27.06.24 15:56

CSCI 165
Introduction to the Internet and the World Wide Web
Lecture 5: Javascript 2

Jetic Gū
2024 Summer Semester (S2)

Overview

- Focus: Course Introduction
- Architecture: WWW
- Core Ideas:
    1. Numerical Calculations, Variables, Function
    2. Data Types
    3. Changing Elements using Javascript

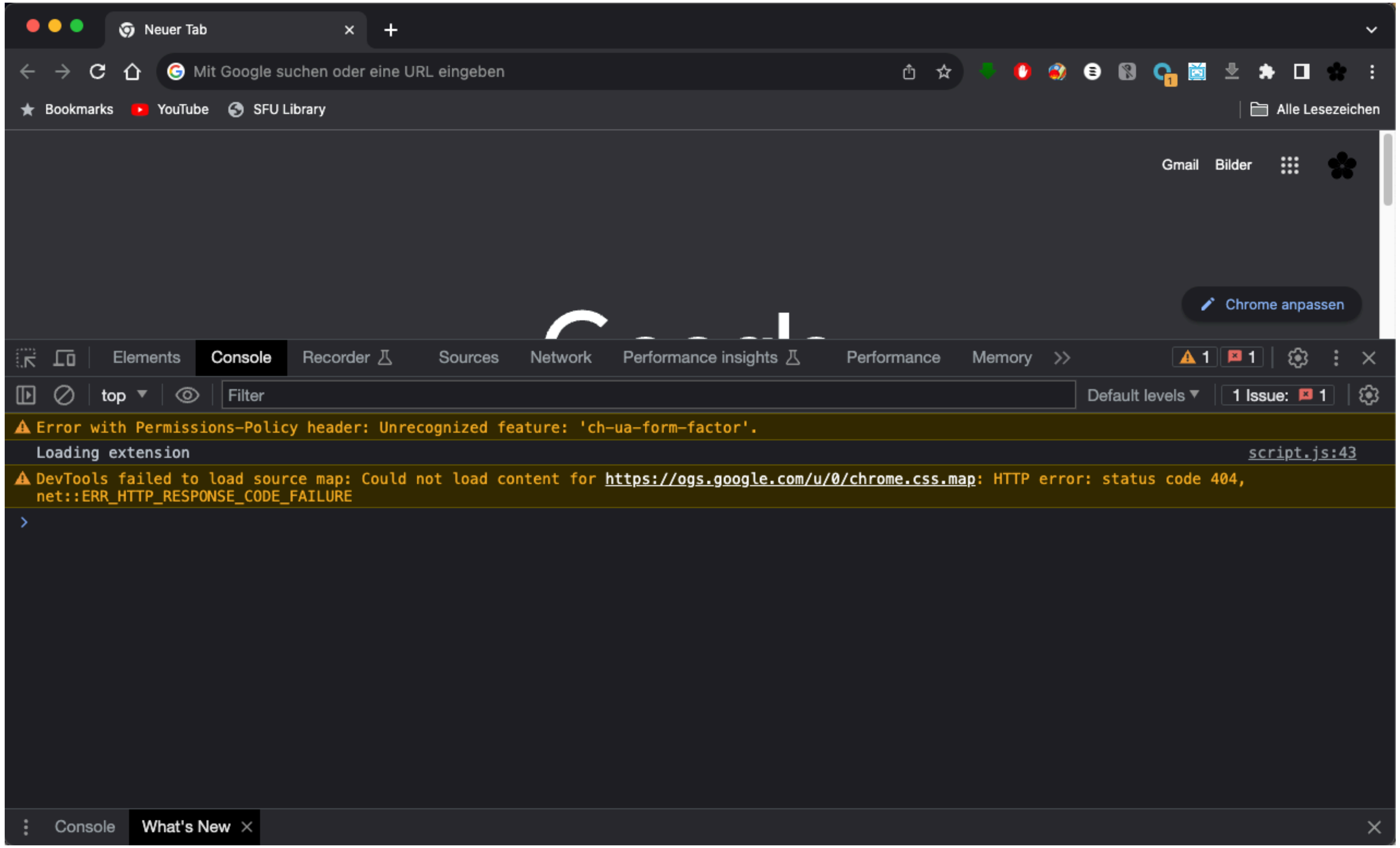
Calculation in Javascript

- Javascript can carry out some basic calculation
- First, let's take a look at a console
  - Every modern browser allows you to access the console in the developer tool

