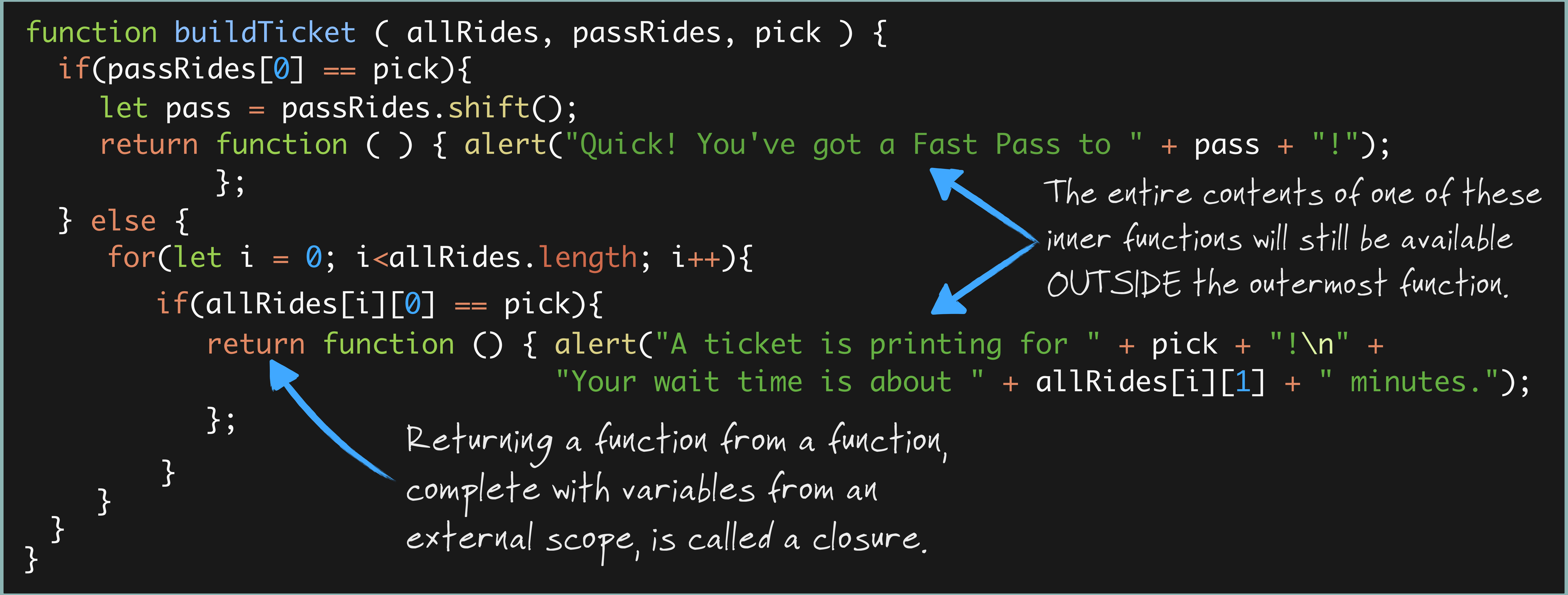
COLD CLOSURES COVE

LEVEL 2

COLD CLOSURES COVE

A PACKAGE DEAL

Guess what? Congratulations! You’ve already made a basic closure!



```
function buildTicket ( allRides, passRides, pick ) {
  if(passRides[0] == pick){
    let pass = passRides.shift();
    return function ( ) { alert("Quick! You've got a Fast Pass to " + pass + "!");
           };
  } else {
     for(let i = 0; i<allRides.length; i++){
        if(allRides[i][0] == pick){
           return function () { alert("A ticket is printing for " + pick + "!\n" +
                                "Your wait time is about " + allRides[i][1] + " minutes.");
           };
        }
    }
  }
}
```

The entire contents of one of these inner functions will still be available OUTSIDE the outermost function.

Returning a function from a function, complete with variables from an external scope, is called a closure.

A PACKAGE DEAL

A closure wraps up an entire environment, binding necessary variables from other scopes.

```
function  testClosure ( ) {

     let x = 4;
     return x;

}
```

