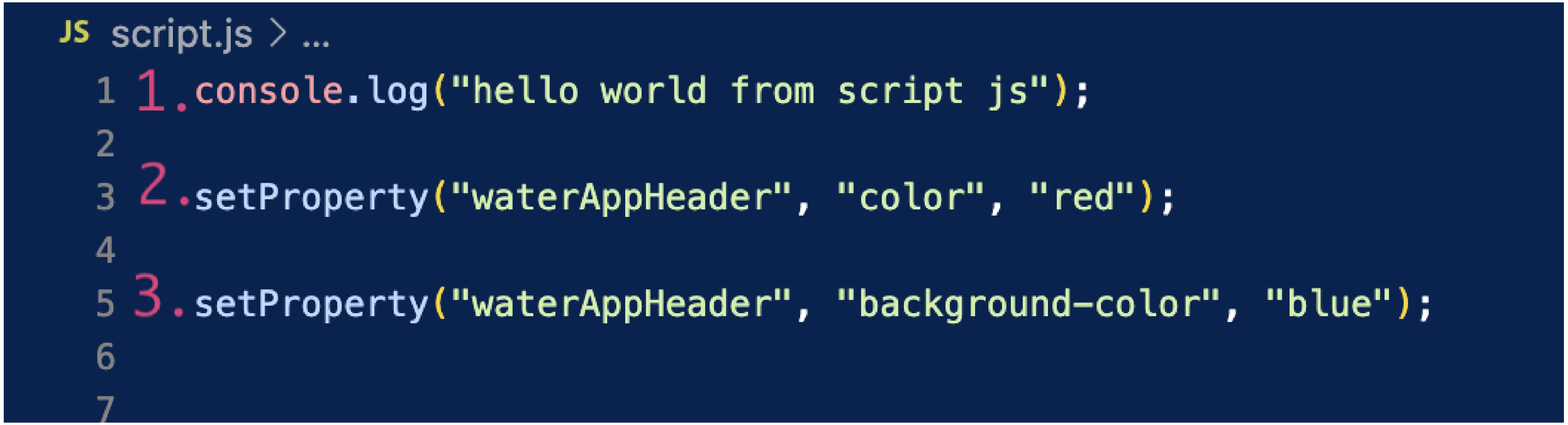
How Events work

Bird’s Eye View

How do programs run?
TOP to BOTTOM

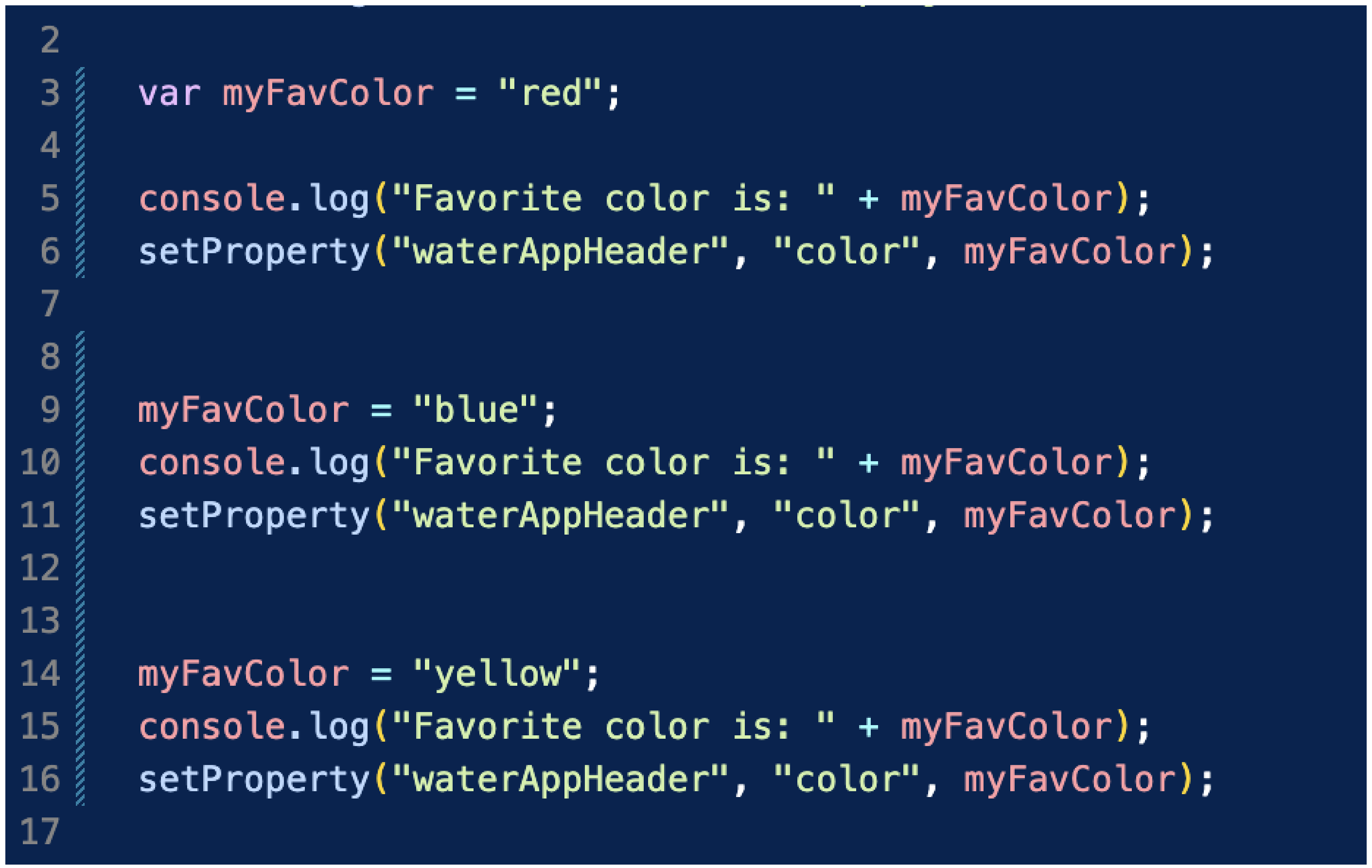
- Each step goes in order.
- Happens faster than you can see it!



script.js > ...

