



See the top of the world at

• **BUILT-IN'S BLUFF** •



LEVEL 3

BUILT-INS BLUFF

EXISTING JAVASCRIPT FUNCTIONS

Built-in functions you can use at any time to get and send information

alert() Sends a message to the user in a small pop-up window

```
alert("Alert! Alert! Last call to board Train #4!");
```



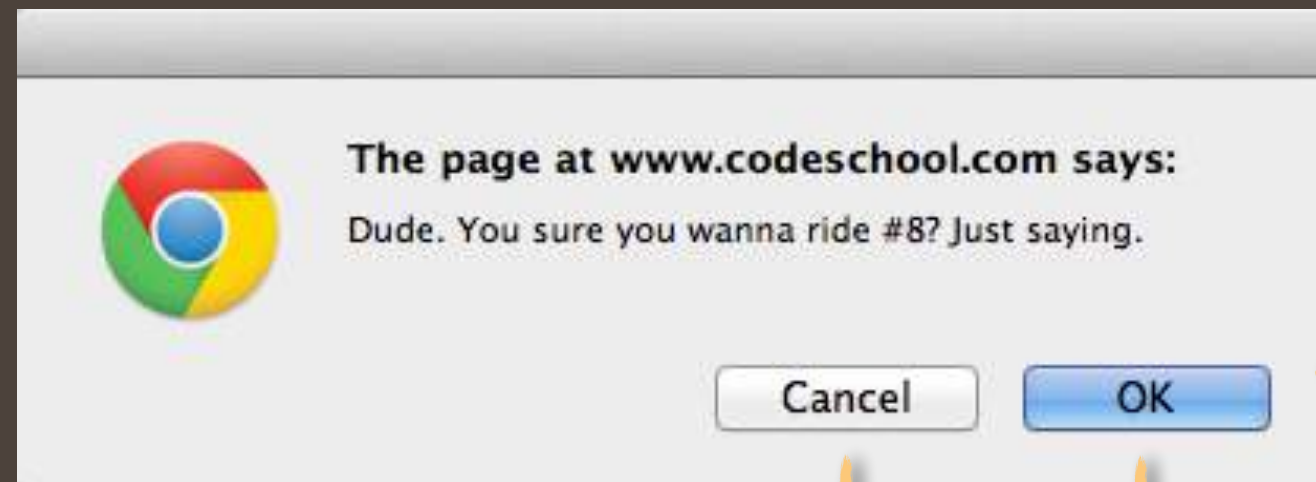
We send the message as a "parameter" to the function by enclosing it in parentheses. It can be any value or String.

EXISTING JAVASCRIPT FUNCTIONS

Built-in functions you can use at any time to get and send information

confirm() Asks user for consent to move forward with an action

```
confirm("Dude. Are you sure you wanna ride #8? Just saying.");
```



false true

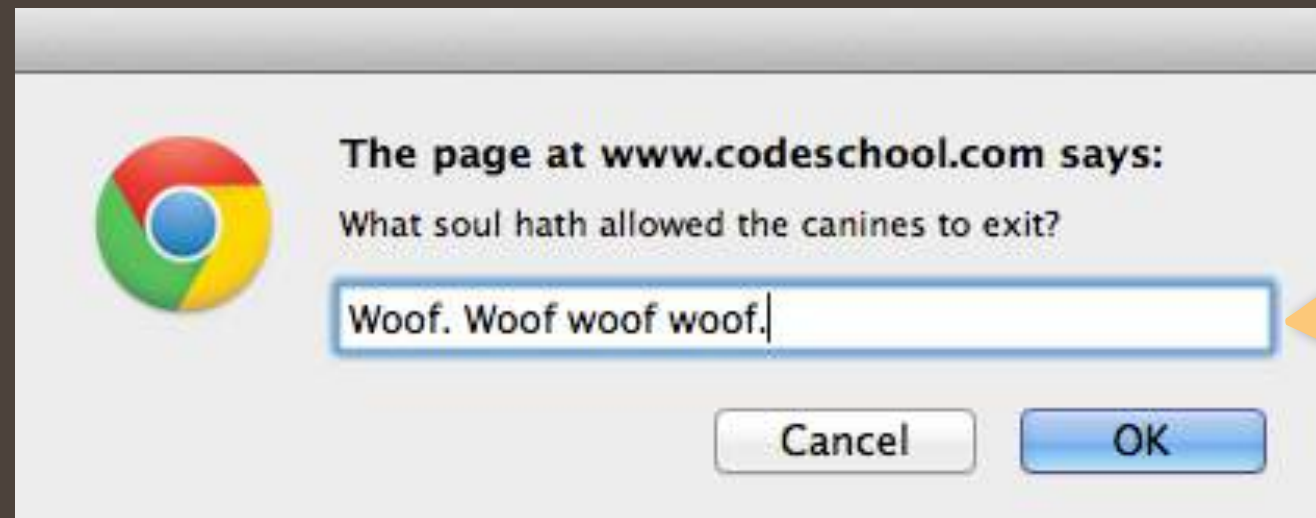
If the user hits OK, the `confirm()` function returns `true`. If the user hits cancel, the `confirm()` function returns `false`.