← Local Variable only!

```
testClosure();
```

→ 4

```
x;
```

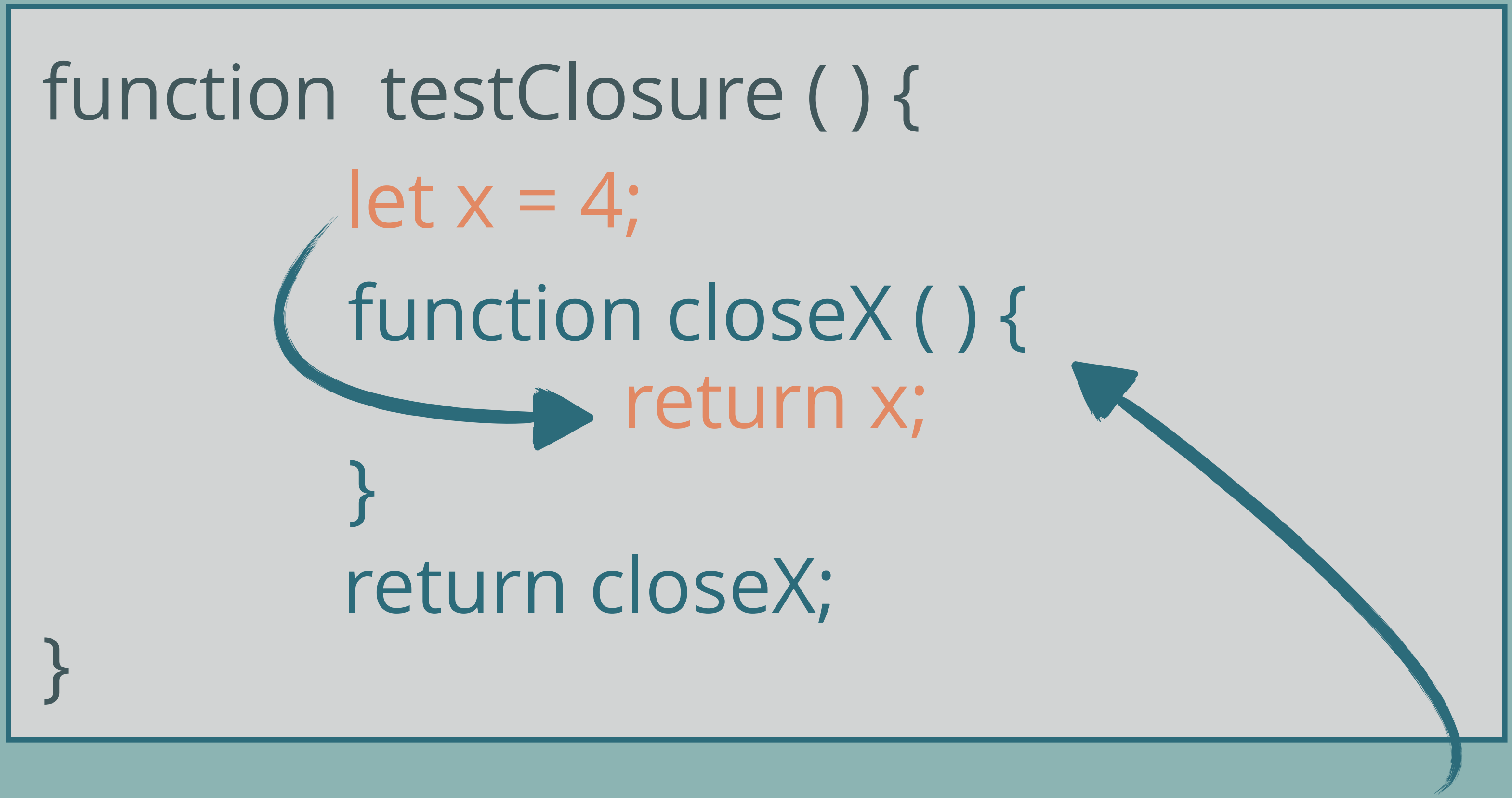
→ undefined

A function's local variables aren't available once the function's scope is closed!

A PACKAGE DEAL

A closure wraps up an entire environment, binding necessary variables from other scopes.

The inner function can access the outer function's variables, because they "feel" like global variables.

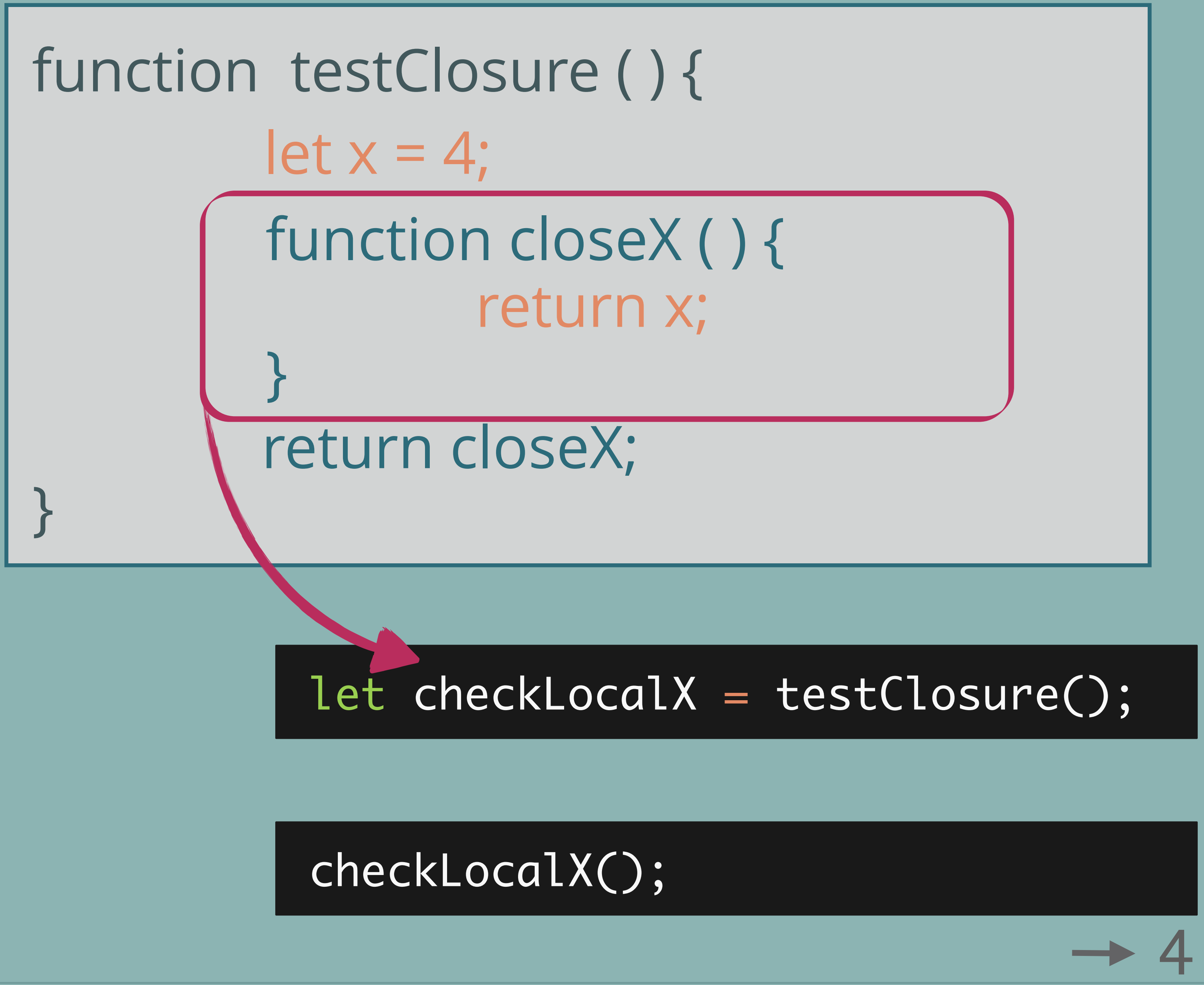


```
function  testClosure ( ) {
          let x = 4;
          function closeX ( ) {
                return x;
          }
          return closeX;
}
```