Concept

P1
Javascript

Calculation in Javascript



```
Error with Permissions-Policy header: Unrecognized feature: 'ch-ua-form-factor'.
Loading extension                                                              script.js:43
DevTools failed to load source map: Could not load content for https://ogs.google.com/u/0/chrome.css.map: HTTP error: status code 404,
net::ERR_HTTP_RESPONSE_CODE_FAILURE
>
```

1. `Console` tool in Google Chrome, right next to `Elements`

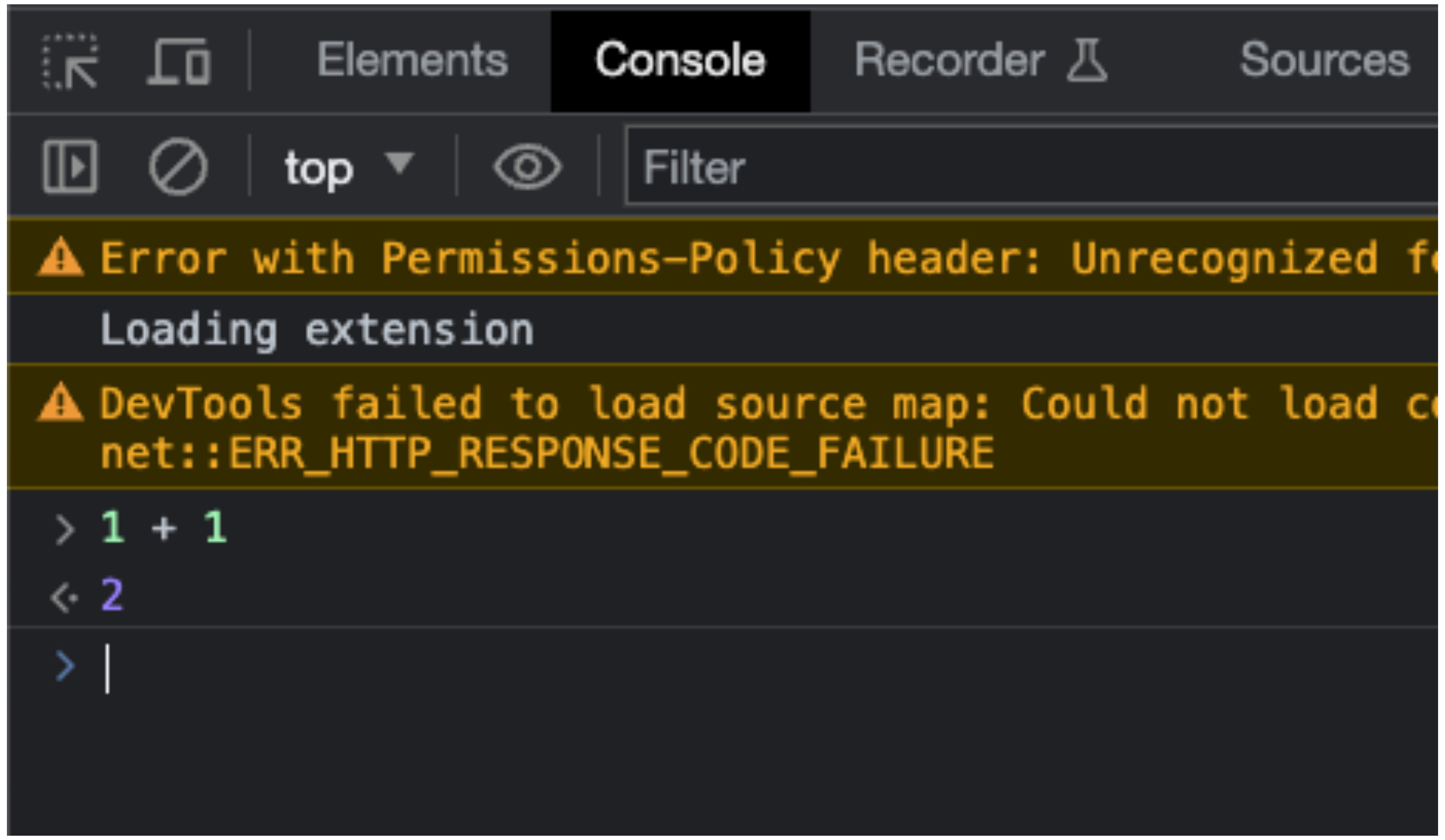
Technical

P1
Javascript

Calculation in Javascript

- You can execute Javascript function calls, as well as general statements etc. here.

- Try some basic calculation of numbers. These are called Expressions[1]



```
⚠ Error with Permissions-Policy header: Unrecognized f
  Loading extension
⚠ DevTools failed to load source map: Could not load c
  net::ERR_HTTP_RESPONSE_CODE_FAILURE
> 1 + 1
<· 2
> |
```