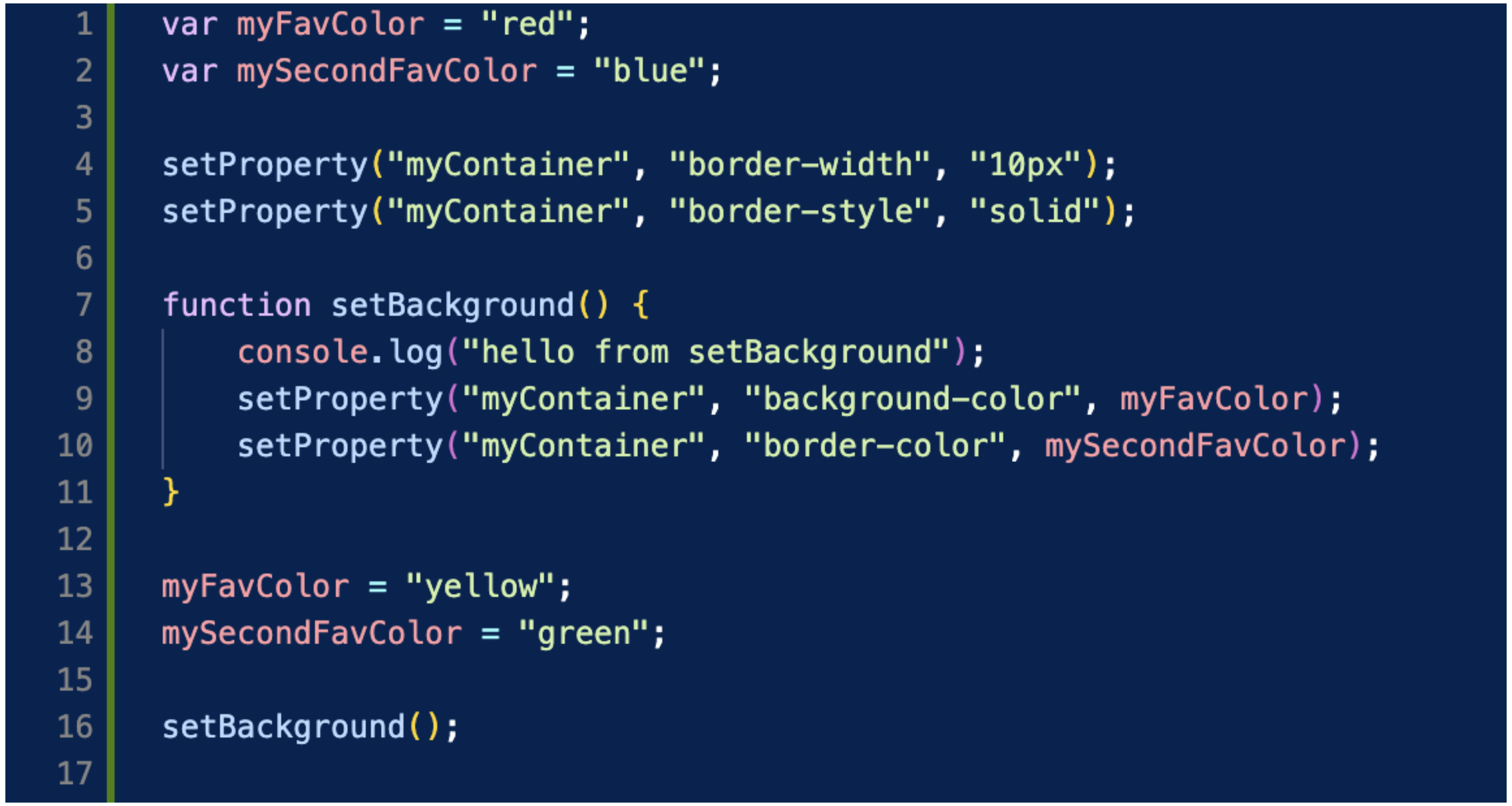
```
1.console.log("hello world from script js");

2.setProperty("waterAppHeader", "color", "red");

3.setProperty("waterAppHeader", "background-color", "blue");


```

Pop Quiz:
* What will you see on the screen?
* What will this print?



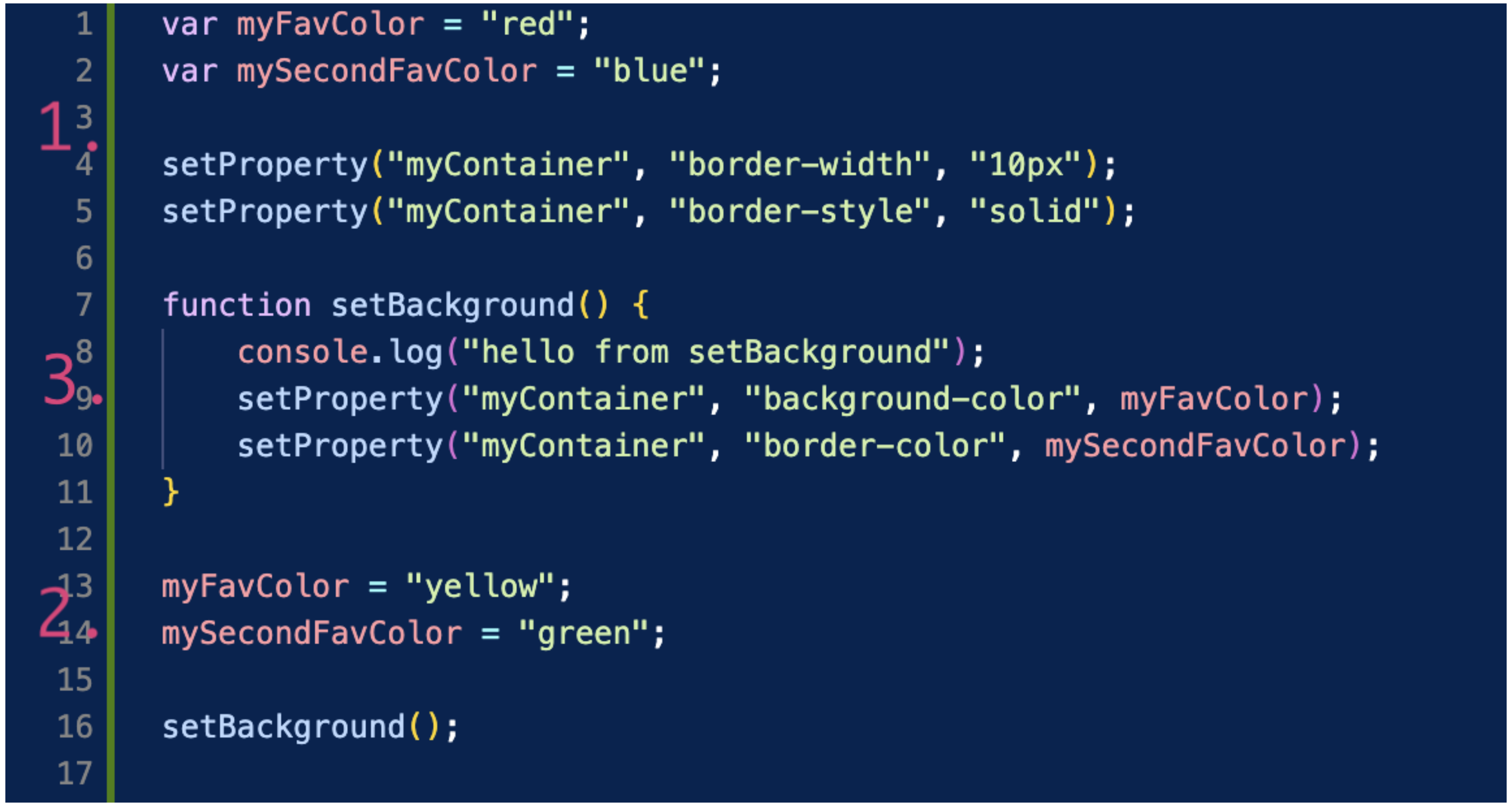
```
var myFavColor = "red";

console.log("Favorite color is: " + myFavColor);
setProperty("waterAppHeader", "color", myFavColor);


myFavColor = "blue";
console.log("Favorite color is: " + myFavColor);
setProperty("waterAppHeader", "color", myFavColor);


myFavColor = "yellow";
console.log("Favorite color is: " + myFavColor);
setProperty("waterAppHeader", "color", myFavColor);
```

Functions change up the order.
What color will the background be?????????