Notice x does not need to be "stored" anywhere in closeX, not even as a parameter!

A PACKAGE DEAL

A closure wraps up an entire environment, binding necessary variables from other scopes.



```
function  testClosure ( ) {
              let x = 4;
              function closeX ( ) {
                     return x;
              }
              return closeX;
}
```

```
let checkLocalX = testClosure();
```

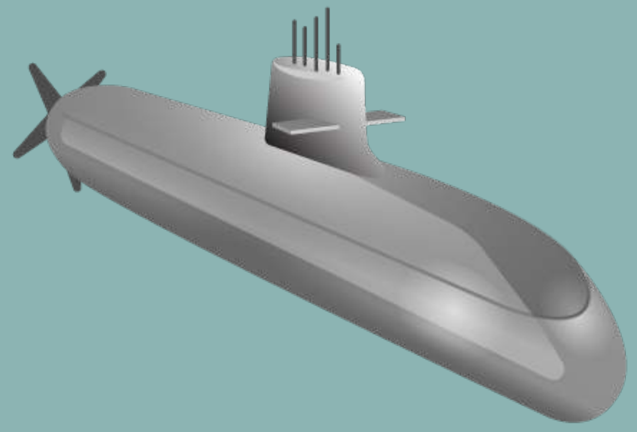
```
checkLocalX();
```

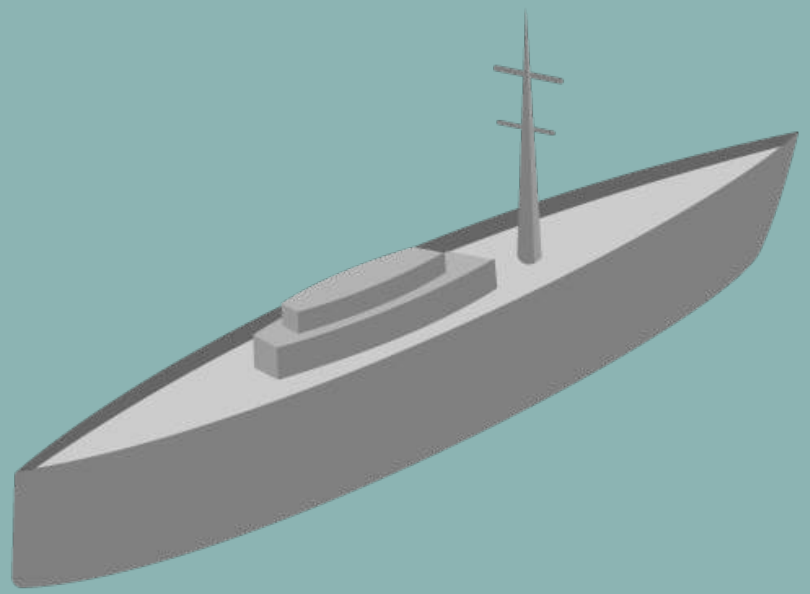
→ 4

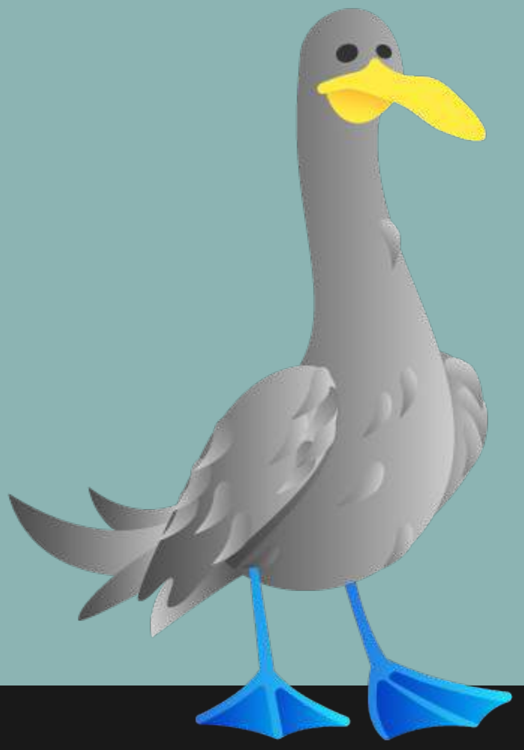
Even though testClosure has finished operating, its local variable is now bound within checkLocalX.

