Variabler og datatyper

Datamatiker

KØBENHAVNS ERHVERVSAKADEMI

Agenda

- Formål
- Variabler: const, let & var
- Scopes, globale & lokale variabler
- Datatyper
- Konvertering, konkatenering & sammenligning

Øvelser

- “use strict”;
- Working with variables
- Hello JavaScript
- CRUD App
- Name things right (or wrong?)
- Datatyper
- String quotes
- Numeric conversion
- String conversion
- Numeric conversion 2
- Concatenation looks like addition
- Type Conversion


Well done 👏

“The cool thing about JavaScript is that you can create stuff that will put a smile on your face, as a developer.

You can create stuff and it will visually look like something, and it'll do stuff, and it makes you feel good, **it makes you fall in love with programming.**”

Lex Fridman
Computer Scientist
https://www.youtube.com/watch?v=GLhyjVZp0cw&t=252s

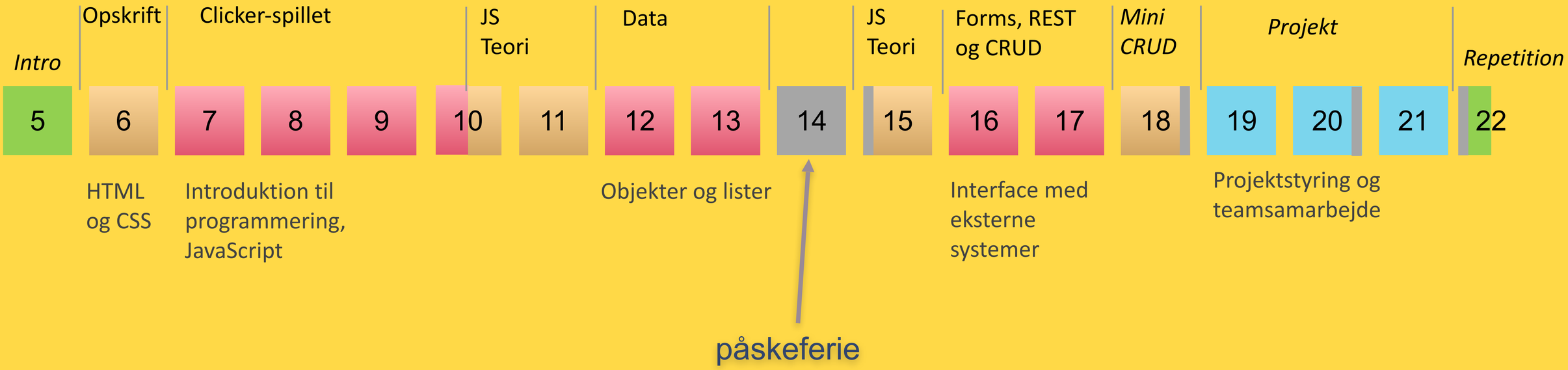
Computer science student

Senior developer, 10+ years experience

https://www.instagram.com/p/BxWAgatgSmn/

1. SEMESTERS STRUKTUR

Semesteret består af 16 undervisningsuger (inkl. en masse helligdage)