EXISTING JAVASCRIPT FUNCTIONS

Built-in functions you can use at any time to get and send information

prompt() Sends a message and retrieves an entry from the user

```
prompt("What soul hath allowed the canines to exit?");
```



"When the party was nice,
the party was jumping."
-William Shakespeare

USING DIALOGS WITH VARIABLES

Requesting information and storing with prompt()

```
let userName = prompt("Yo passenger! What's your name?");
```

The prompt() function returns Gregg's entry, which is then stored in the variable userName.



```
> userName
```

→ "Gregg Pollack"

CONFIRMING OUR PASSENGER'S NAME

Using `confirm()` to ensure the user is satisfied with their entry

```
let userName = prompt("Yo passenger! What's your name?");
```



```
confirm("Are you sure your name is " + userName + "?");
```



But wait a minute. This doesn't really do anything. What if Gregg selects cancel, in either dialog box?



USING THE TYPEOF OPERATOR

Identifying the “type” of value inside a variable or expression

```
> typeof true
```

→ "boolean"

```
> typeof "That's not a valid entry!"
```

→ "string"

```
> typeof 42
```

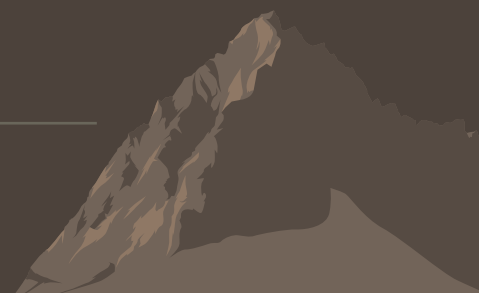
→ "number"

```
> typeof undefined
```

→ "undefined"

```
> typeof null
```

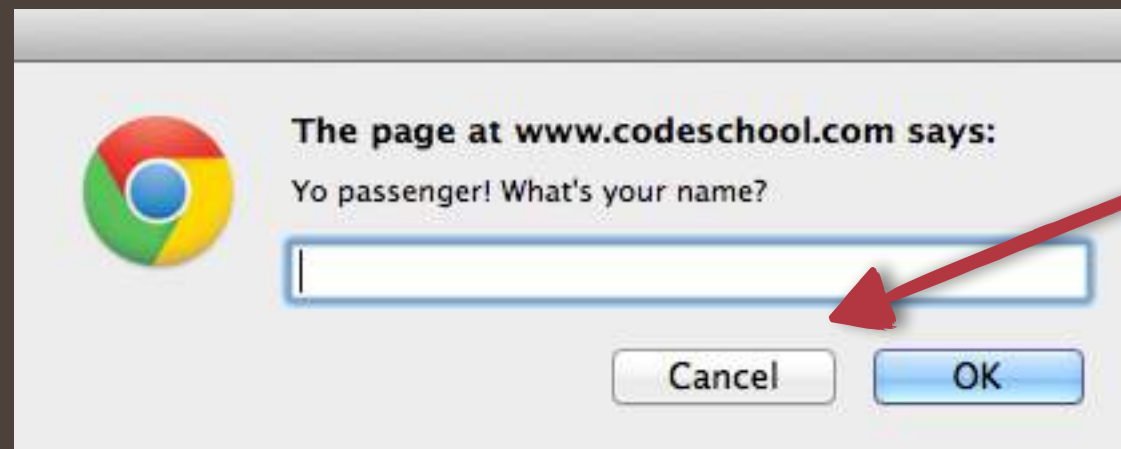
→ "object"



IDENTIFYING A USER'S RESPONSE

The `typeof()` method is useful in checking a variable's contents

```
let userName = prompt("Yo passenger! What's your name?");
```



If the user selects cancel without entering anything, `prompt()` will return a special value called "null," which is not a String.

```
> typeof userName
```

→ "object"

If `userName` is set to "null" by `prompt()`, then it will not be a String, but instead, a generic JavaScript "object."