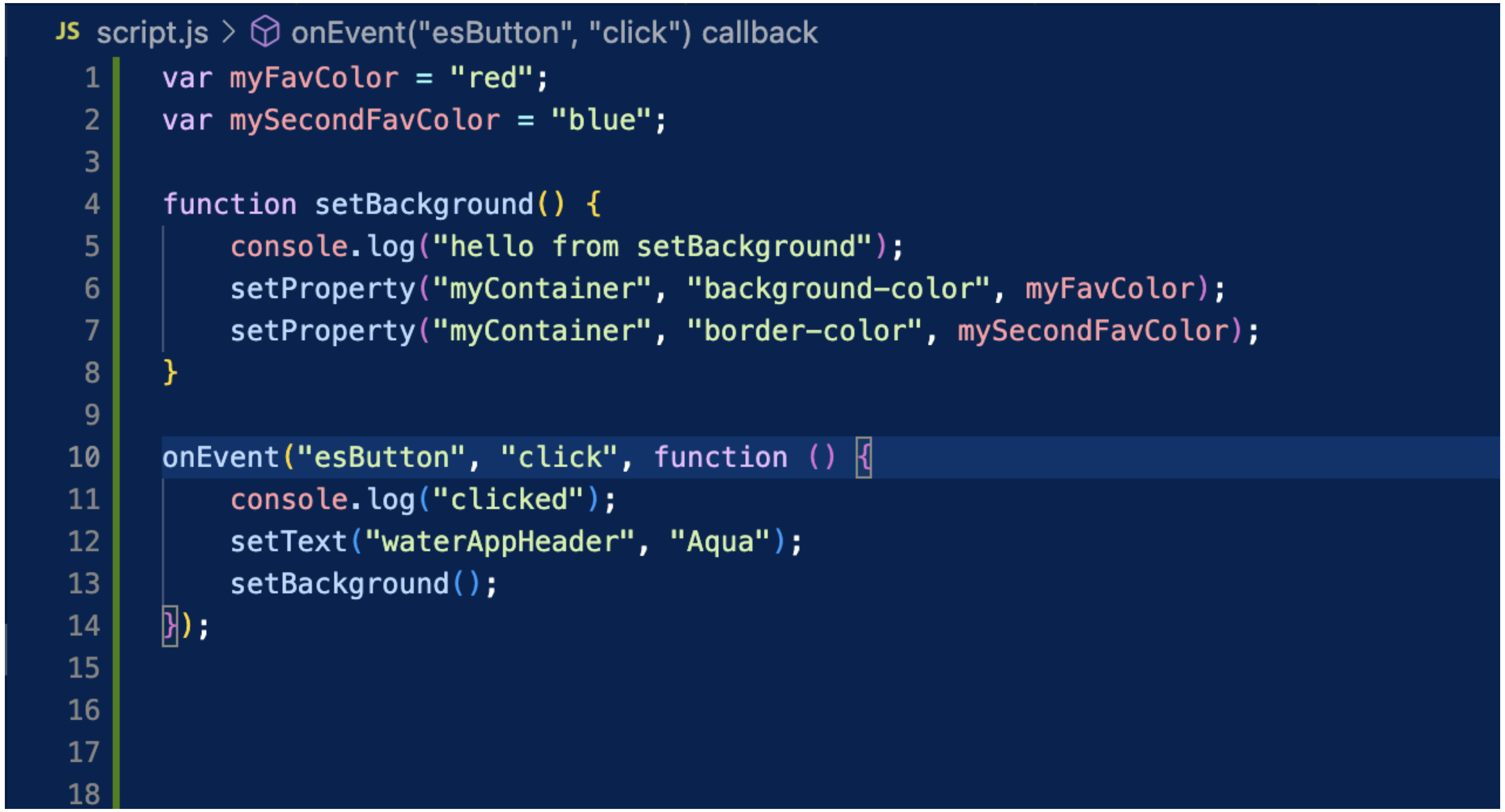
```javascript
var myFavColor = "red";
var mySecondFavColor = "blue";

setProperty("myContainer", "border-width", "10px");
setProperty("myContainer", "border-style", "solid");

function setBackground() {
    console.log("hello from setBackground");
    setProperty("myContainer", "background-color", myFavColor);
    setProperty("myContainer", "border-color", mySecondFavColor);
}

myFavColor = "yellow";
mySecondFavColor = "green";

setBackground();

```

Functions change up the order. What color will the background be?????????



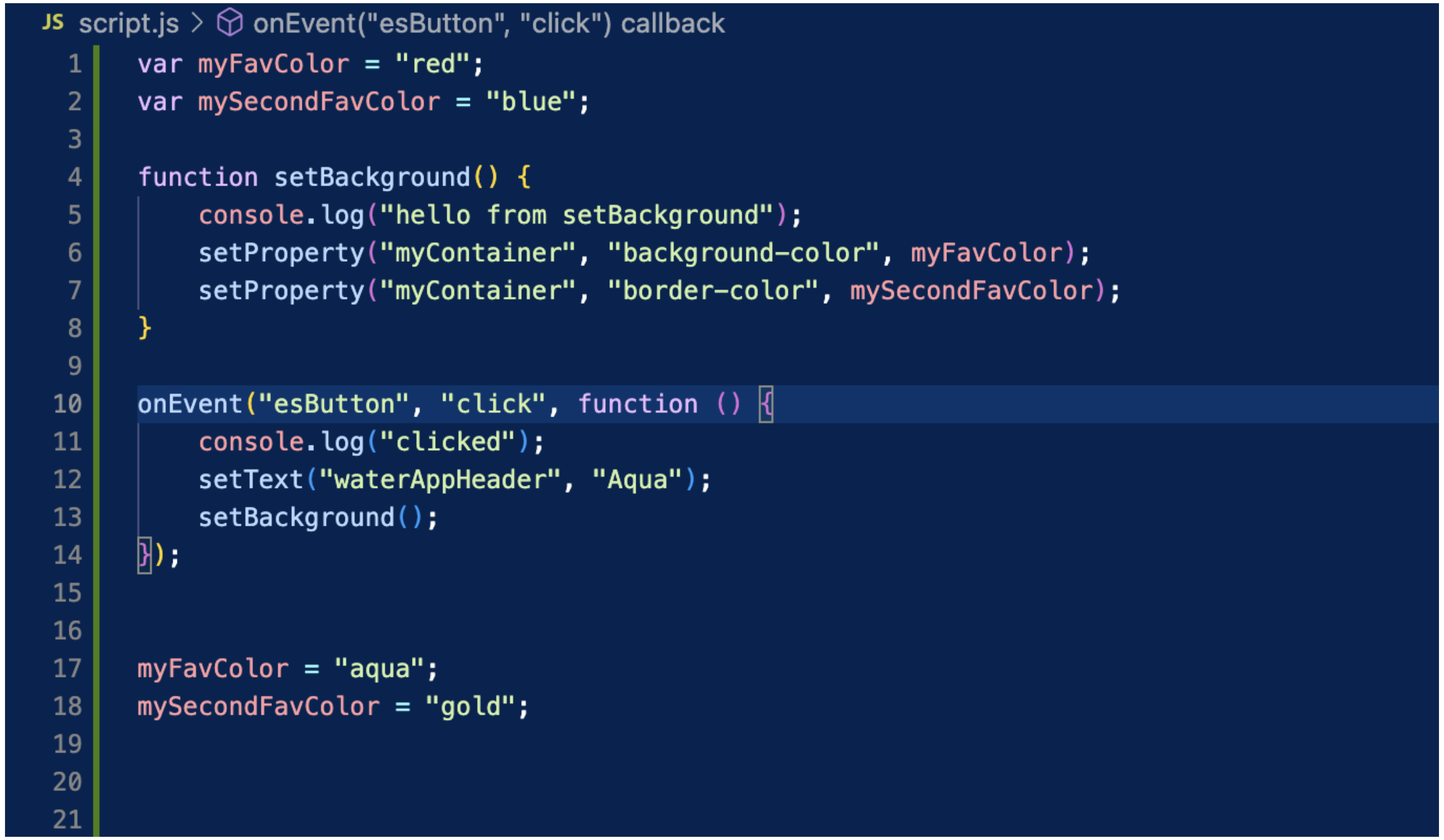
```
  var myFavColor = "red";
  var mySecondFavColor = "blue";

1 setProperty("myContainer", "border-width", "10px");
  setProperty("myContainer", "border-style", "solid");

  function setBackground() {
      console.log("hello from setBackground");
3     setProperty("myContainer", "background-color", myFavColor);
      setProperty("myContainer", "border-color", mySecondFavColor);
  }

  myFavColor = "yellow";
2 mySecondFavColor = "green";

  setBackground();
```

Events can fire off a function whenever you want using a CALLBACK function.