Eksamen ligger 12.-14. juni

kea
KØBENHAVNS ERHVERVSAKADEMI

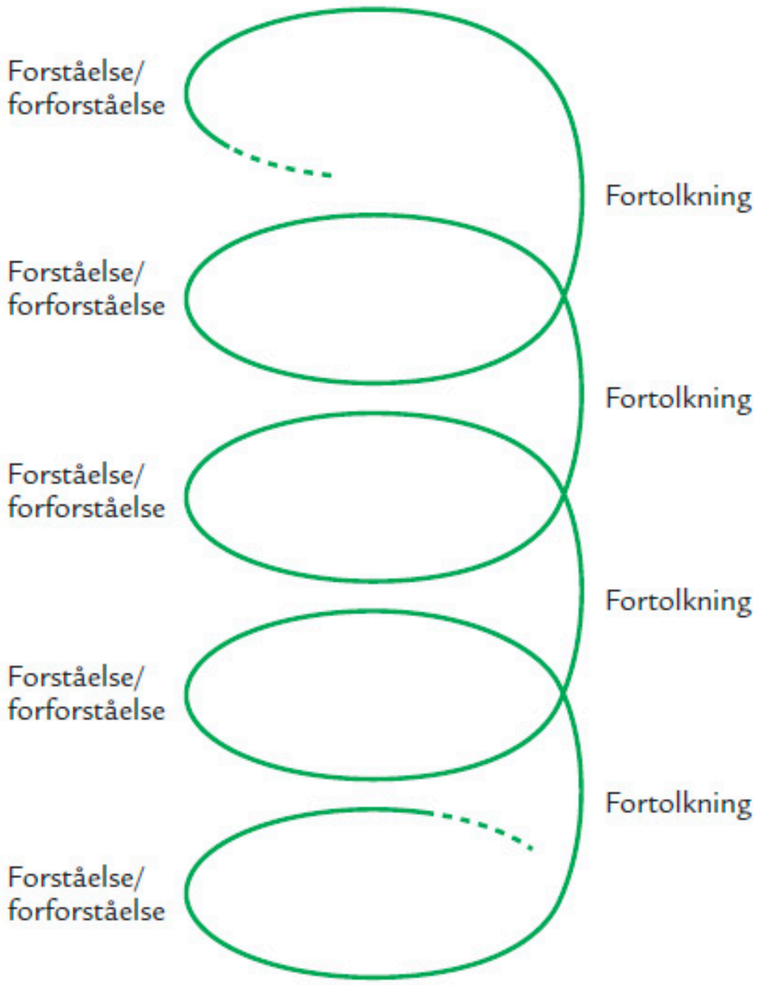
Formål

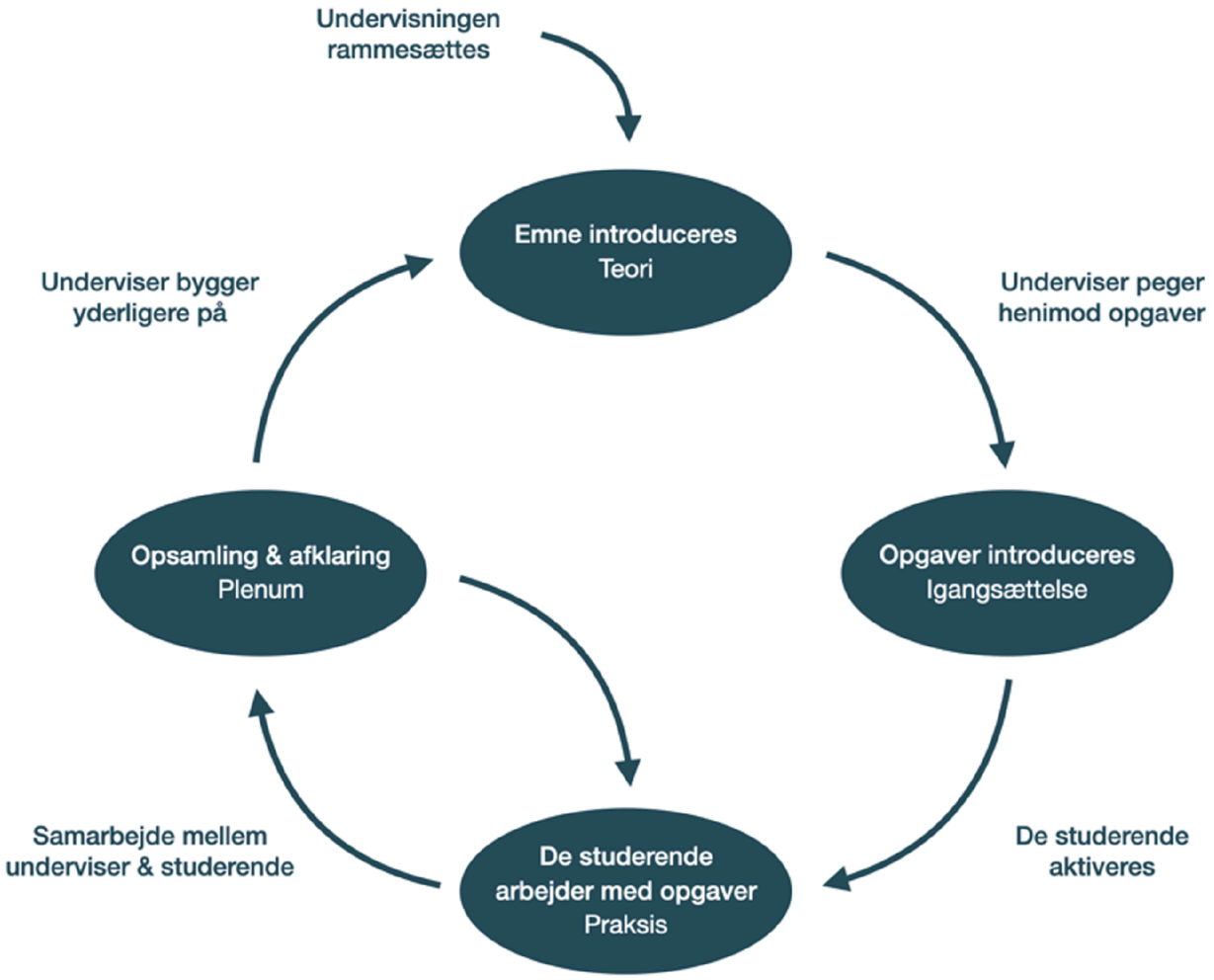
- Opsamling og dybere viden om variabler og datatyper
- Viden om konvertering og konkatenering
- Hands-on på mindre opgaver