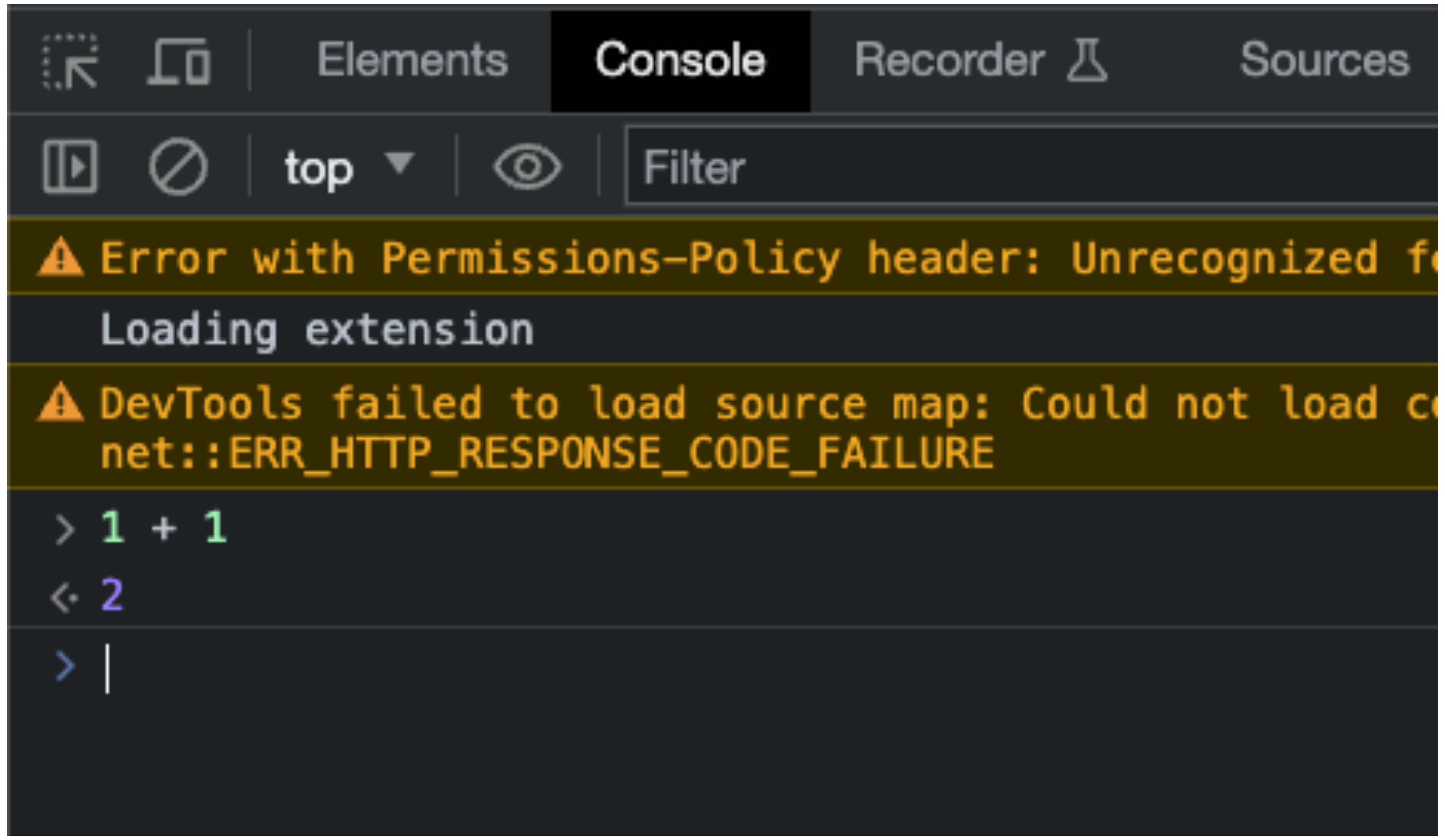
Technical

1. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Expressions_and_Operators

P1
Javascript

Calculation in Javascript

- Expressions include:
  - Mathematical expressions;
  - Logical Expressions; as well as
  - Function call returns
- Expressions themselves are NOT full statements, they form part of the statement such as assignments



```
⚠ Error with Permissions-Policy header: Unrecognized fe
  Loading extension
⚠ DevTools failed to load source map: Could not load co
  net::ERR_HTTP_RESPONSE_CODE_FAILURE
> 1 + 1
<· 2
> |
```

Technical

1. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Expressions_and_Operators

Calculation in Javascript

```
x = 10;
y = x * x + 2 * x + 1;
```

- These two are full statements. They must end with semicolon.
- x and y here are variables.
- When **executed** (by pressing enter), the **expressions** are **evaluated** (calculated), and their results **assigned** to Variables `x` and `y`
- If variables to be assigned doesn’t exist, they will first be created (**declared**) in memory
- To checkout the values of already declared variables, type the name of the variable in the console, then press enter

P1
Javascript

Calculation in Javascript

- Write a new function in the console like this:

```
circumference_circle = function(radius) {
      result = radius * 2 * 3.1415926;
      console.log("With radius", radius);
      console.log("The circumference is", result);
};
```

- Then, execute it by calling this function:

```
circumference_circle(10);
```

Example

Calculation in Javascript

```
circumference_circle = function(radius) {
     result = radius * 2 * 3.1415926;
     console.log("With radius", radius);
     console.log("The circumference is", result);
};
```

- This is a variable, which is declared as a **function** that can be called

Example

Calculation in Javascript

```
circumference_circle = function(radius) {
        result = radius * 2 * 3.1415926;
        console.log("With radius", radius);
        console.log("The circumference is", result);
};
```

- This is a function declaration
- A function declaration has 2 parts

Example

Calculation in Javascript

```
circumference_circle = function(radius) {
     result = radius * 2 * 3.1415926;
     console.log("With radius", radius);
     console.log("The circumference is", result);
};
```

- This is a function declaration
- A function declaration has 2 parts

1. Argument list: this will be the part within the parenthesis. These **arguments** are used as **variables** inside the function, with values given during function calls.