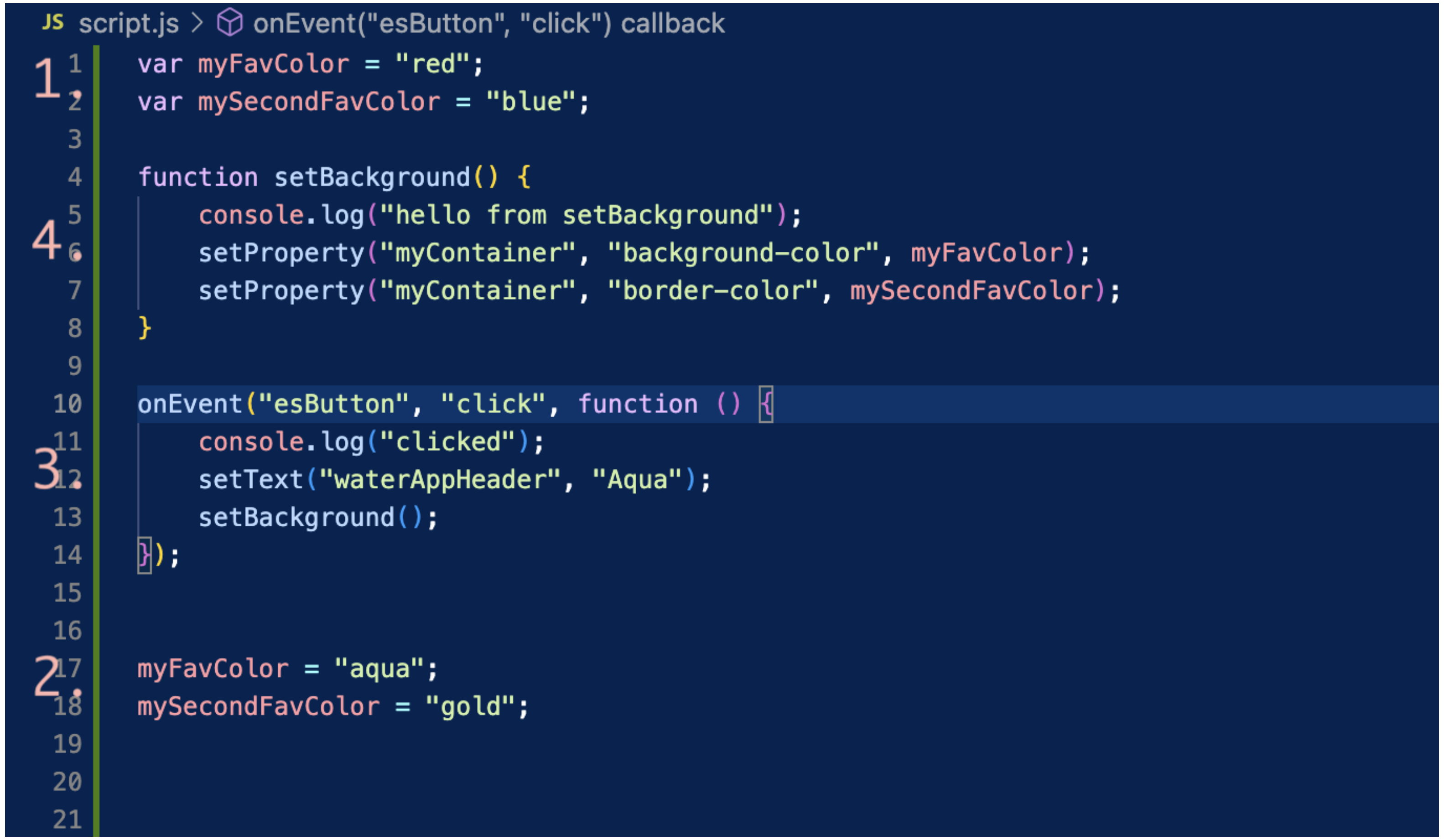
```
var myFavColor = "red";
var mySecondFavColor = "blue";

function setBackground() {
    console.log("hello from setBackground");
    setProperty("myContainer", "background-color", myFavColor);
    setProperty("myContainer", "border-color", mySecondFavColor);
}

onEvent("esButton", "click", function () {
    console.log("clicked");
    setText("waterAppHeader", "Aqua");
    setBackground();
});
```

Pop Quiz: What color will the background be AFTER the esButton is clicked?


JS script.js > onEvent("esButton", "click") callback

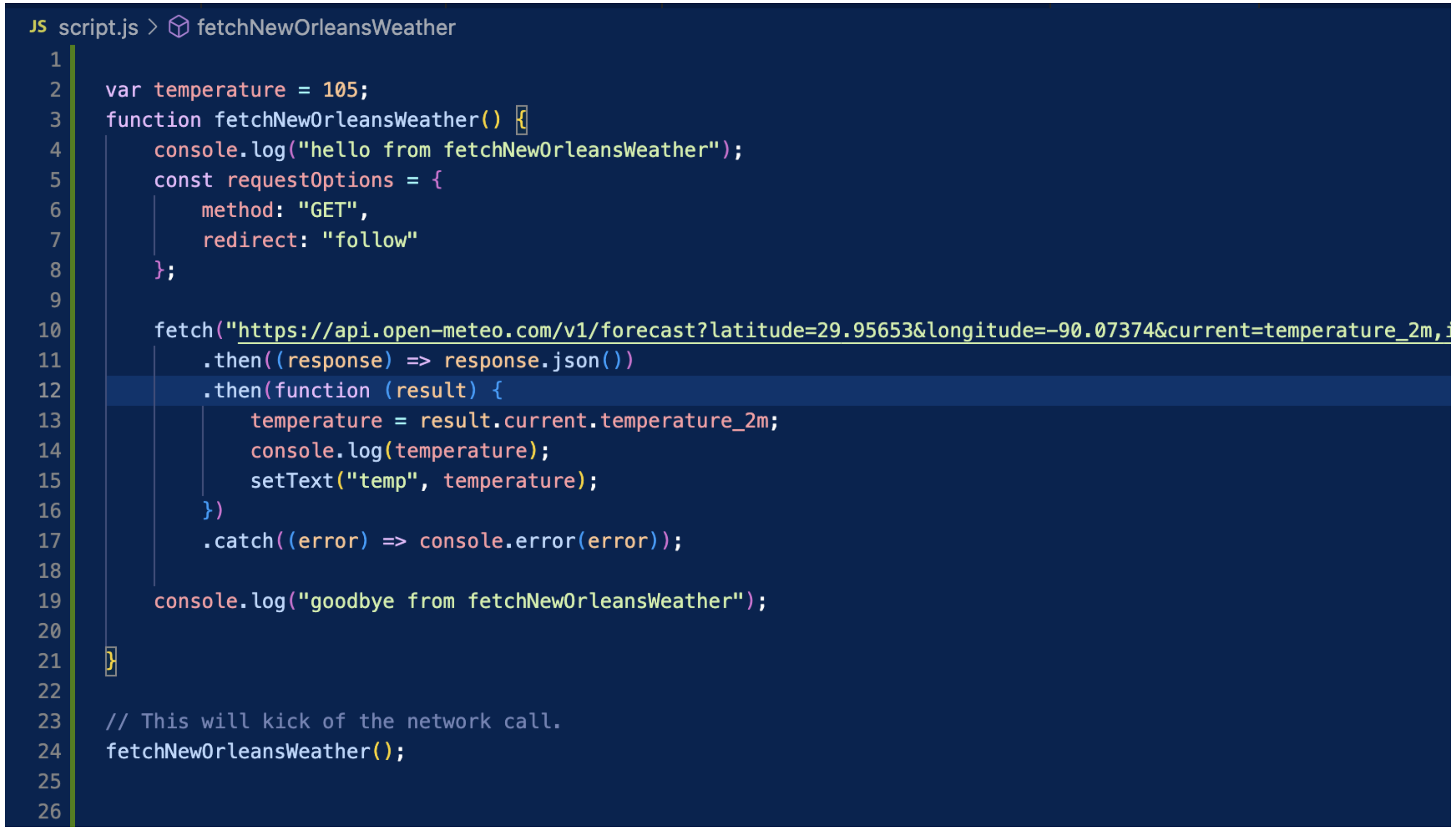
```javascript
var myFavColor = "red";
var mySecondFavColor = "blue";

function setBackground() {
    console.log("hello from setBackground");
    setProperty("myContainer", "background-color", myFavColor);
    setProperty("myContainer", "border-color", mySecondFavColor);
}

onEvent("esButton", "click", function () {
    console.log("clicked");
    setText("waterAppHeader", "Aqua");
    setBackground();
});


myFavColor = "aqua";
mySecondFavColor = "gold";