DU SKAL IKKE BARE VIDE.
DU SKAL KUNNE.

DU SKAL IKKE BARE VIDE.
DU SKAL KUNNE.

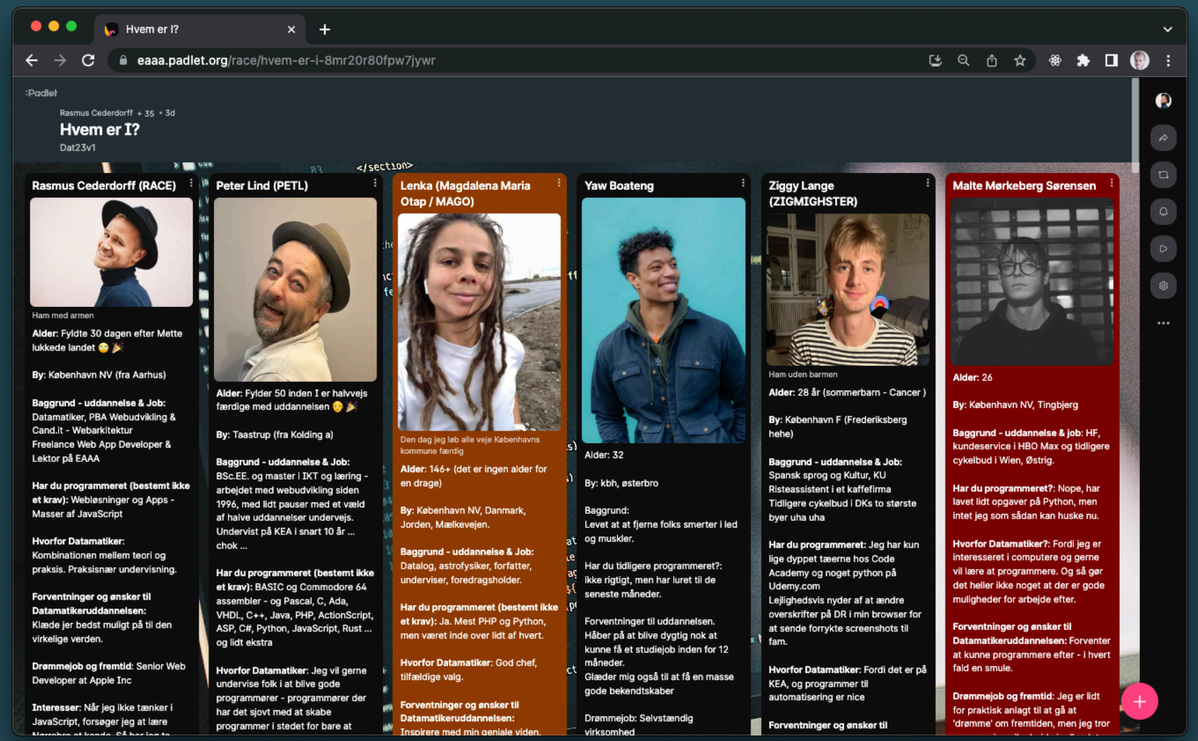
kea
KØBENHAVNS ERHVERVSAKADEMI

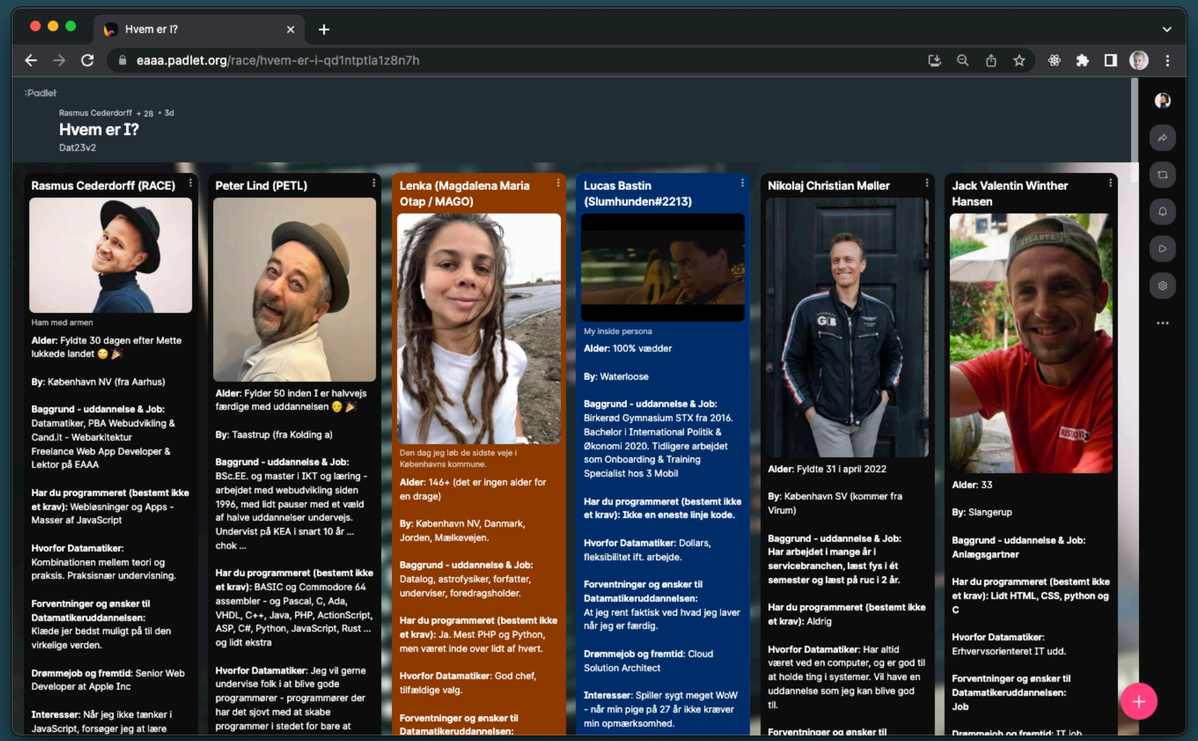

Forståelse/
forforståelse
Fortolkning
Forståelse/
forforståelse
Fortolkning
Forståelse/
forforståelse
Fortolkning
Forståelse/
forforståelse
Fortolkning
Forståelse/
forforståelse


Undervisningen
rammesættes
Emne introduceres
Teori
Underviser peger
henimod opgaver
Opgaver introduceres
Igangsættelse
De studerende
aktiveres
De studerende
arbejder med opgaver
Praksis
Samarbejde mellem
underviser & studerende
Opsamling & afklaring
Plenum
Underviser bygger
yderligere på


Programming knowledge
JavaScript

Variabler?


Hvem er I?
eaaa.padlet.org/race/hvem-er-i-8mr20r80fpw7jywr
Hvem er I?
Dat23v1
Rasmus Cederdorff (RACE)
Ham med armen
Alder: Fyldte 30 dagen efter Mette lukkede landet
By: København NV (fra Aarhus)
Baggrund - uddannelse & Job: Datamatiker, PBA Webudvikling & Cand.it - Webarkitektur Freelance Web App Developer & Lektor på EAAA