CLOSURES HELP IN FUNCTION "CONSTRUCTION ZONES"

A closure can make the creation of very similar functions ultra-efficient.







```
function buildCoveTicketMaker( transport ) {
         return function ( name ) {
            alert("Here is your transportation ticket via the " + transport + ".\n" +
                  "Welcome to the Cold Closures Cove, " + name + "!");
         }
}
```

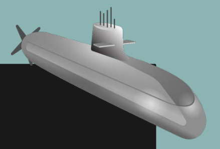
CLOSURES HELP IN FUNCTION “CONSTRUCTION ZONES”

A closure can make the creation of very similar functions ultra-efficient.

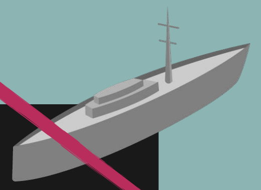
```
function buildCoveTicketMaker( transport ) {

         return function ( name ) {
            alert("Here is your transportation ticket via the " + transport + ".\n" +
                  "Welcome to the Cold Closures Cove, " + name + "!");
         }
}
```

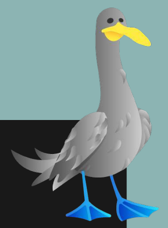
```
let getSubmarineTicket = buildCoveTicketMaker("Submarine");
```



```
let getBattleshipTicket = buildCoveTicketMaker("Battleship");
```



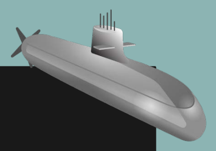
```
let getGiantSeagullTicket = buildCoveTicketMaker("Giant Seagull");
```



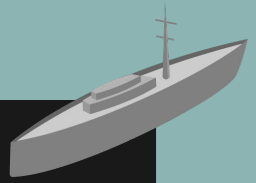
We give buildCoveTicketMaker the mode of transportation, which is closed into the returned anonymous function.

CLOSURES HELP IN FUNCTION "CONSTRUCTION ZONES"

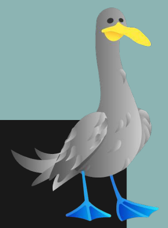
A closure can make the creation of very similar functions ultra-efficient.



```
let getSubmarineTicket = buildCoveTicketMaker("Submarine");
```



```
let getBattleshipTicket = buildCoveTicketMaker("Battleship");
```

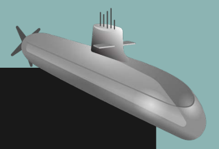


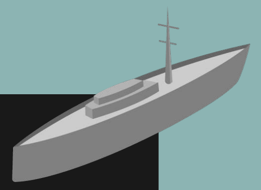
```
let getGiantSeagullTicket = buildCoveTicketMaker("Giant Seagull");
```

CLOSURES HELP IN FUNCTION “CONSTRUCTION ZONES”

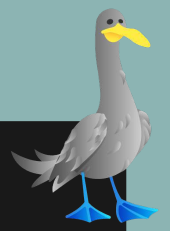
A closure can make the creation of very similar functions ultra-efficient.

```
getSubmarineTicket
```





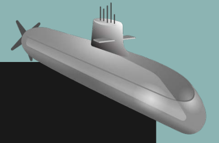
```
getBattleshipTicket
```



```
getGiantSeagullTicket
```

BEWARE: BOUND VARIABLES WON'T BE EVIDENT IN THE STORED FUNCTION

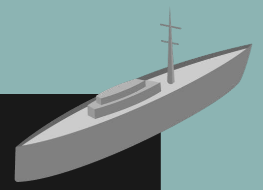
Examining the contents of our new variables doesn't reveal closures.



```
getSubmarineTicket;
```

```
function ( name ) {
   alert("Here is your transportation ticket via the " + transport + ".\n" +
         "Welcome to the Cold Closures Cove, " + name + "!");
}
```

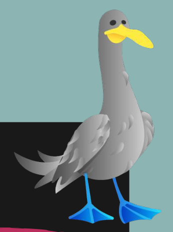
Holds "Submarine"



```
getBattleshipTicket;
```

```
function ( name ) {
   alert("Here is your transportation ticket via the " + transport + ".\n" +
         "Welcome to the Cold Closures Cove, " + name + "!");
}
```

Holds "Battleship"



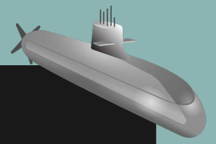
```
getGiantSeagullTicket;
```

```
function ( name ) {
   alert("Here is your transportation ticket via the " + transport + ".\n" +
         "Welcome to the Cold Closures Cove, " + name + "!");
}
```

Holds "Giant Seagull"

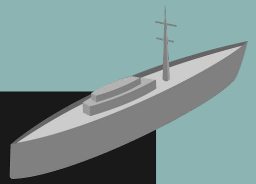
BEWARE: BOUND VARIABLES WON'T BE EVIDENT IN THE STORED FUNCTION

Examining the contents of our new variables doesn't reveal closures.



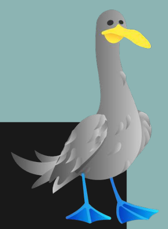
```
getSubmarineTicket;
```

```
function ( name ) {
   alert("Here is your transportation ticket via the " + transport + ".\n" +
         "Welcome to the Cold Closures Cove, " + name + "!");
}
```



```
getBattleshipTicket;
```

```
function ( name ) {
   alert("Here is your transportation ticket via the " + transport + ".\n" +
         "Welcome to the Cold Closures Cove, " + name + "!");
}
```



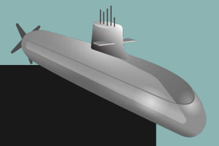
```
getGiantSeagullTicket;
```

```
function ( name ) {
   alert("Here is your transportation ticket via the " + transport + ".\n" +
         "Welcome to the Cold Closures Cove, " + name + "!");
}
```

Until we call any of these functions with a parameter, the name variable is still undefined.