```

Can you follow the thread?



```javascript
var myFavColor = "red";
var mySecondFavColor = "blue";

function setBackground() {
    console.log("hello from setBackground");
    setProperty("myContainer", "background-color", myFavColor);
    setProperty("myContainer", "border-color", mySecondFavColor);
}

onEvent("esButton", "click", function () {
    console.log("clicked");
    setText("waterAppHeader", "Aqua");
    setBackground();
});


myFavColor = "aqua";
mySecondFavColor = "gold";
```

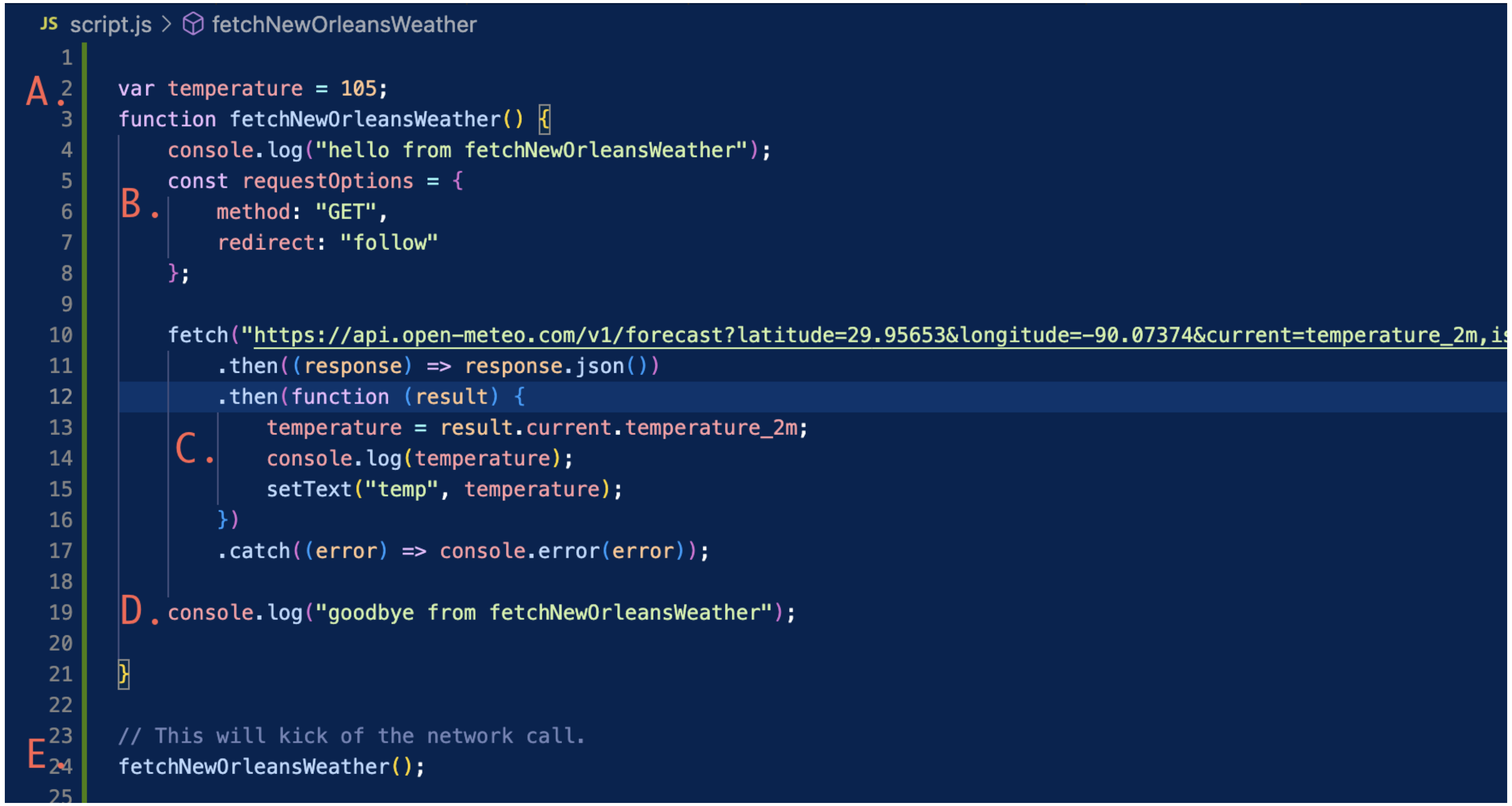
What about API calls? Where does the thread go?



script.js > fetchNewOrleansWeather

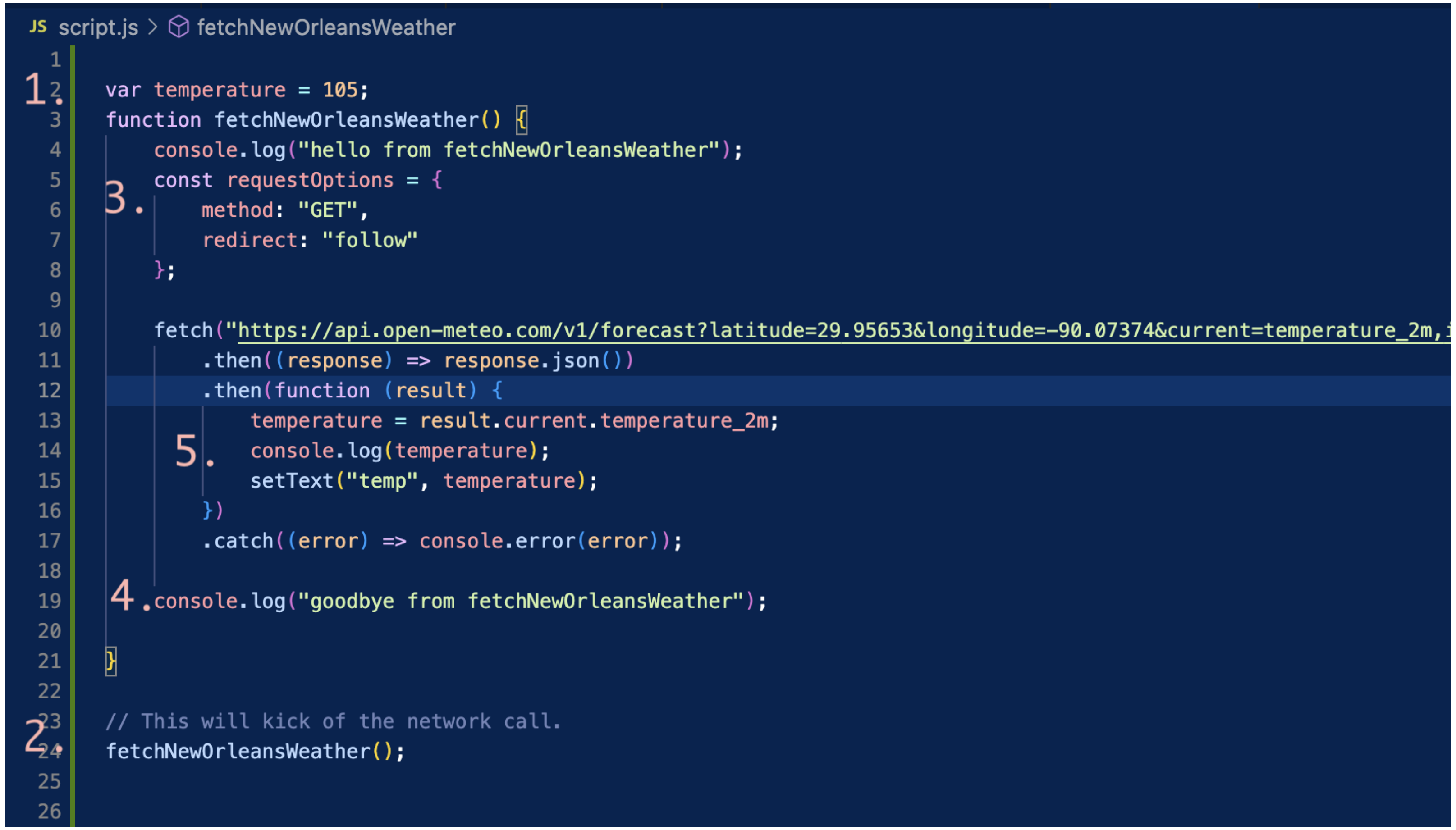
```javascript

var temperature = 105;
function fetchNewOrleansWeather() {
    console.log("hello from fetchNewOrleansWeather");
    const requestOptions = {
        method: "GET",
        redirect: "follow"
    };

    fetch("https://api.open-meteo.com/v1/forecast?latitude=29.95653&longitude=-90.07374&current=temperature_2m,
        .then((response) => response.json())
        .then(function (result) {
            temperature = result.current.temperature_2m;
            console.log(temperature);
            setText("temp", temperature);
        })
        .catch((error) => console.error(error));

    console.log("goodbye from fetchNewOrleansWeather");

}

// This will kick of the network call.
fetchNewOrleansWeather();


