

Understanding Callbacks and How to Respond to Events

```
$0.addEventListener("click", function() {  
  console.log("I got clicked");  
});
```

Higher Order Functions:

These are the functions that create to take functions as input.

Examples:

Higher Order Function

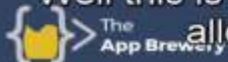
```
document.addEventListener("keypress", respondToKey(event));  
  
function respondToKey(event) {  
  console.log("Key pressed.");  
}
```

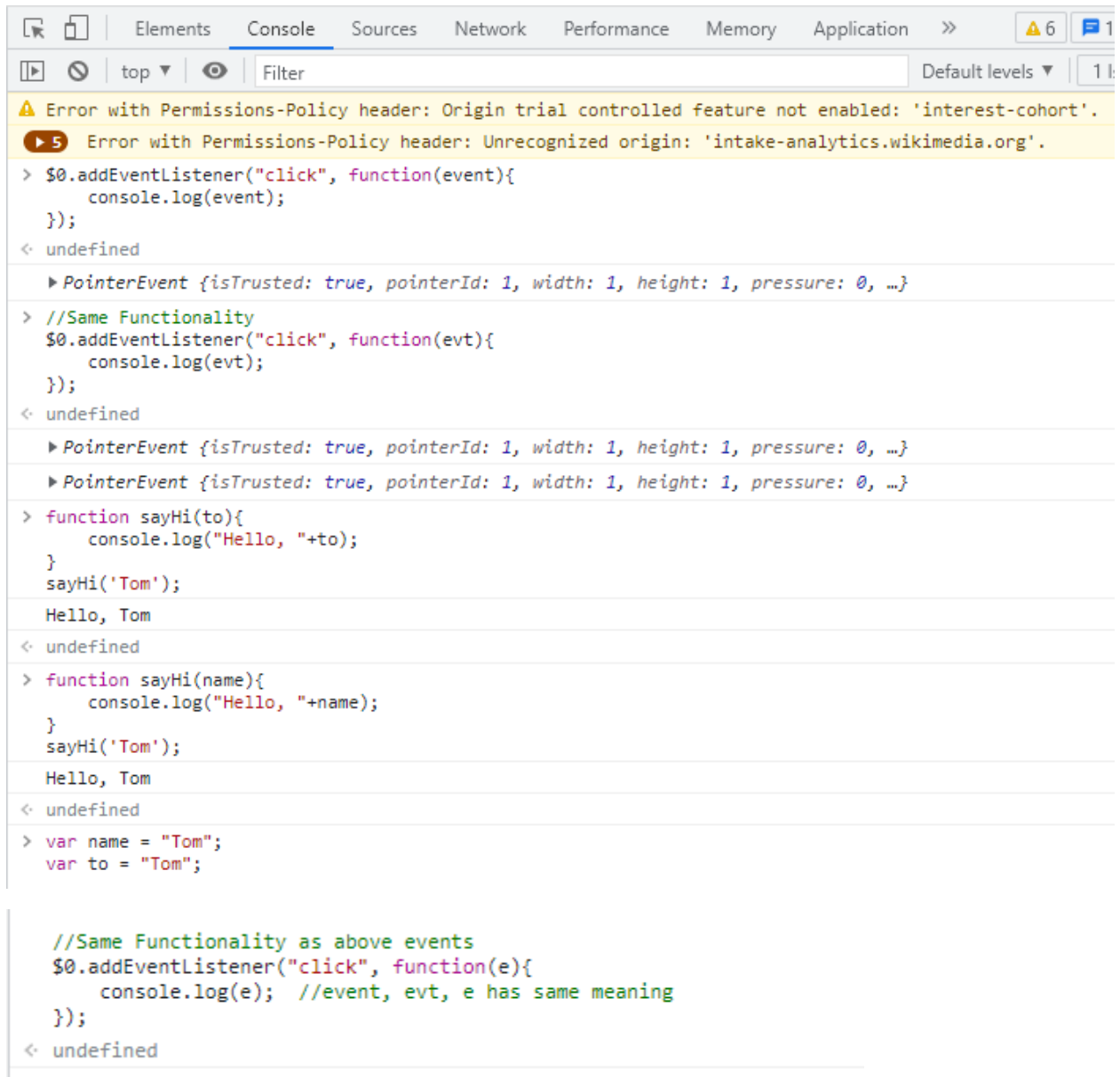
RespondToKey() is a function taken as input function.