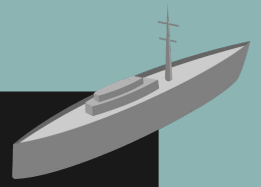
LET'S MAKE SOME TICKETS!

Passing a name to any of our ticket makers will complete our ticket-making process.



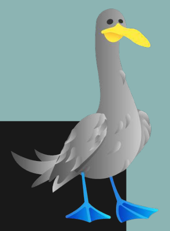
```
getSubmarineTicket;
```

```
function ( name ) {
   alert("Here is your transportation ticket via the " + transport + ".\n" +
         "Welcome to the Cold Closures Cove, " + name + "!");
}
```



```
getBattleshipTicket;
```

```
function ( name ) {
   alert("Here is your transportation ticket via the " + transport + ".\n" +
         "Welcome to the Cold Closures Cove, " + name + "!");
}
```

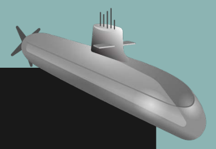


```
getGiantSeagullTicket;
```

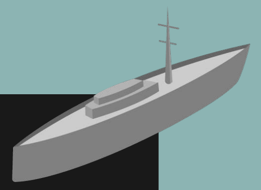
```
function ( name ) {
   alert("Here is your transportation ticket via the " + transport + ".\n" +
         "Welcome to the Cold Closures Cove, " + name + "!");
}
```

LET'S MAKE SOME TICKETS!

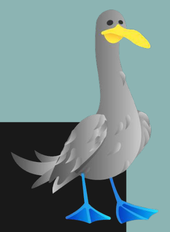
Passing a name to any of our ticket makers will complete our ticket-making process.



```
getSubmarineTicket;
```



```
getBattleshipTicket;
```

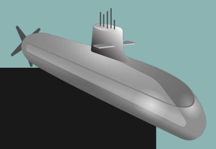


```
getGiantSeagullTicket;
```

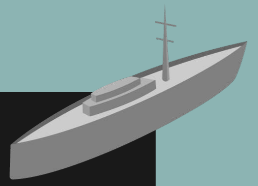
LET'S MAKE SOME TICKETS!

Passing a name to any of our ticket makers will complete our ticket-making process.

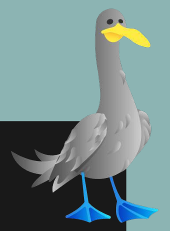
```
getSubmarineTicket("Mario");
```



```
getBattleshipTicket("Luigi");
```

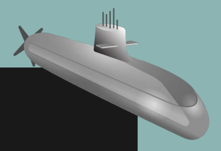


```
getGiantSeagullTicket("Bowser");
```

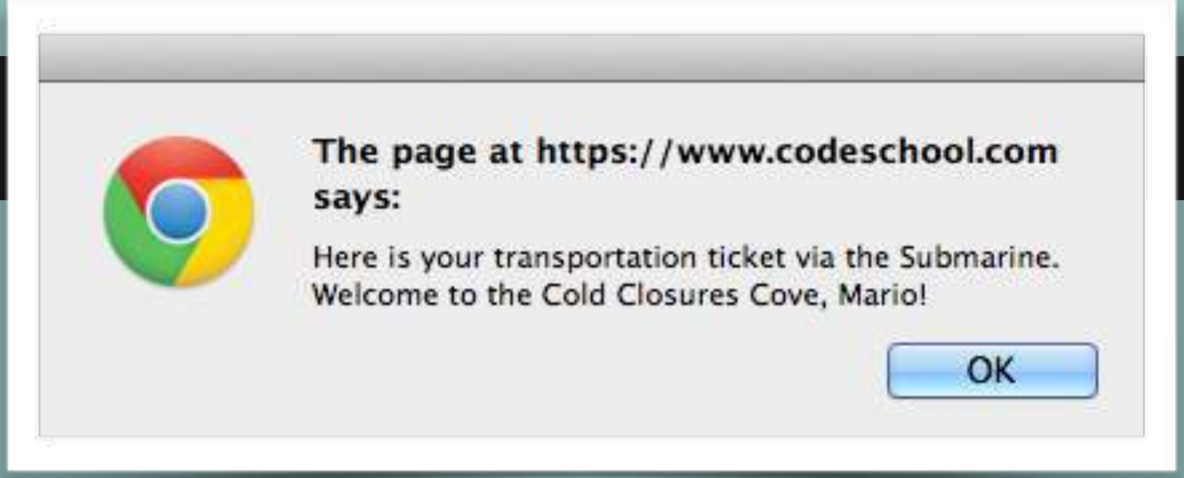


LET'S MAKE SOME TICKETS!

Passing a name to any of our ticket makers will complete our ticket-making process.