```

What about API calls? Where does the thread go? Put the letters in order from first to last, by time.



```
A. var temperature = 105;
function fetchNewOrleansWeather() {
    console.log("hello from fetchNewOrleansWeather");
    const requestOptions = {
B.      method: "GET",
        redirect: "follow"
    };

    fetch("https://api.open-meteo.com/v1/forecast?latitude=29.95653&longitude=-90.07374&current=temperature_2m,i
        .then((response) => response.json())
        .then(function (result) {
            temperature = result.current.temperature_2m;
C.          console.log(temperature);
            setText("temp", temperature);
        })
        .catch((error) => console.error(error));

D.  console.log("goodbye from fetchNewOrleansWeather");

}

// This will kick of the network call.
E. fetchNewOrleansWeather();
```

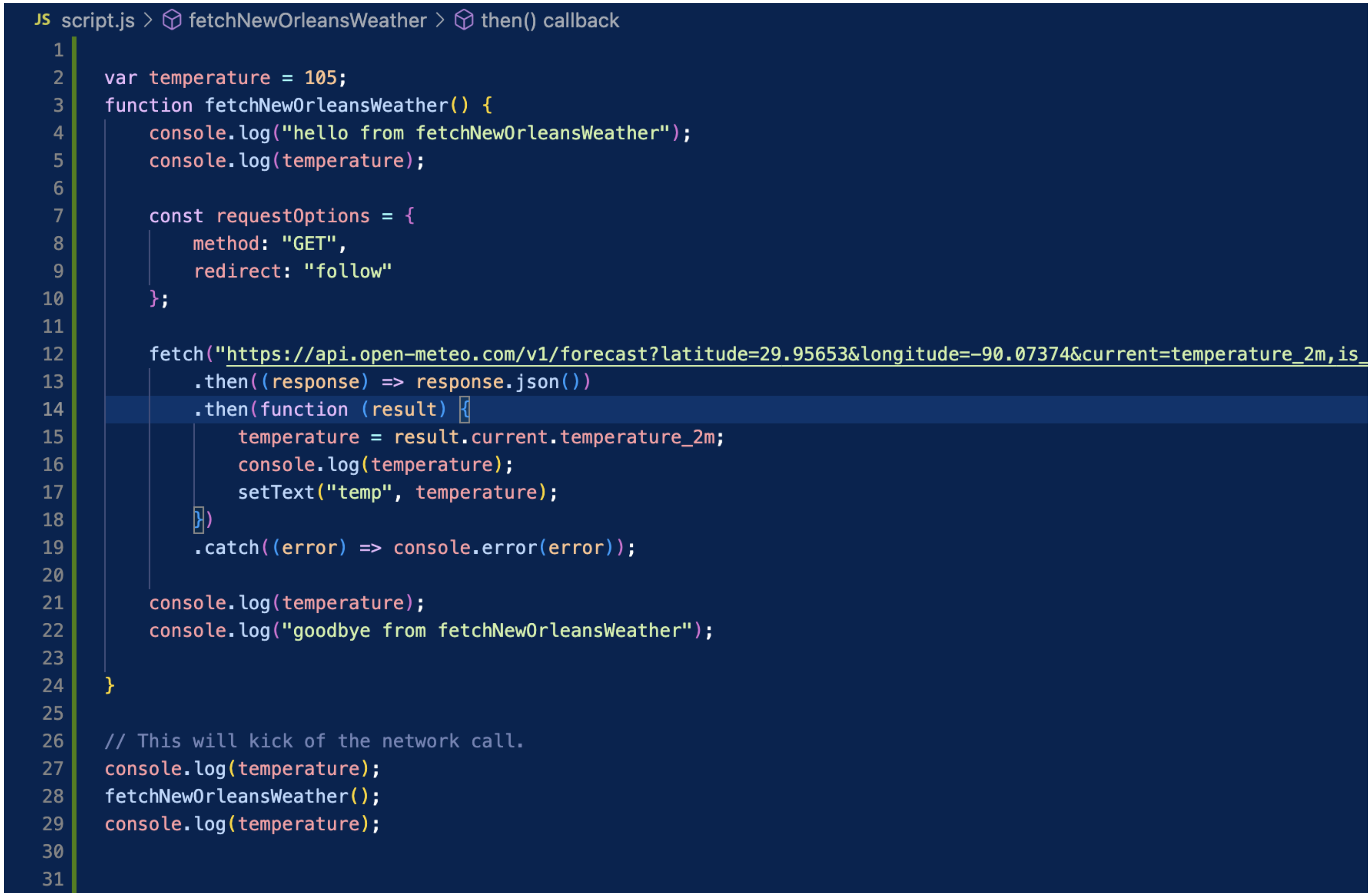
What about API calls? Where does the thread go?



script.js > fetchNewOrleansWeather

```javascript

1. var temperature = 105;
function fetchNewOrleansWeather() {
    console.log("hello from fetchNewOrleansWeather");
    const requestOptions = {
3.      method: "GET",
        redirect: "follow"
    };

    fetch("https://api.open-meteo.com/v1/forecast?latitude=29.95653&longitude=-90.07374&current=temperature_2m,
        .then((response) => response.json())
        .then(function (result) {
            temperature = result.current.temperature_2m;
5.          console.log(temperature);
            setText("temp", temperature);
        })
        .catch((error) => console.error(error));

4.  console.log("goodbye from fetchNewOrleansWeather");

}

// This will kick of the network call.
2. fetchNewOrleansWeather();


```

Pop Quiz: What will this print?