CANCELING A CONFIRM DIALOG BOX

Using a cancel to divert or restart a process

```
confirm("Are you sure your name is " + userName + "?");
```



Since confirm returns a true or false value, we can use it in conditionals!

```
if ( confirm(*user selects OK here*) ) {  
    *do some code, yo!*  
}
```

A CONFIRMATION LOOP

Let's write some pseudo-code that plans our solution using functions

trains.js

```
...
*make a confirmation flag*
*until the user has confirmed a name, do this:*{
  *request the name using prompt()*

  *if the user says OK at confirm()*{
    *acknowledge the accepted entry*
    *adjust flag to exit the loop*
  }

  *otherwise, cycle back to the top*

}
```

A CONFIRMATION LOOP

Now, we'll create code that matches our intent

trains.js

```
...
*make a confirmation flag*
*until the user has confirmed a name, do this:*{
  *request the name using prompt()*

  *if the user says OK at confirm()*{
    *acknowledge the accepted entry*
    *adjust flag to exit the loop*
  }

  *otherwise, cycle back to the top*

}
```

A CONFIRMATION LOOP

Now, we'll create code that matches our intent

trains.js

```
...  
let gotName = false;  
*until the user has confirmed a name, do this:*{  
  *request the name using prompt()*  
  
  *if the user says OK at confirm()*{  
    *acknowledge the accepted entry*  
    *adjust flag to exit the loop*  
  }  
  
  *otherwise, cycle back to the top*  
  
}
```

This 'flag' will control our loop, based on whether we've got the user's correct name yet.

A CONFIRMATION LOOP

Now, we'll create code that matches our intent

trains.js

```
...  
let gotName = false;  
while(gotName == false){  
    *request the name using prompt()*  
  
    *if the user says OK at confirm()*{  
        *acknowledge the accepted entry*  
        *adjust flag to exit the loop*  
    }  
  
    *otherwise, cycle back to the top*  
  
}
```

A CONFIRMATION LOOP

Now, we'll create code that matches our intent

trains.js

```
...  
let gotName = false;  
while(gotName == false){  
    let userName = prompt("Yo passenger! What's your name?");  
  
    *if the user says OK at confirm()*{  
        *acknowledge the accepted entry*  
        *adjust flag to exit the loop*  
    }  
  
    *otherwise, cycle back to the top*  
  
}
```

A CONFIRMATION LOOP

Now, we'll create code that matches our intent

trains.js

```
...
let gotName = false;
while(gotName == false){
    let userName = prompt("Yo passenger! What's your name?");

    if ( confirm("Are you sure your name is " + userName + "?") ){
        *acknowledge the accepted entry*
        *adjust flag to exit the loop*
    }

    *otherwise, cycle back to the top*
}
```


A CONFIRMATION LOOP

Now, we'll create code that matches our intent

trains.js

```
...  
let gotName = false;  
while(gotName == false){  
    let userName = prompt("Yo passenger! What's your name?");  
  
    if ( confirm("Are you sure your name is " + userName + "?") ){  
        alert("'Sup " + userName + "!");  
        *adjust flag to exit the loop*  
    }  
  
    *otherwise, cycle back to the top*  
  
}
```

A CONFIRMATION LOOP

Now, we'll create code that matches our intent

trains.js


```
...  
let gotName = false;  
while(gotName == false){  
    let userName = prompt("Yo passenger! What's your name?");  
  
    if ( confirm("Are you sure your name is " + userName + "?") ){  
        alert("'Sup " + userName + "!");  
        gotName = true;  
    }  
  
    *otherwise, cycle back to the top*  
  
}
```

A CONFIRMATION LOOP

Now, we'll create code that matches our intent

trains.js

```
...  
let gotName = false;  
while(gotName == false){  
    let userName = prompt("Yo passenger! What's your name?");  
  
    if ( confirm("Are you sure your name is " + userName + "?") ){  
        alert("'Sup " + userName + "!");  
        gotName = true;  
    }  
}
```



If this statement never executes, the loop will restart!

```
}
```

A CONFIRMATION LOOP

Now, we'll create code that matches our intent

trains.js



```
...
let gotName = false;
while(gotName == false){
    let userName = prompt("Yo passenger! What's your name?");

    if ( confirm("Are you sure your name is " + userName + "?") ){
        alert("'Sup " + userName + "!");
        gotName = true;
    }
}
```

Now to see it in action!

**CHECK OUT THE LAST VIDEO OF THIS
LEVEL TO REVIEW THE SCREENCAST OF
THIS EXECUTION!**