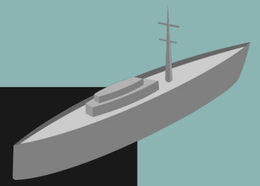


```
getSubmarineTicket("Mario");
```

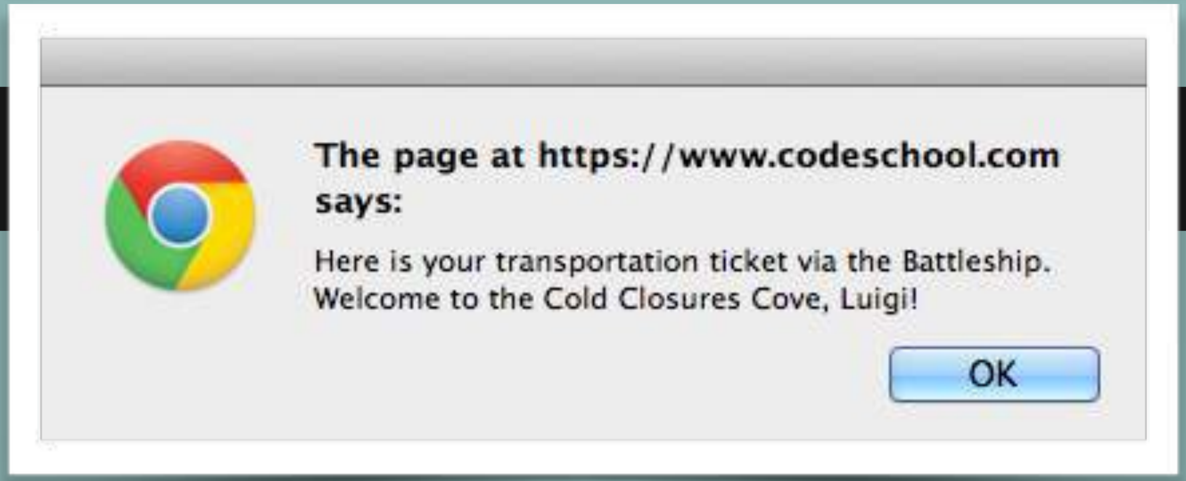


The page at https://www.codeschool.com says:

Here is your transportation ticket via the Submarine.
Welcome to the Cold Closures Cove, Mario!

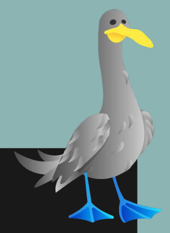


```
getBattleshipTicket("Luigi");
```

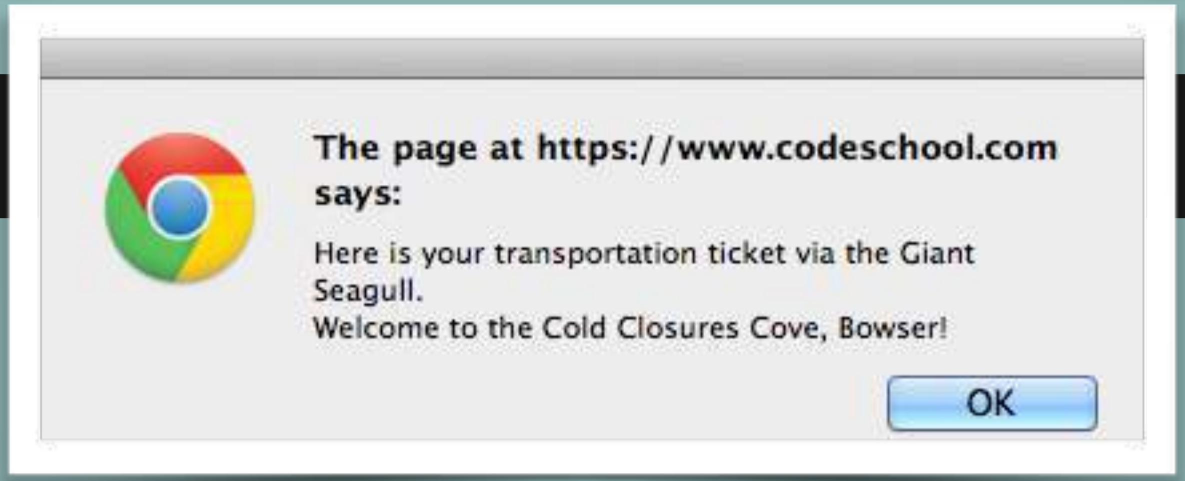


The page at https://www.codeschool.com says:

Here is your transportation ticket via the Battleship.
Welcome to the Cold Closures Cove, Luigi!



```
getGiantSeagullTicket("Bowser");
```



The page at https://www.codeschool.com says:

Here is your transportation ticket via the Giant Seagull.
Welcome to the Cold Closures Cove, Bowser!

ADDING A PASSENGER TRACKER

Closure functions can even modify bound variables in the background

```
function buildCoveTicketMaker( transport ) {
         return function ( name ) {
            alert("Here is your transportation ticket via the " + transport + ".\n" +
                  "Welcome to the Cold Closures Cove, " + name + "!");
         }
}
```

ADDING A PASSENGER TRACKER

Closure functions can even modify bound variables in the background

```
function buildCoveTicketMaker( transport ) {
         let passengerNumber = 0;
         return function ( name ) {

            alert("Here is your transportation ticket via the " + transport + ".\n" +
                  "Welcome to the Cold Closures Cove, " + name + "!\n" +
                                                              );
         }
}
```

We'll start every ticket maker's tracker at 0 passengers.

ADDING A PASSENGER TRACKER

Closure functions can even modify bound variables in the background

```
function buildCoveTicketMaker( transport ) {
         let passengerNumber = 0;
         return function ( name ) {
            passengerNumber++;
            alert("Here is your transportation ticket via the " + transport + ".\n" +
                  "Welcome to the Cold Closures Cove, " + name + "!\n" +
                                                                ");
         }
}
```

When a particular ticket maker is called, we know a new passenger should be added, so we'll increase the tracker.

ADDING A PASSENGER TRACKER

Closure functions can even modify bound variables in the background

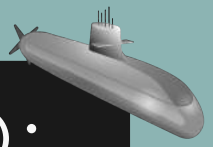
```
function buildCoveTicketMaker( transport ) {
         let passengerNumber = 0;
         return function ( name ) {
            passengerNumber++;
            alert("Here is your transportation ticket via the " + transport + ".\n" +
                  "Welcome to the Cold Closures Cove, " + name + "!\n" +
                  "You are passenger #" + passengerNumber + ".");
         }
}
```

Each time a ticket is "printed," this passengerNumber will contain the precise amount of times this kind of ticket has been given.