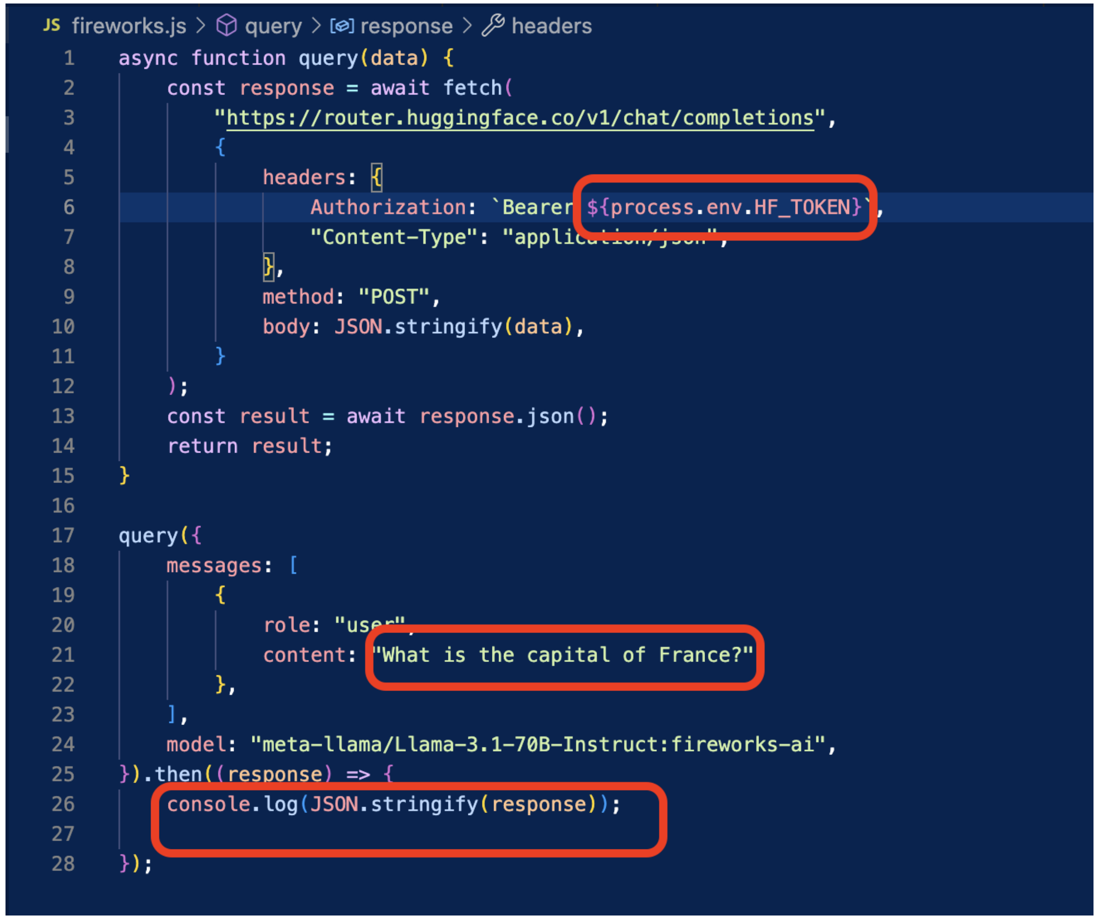
script.js > fetchNewOrleansWeather > then() callback

```javascript

var temperature = 105;
function fetchNewOrleansWeather() {
    console.log("hello from fetchNewOrleansWeather");
    console.log(temperature);

    const requestOptions = {
        method: "GET",
        redirect: "follow"
    };

    fetch("https://api.open-meteo.com/v1/forecast?latitude=29.95653&longitude=-90.07374&current=temperature_2m,is_
        .then((response) => response.json())
        .then(function (result) {
            temperature = result.current.temperature_2m;
            console.log(temperature);
            setText("temp", temperature);
        })
        .catch((error) => console.error(error));

    console.log(temperature);
    console.log("goodbye from fetchNewOrleansWeather");

}

// This will kick of the network call.
console.log(temperature);
fetchNewOrleansWeather();
console.log(temperature);


```

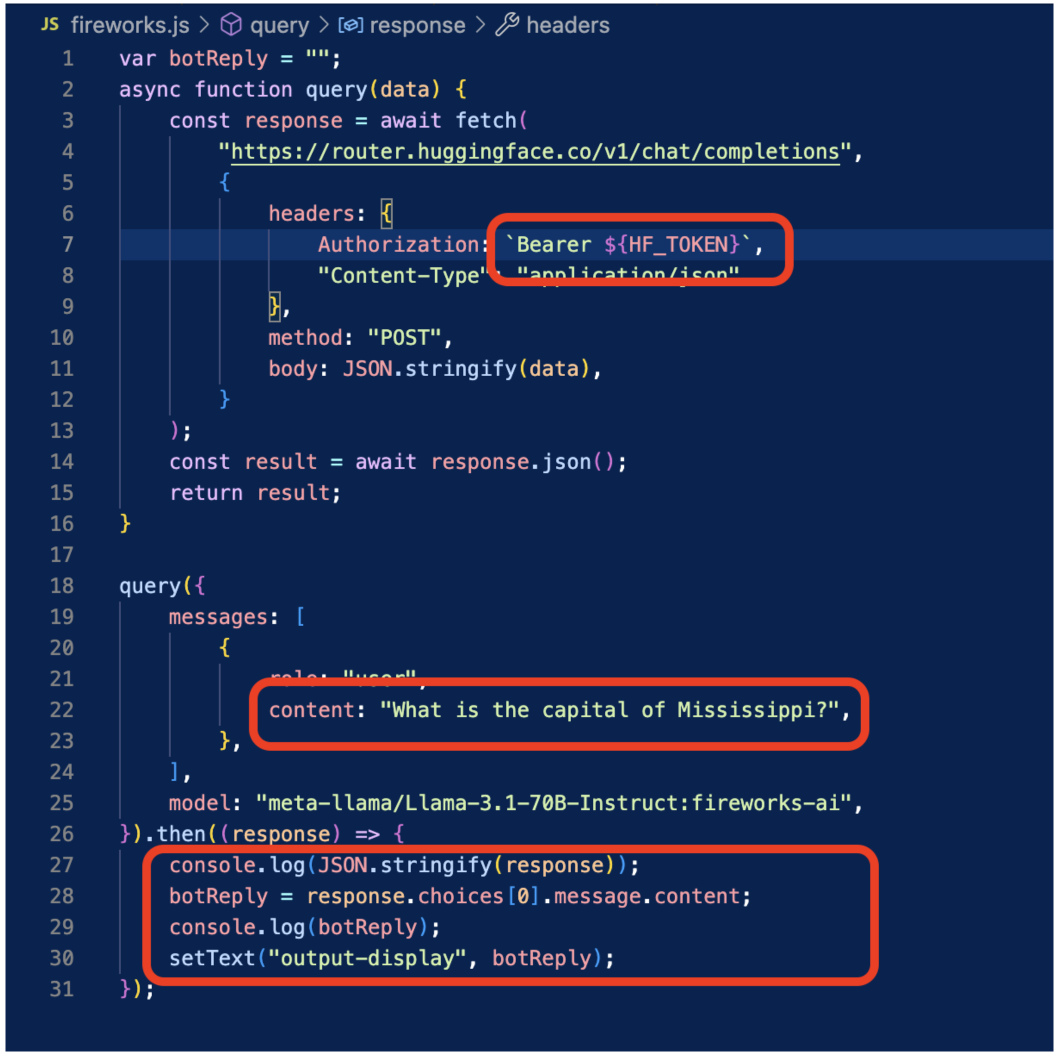
Overview

- Programs run in order
- Programs define functions and assign variables (in order).
- Programs have functions that change the flow.
- Events add a way to run a function when someone clicks (or other events). This is called “listening for an event”
- Fetching an API takes time.
- When the API returns the value (ex: weather), that is another kind of event.
- The `.then(response => { })` part of the program is listening for the data to return.

Cheat Sheet for Hugging Face generated code

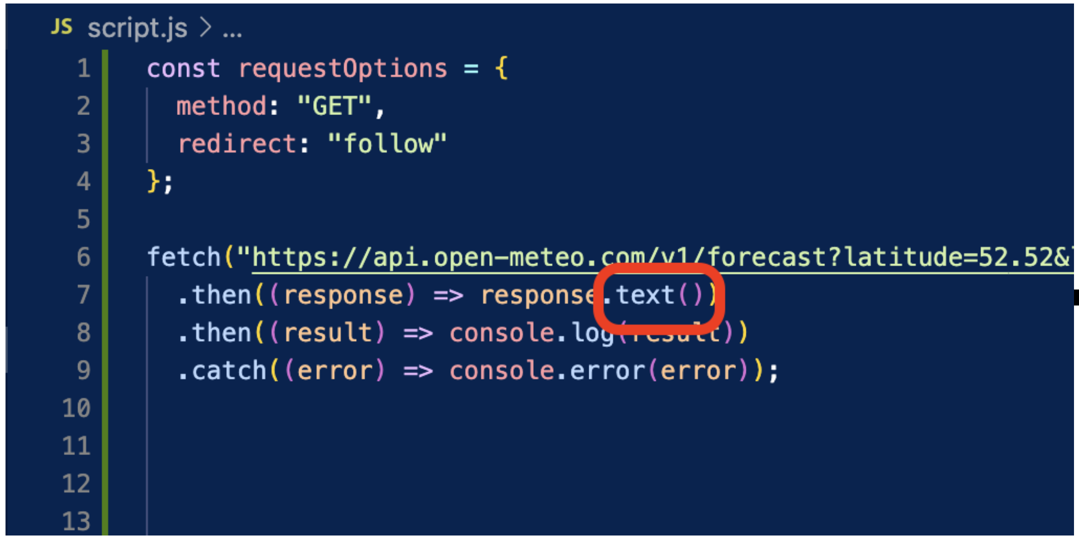


```javascript
async function query(data) {
    const response = await fetch(
        "https://router.huggingface.co/v1/chat/completions",
        {
            headers: {
                Authorization: `Bearer ${process.env.HF_TOKEN}`,
                "Content-Type": "application/json",
            },
            method: "POST",
            body: JSON.stringify(data),
        }
    );
    const result = await response.json();
    return result;
}

query({
    messages: [
        {
            role: "user",
            content: "What is the capital of France?",
        },
    ],
    model: "meta-llama/Llama-3.1-70B-Instruct:fireworks-ai",
}).then((response) => {
    console.log(JSON.stringify(response));

});
```