Callback Function

```
document.addEventListener("keypress", respondToKey(event));  
  
function respondToKey(event) {  
  console.log("Key pressed.");  
}
```

Well this is actually called a callback function because it allows us to wait for something to finish





The screenshot shows a web browser's developer console with the following content:

- Two yellow error messages at the top:
 - Error with Permissions-Policy header: Origin trial controlled feature not enabled: 'interest-cohort'.
 - Error with Permissions-Policy header: Unrecognized origin: 'intake-analytics.wikimedia.org'.
- A series of JavaScript commands and their outputs:
 - `> $0.addEventListener("click", function(event){ console.log(event); });`
Output: `undefined`
Log: `MouseEvent {isTrusted: true, pointerId: 1, width: 1, height: 1, pressure: 0, ...}`
 - `> //Same Functionality $0.addEventListener("click", function(evt){ console.log(evt); });`
Output: `undefined`
Log: `MouseEvent {isTrusted: true, pointerId: 1, width: 1, height: 1, pressure: 0, ...}`
 - `> function sayHi(to){ console.log("Hello, "+to); } sayHi('Tom');`
Output: `Hello, Tom`
 - `> function sayHi(name){ console.log("Hello, "+name); } sayHi('Tom');`
Output: `Hello, Tom`
 - `> var name = "Tom"; var to = "Tom";`
 - `> //Same Functionality as above events $0.addEventListener("click", function(e){ console.log(e); //event, evt, e has same meaning });`
Output: `undefined`

```
function anotherAddEventListener(typeOfEvent, callback){
```

```
    // Detect Event Code....
```

```
    var eventThatHappened = {
```

```
        eventType: "keypress",
```

```
        key: "p",
```

```
        durationOfKeypress: 2
```

```

    }
    if(eventThatHappened.eventType === typeOfEvent){
        callback(eventThatHappened);
    }
}
anotherAddEventListener("keypress", function(event){
    console.log(event);
});

```

```

document.addEventListener("keypress", function(event){
    console.log(event);
});

```

```

> function anotherAddEventListener(typeOfEvent, callback){
  // Detect Event Code....

  var eventThatHappened = {
    eventType: "keypress",
    key: "p",
    durationOfKeypress: 2
  }
  if(eventThatHappened.eventType === typeOfEvent){
    callback(eventThatHappened);
  }
}
< undefined

> anotherAddEventListener("keypress", function(event){
  console.log(event);
});
  ▶ {eventType: 'keypress', key: 'p', durationOfKeypress: 2}
< undefined

> document.addEventListener("keypress", function(event){
  console.log(event);
});
< undefined

> |

```

With Debugger

```

function anotherAddEventListener(typeOfEvent, callback){

```

```
// Detect Event Code....
```

```
var eventThatHappened = {  
  eventType: "keypress",  
  key: "p",  
  durationOfKeypress: 2  
}  
if(eventThatHappened.eventType === typeOfEvent){  
  callback(eventThatHappened);  
}  
}
```

```
//Use Debugger
```

```
debugger;  
anotherAddEventListener("keypress", function(event){  
  console.log(event);  
});
```

Elements Console Sources Network Performance Memory Application Security Lighthouse

top Filter

⚠ Error with Permissions-Policy header: Origin trial controlled feature not enabled: 'interest-cohort'.

5 ⚠ Error with Permissions-Policy header: Unrecognized origin: 'intake-analytics.wikimedia.org'.

```
> function anotherAddEventListener(typeOfEvent, callback){
  // Detect Event Code....

  var eventThatHappened = {
    eventType: "keypress",
    key: "p",
    durationOfKeypress: 2
  }
  if(eventThatHappened.eventType === typeOfEvent){
    callback(eventThatHappened);
  }
}

//Use Debugger
debugger;
anotherAddEventListener("keypress", function(event){
  console.log(event);
});

▶ {eventType: 'keypress', key: 'p', durationOfKeypress: 2}
```

> |

ents Console Sources Network Performance Memory Application Security Lighthouse Recorder ⚠ Performance insights ⚠

load.php?lang=e...=1&skin=vector load.php?lang=e...&version=re1mk content.js VM80 x

```
1 function anotherAddEventListener(typeOfEvent, callback){ typeOfEvent = "keypress", callback = f (event)
2   // Detect Event Code....
3
4   var eventThatHappened = { eventThatHappened = {eventType: 'keypress', key: 'p', durationOfKeypress: 2}
5     eventType: "keypress",
6     key: "p",
7     durationOfKeypress: 2
8   }
9   if(eventThatHappened.eventType === typeOfEvent){ eventThatHappened = {eventType: 'keypress', key: 'p', durationOfKeypress: 2}
10     callback(eventThatHappened); callback = f (event)
11   }
12 }
13
14 //Use Debugger
15 debugger;
16 anotherAddEventListener("keypress", function(event){
17   console.log(event);
18 });
```