ADDING A PASSENGER TRACKER

Closure functions can even modify bound variables in the background

```
function buildCoveTicketMaker( transport ) {
         let passengerNumber = 0;
         return function ( name ) {
            passengerNumber++;
            alert("Here is your transportation ticket via the " + transport + ".\n" +
                  "Welcome to the Cold Closures Cove, " + name + "!\n" +
                  "You are passenger #" + passengerNumber + ".");
         }
 }
```



```
let getSubmarineTicket = buildCoveTicketMaker("Submarine");
getSubmarineTicket;
```

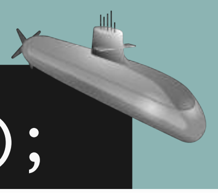
```
function (name) {
    passengerNumber++;
    alert("Here is your transportation ticket via the " + transport + ".\n" +
          "Welcome to the Cold Closures Cove, " + name + "!\n" +
          "You are passenger #" + passengerNumber + ".");
}
```

Notice that no initial value for passengerNumber is evident in our new function. Its value starts at 0 and is adjusted with each call to getSubmarineTicket.

ADDING A PASSENGER TRACKER

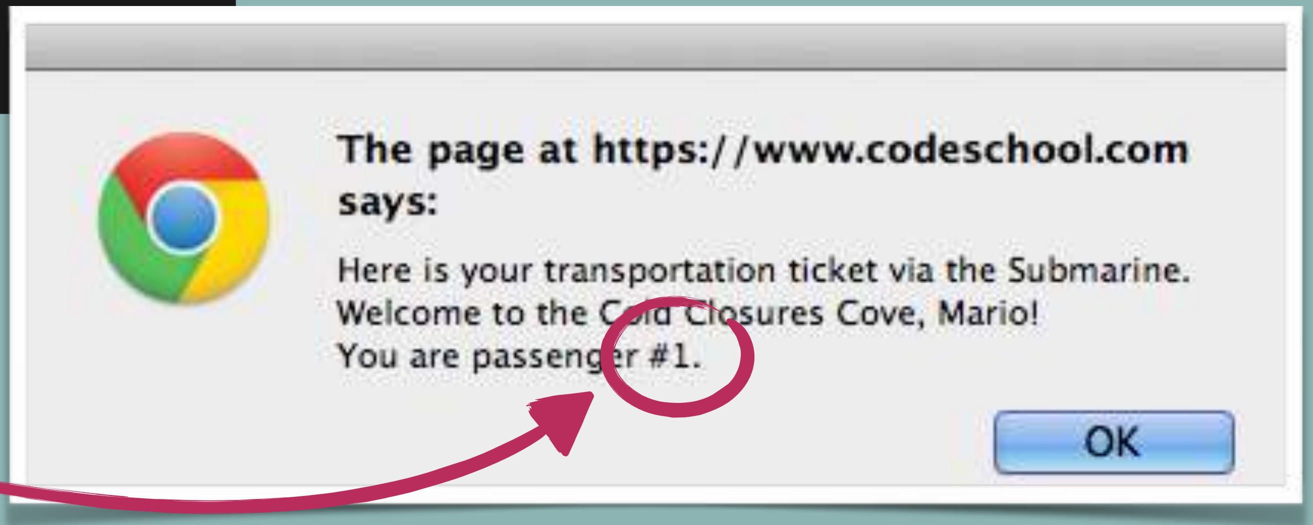
Closure functions can even modify bound variables in the background

```
function buildCoveTicketMaker( transport ) {
         let passengerNumber = 0;
         return function ( name ) {
            passengerNumber++;
            alert("Here is your transportation ticket via the " + transport + ".\n" +
                  "Welcome to the Cold Closures Cove, " + name + "!\n" +
                  "You are passenger #" + passengerNumber + "." );
         }
}
```



```
let getSubmarineTicket = buildCoveTicketMaker("Submarine");
getSubmarineTicket("Mario");
```

On our first call to the new getSubmarineTicket, passengerNumber is incremented to 1.



The page at https://www.codeschool.com says:

Here is your transportation ticket via the Submarine.
Welcome to the Cold Closures Cove, Mario!
You are passenger #1.