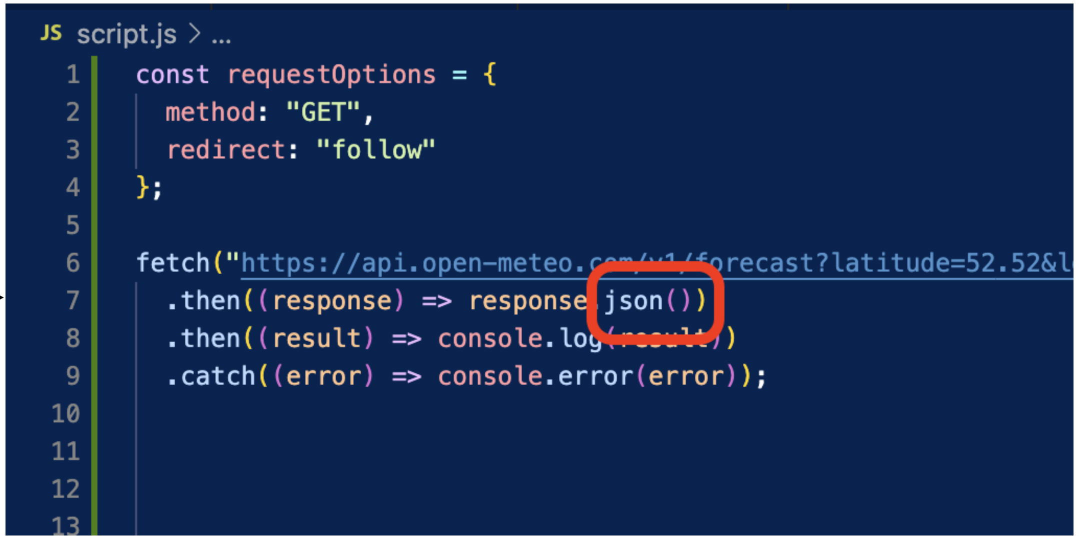


```javascript
var botReply = "";
async function query(data) {
    const response = await fetch(
        "https://router.huggingface.co/v1/chat/completions",
        {
            headers: {
                Authorization: `Bearer ${HF_TOKEN}`,
                "Content-Type": "application/json",
            },
            method: "POST",
            body: JSON.stringify(data),
        }
    );
    const result = await response.json();
    return result;
}

query({
    messages: [
        {
            role: "user",
            content: "What is the capital of Mississippi?",
        },
    ],
    model: "meta-llama/Llama-3.1-70B-Instruct:fireworks-ai",
}).then((response) => {
    console.log(JSON.stringify(response));
    botReply = response.choices[0].message.content;
    console.log(botReply);
    setText("output-display", botReply);
});
```

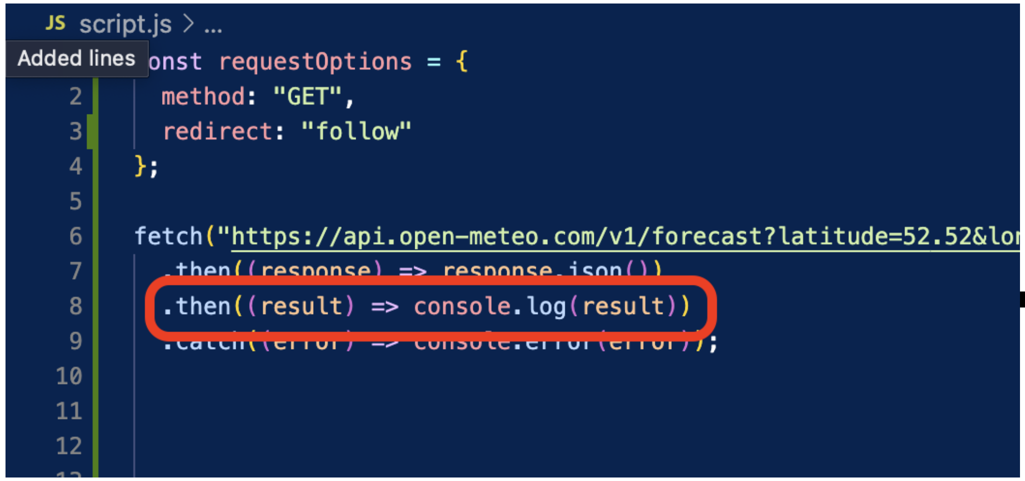
Cheat Sheet for Postman API generated code



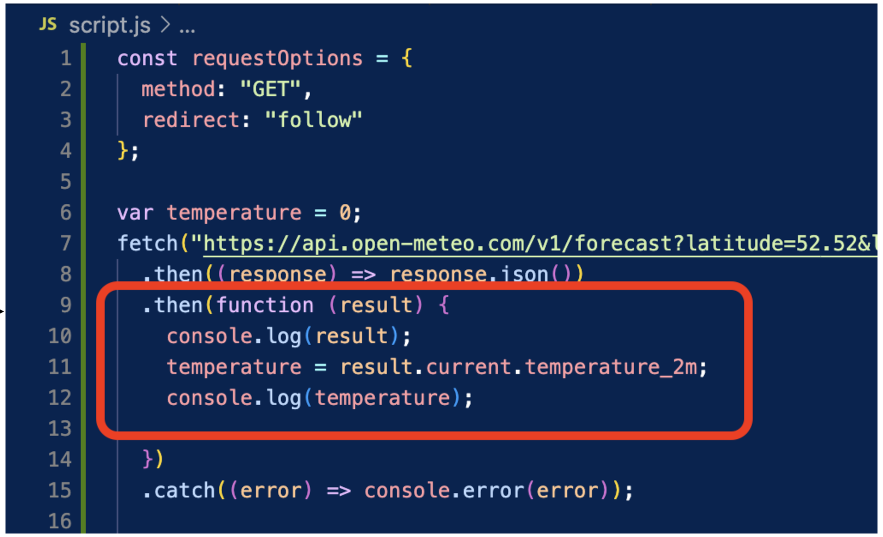
```javascript
const requestOptions = {
  method: "GET",
  redirect: "follow"
};

fetch("https://api.open-meteo.com/v1/forecast?latitude=52.52&
  .then((response) => response.text())
  .then((result) => console.log(result))
  .catch((error) => console.error(error));
```



```javascript
const requestOptions = {
  method: "GET",
  redirect: "follow"
};

fetch("https://api.open-meteo.com/v1/forecast?latitude=52.52&l
  .then((response) => response.json())
  .then((result) => console.log(result))
  .catch((error) => console.error(error));
```



```javascript
onst requestOptions = {
  method: "GET",
  redirect: "follow"
};

fetch("https://api.open-meteo.com/v1/forecast?latitude=52.52&lon
  .then((response) => response.json())
  .then((result) => console.log(result))
  .catch((error) => console.error(error));
```



```javascript
const requestOptions = {
  method: "GET",
  redirect: "follow"
};

var temperature = 0;
fetch("https://api.open-meteo.com/v1/forecast?latitude=52.52&l
  .then((response) => response.json())
  .then(function (result) {
    console.log(result);
    temperature = result.current.temperature_2m;
    console.log(temperature);

  })
  .catch((error) => console.error(error));
```