Example

Calculation in Javascript

```
say_hello = function(my_name, your_name) {
    console.log("Hello", your_name, "from", my_name);
};
```

- This is a function declaration
- A function declaration has 2 parts

1. Argument list: a function can have multiple arguments, separated by comma

Calculation in Javascript

```
circumference_circle = function(radius) {
        result = radius * 2 * 3.1415926;
        console.log("With radius", radius);
        console.log("The circumference is", result);
};
```

- This is a function declaration
- A function declaration has 2 parts

2. Subroutine: you can write standard Javascript statements here
   These code will be executed whenever the function is called

Example

Calculation in Javascript

```
circumference_circle = function(radius) {
        result = radius * 2 * 3.1415926;
        console.log("With radius", radius);
        console.log("The circumference is", result);
};
```

- This is a function declaration
- A function declaration has 2 parts

2. Subroutine: subroutines starts and ends with curly brackets
   Inside, you must write lines of **full statements**

Example

P1
Javascript

console.log()

```
console.log("With radius", radius);
```

- This is also a function
- The arguments of this function call will be **printed** (outputted) in the console
- `"With radius"` is a string value, it gets printed as is
- `radius` is a variable, it’s internal value will be printed
- `console.log()` can have as many arguments as you want

Example

P1
Javascript

`console.log()`

- Regular users usually won’t access the console
- This is usually used for development and debugging
- That being said… don’t write your credit card number in your console, nor in your code, nor anywhere others can see

Concept

Exercise 1

- Create `script.js` with the following inside
  - Declare a function called `calc_tax`, that takes `price` as input
  - Here's what you should output using `console.log` in 3 lines:
    - Original Price: `price`
    - GST is 7% of the `price`
    - PST is 5% of the `price`
    - Total cost is `price` plus `GST` and `PST`
- Include `script.js` in an `index.html`, open `index.html` in your browser, call the following in console:
  `calc_tax(100);`

```
> calc_tax(200);
  Price: CAD 200
  GST:   CAD 14
  PST:   CAD 10
  Total: CAD 224
>
```

Exercise

P1
Javascript

Exercise 2

- At the end of your `script.js`'s function, add this:
  ```
  return price + gst + pst;
  ```

```
    ...
    return price + gst + pst;
}
```

- Open `index.html` in your browser, call the following in console:
  ```
  x = calc_tax(100);
  ```
  - Now check the value of `x`, what do you get?
  - This statement is called `return`, it allows a function call to be used in an **expression** to provide values.
  - The default return value is `undefined`, which doesn't mean anything

Exercise

Javascript Data Types

- We have now encountered 3 data types
  - Number: integers and float numbers, e.g. `100`, `3.14`
  - String: text values, e.g. `"Hello World!"`
  - Functions: that can be called, e.g. `console.log`, `alert`, etc.
- Javascript **Variables** carry data **values** of various **types**, typing is **dynamic**
  - **Dynamic**: the same variable can take on values of all types on the fly (C and C++ doesn't support this, Python and Javascript does)

Changing Appearance

- Change HTML content (e.g. whatever is between the opening and closing tag)
  - `document.getElementById("content").innerHTML = "New text!";`
    This changes element with id `content`'s HTML content to `New text!`
- Change CSS Style
  - `document.getElementById("content").style.color = "blue";`
    This changes element with id `content`'s colour to `blue`

Changing Appearance

- `document.getElementById("content")`
  - This returns the entire element with `id="content"`
  - `document.getElementById("content").innerHTML` returns just the content betweeen the opening and closing tag
  - `document` here is called an *object*
  - `getElementById` here is called a *method* of that above object
  - Different objects have different methods

Input Type HTML Elements

```
<input type="text" id="price">
```

- This creates a textbox, for which the user can input text
- You can retrieve the value from it through Javascript

```
x = document.getElementById("price").value;
```

- This allows you to get the input text as a `string`

Input Type HTML Elements

```
price = parseFloat(x);
```

- This converts `x`'s `string` value to a number
- and now, you can assemble the whole webpage!

Exercise 3

- In your `index.html`, include a Textbox and relevant text to prompt the user to type in a price e.g. code on the right

- Upon pressing the button, the user should see the calculated price appear on the webpage

```
<html>
<head>
    <title>Tax Calculator</title>
</head>
<body>
    <input id="price" type="text">
    <button onclick="button()">
        Calculate Tax</button>
    <p id="results"></p>
</body>
</html>
```

Exercise