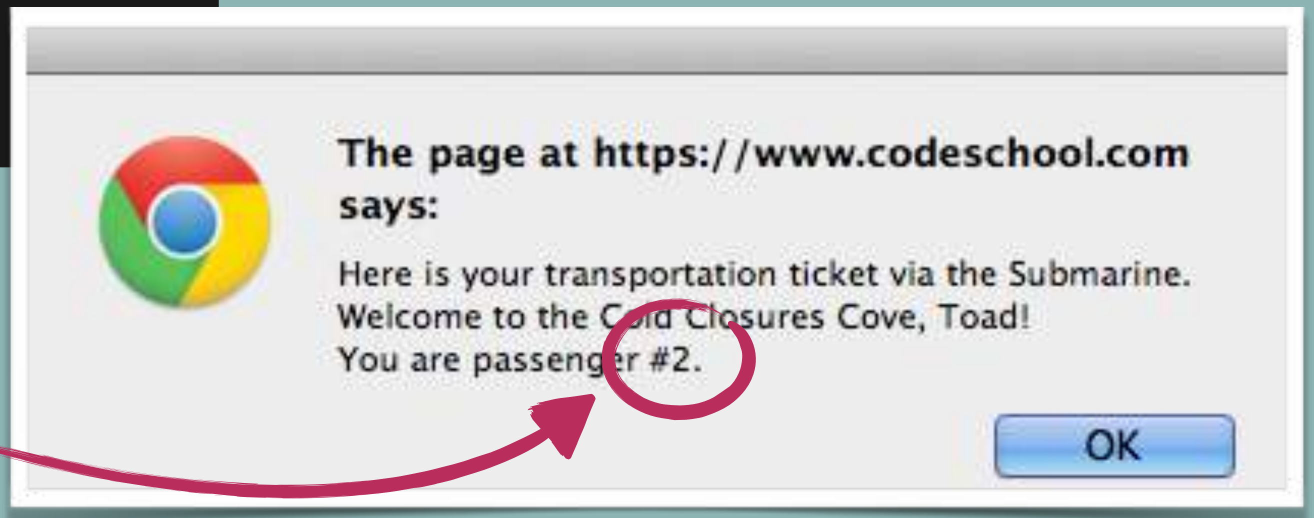
ADDING A PASSENGER TRACKER

Closure functions can even modify bound variables in the background

```
function buildCoveTicketMaker( transport ) {
         let passengerNumber = 0;
         return function ( name ) {
            passengerNumber++;
            alert("Here is your transportation ticket via the " + transport + ".\n" +
                  "Welcome to the Cold Closures Cove, " + name + "!\n" +
                  "You are passenger #" + passengerNumber + ".");
         }
}
```

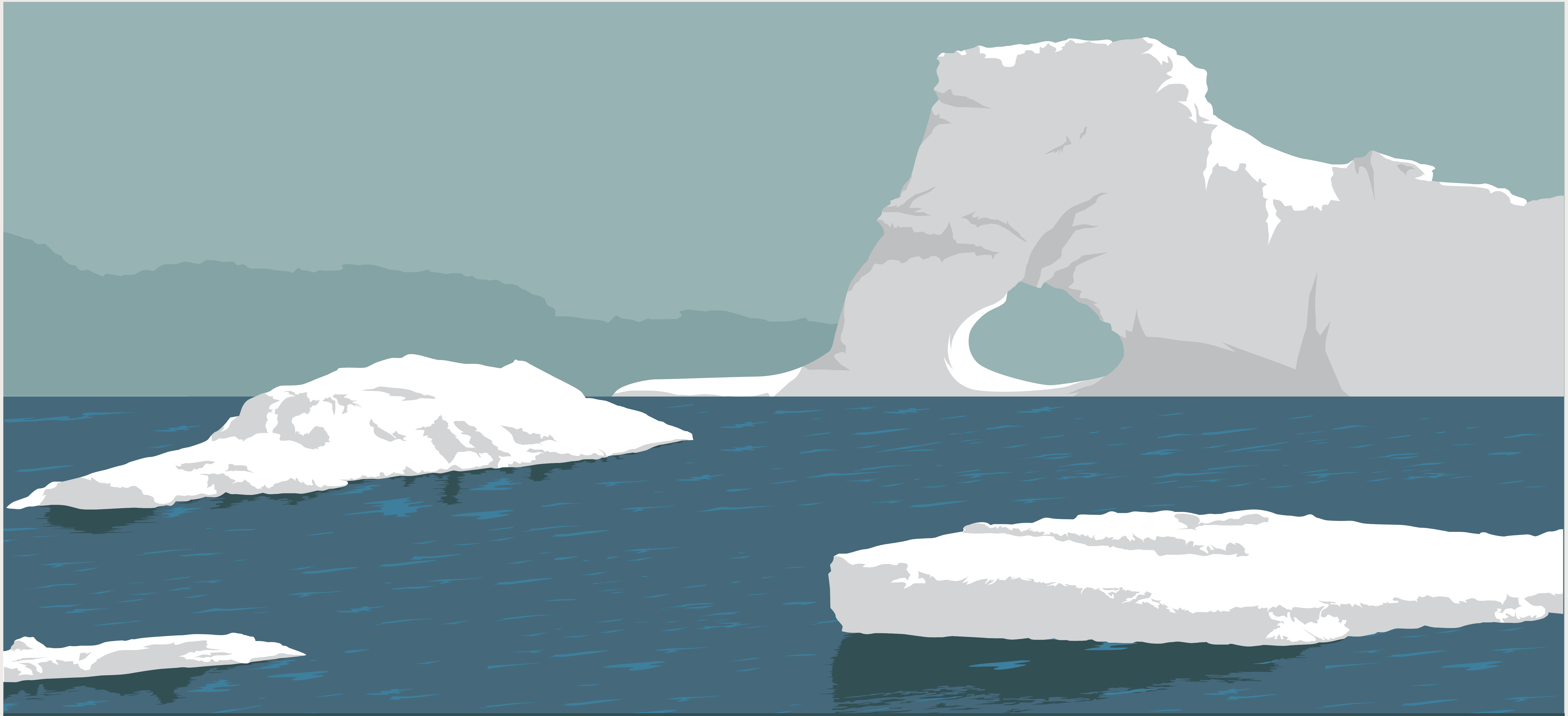
```
let getSubmarineTicket = buildCoveTicketMaker("Submarine");
getSubmarineTicket("Mario");
getSubmarineTicket("Toad");
```

Another call to getSubmarineTicket has passengerNumber incremented to 2! Wow, even though the function's local scope disappeared after Mario's ticket, it KEPT the progress of passengerNumber!



The page at https://www.codeschool.com says:

Here is your transportation ticket via the Submarine.
Welcome to the Cold Closures Cove, Toad!
You are passenger #2.



Explore

COLD CLOSURES COVE