Har du programmeret (bestemt ikke et krav): Weblesninger og Apps - Masser af JavaScript
Hvorfor Datamatiker: Kombinationen mellem teori og praksis. Praksisnær undervisning.
Forventninger og ønsker til Datamatikeruddannelsen: Klæde jer bedst muligt på til den virkelige verden.
Drømmejob og fremtid: Senior Web Developer at Apple Inc
Interesser: Når jeg ikke tænker i JavaScript, forsøger jeg at lære
Peter Lind (PETL)
Alder: Fylder 50 inden I er halvvejs færdige med uddannelsen
By: Taastrup (fra Kolding a)
Baggrund - uddannelse & Job: BSc.EE. og master i IKT og læring - arbejdet med webudvikling siden 1996, med lidt pauser med et væld af halve uddannelser undervejs. Undervist på KEA i snart 10 år ... chok ...
Har du programmeret (bestemt ikke et krav): BASIC og Commodore 64 assembler - og Pascal, C, Ada, VHDL, C++, Java, PHP, ActionScript, ASP, C#, Python, JavaScript, Rust ... og lidt ekstra
Hvorfor Datamatiker: Jeg vil gerne undervise folk i at blive gode programmører - programmører der har det sjovt med at skabe programmer i stedet for bare at
Lenka (Magdalena Maria Otap / MAGO)
Den dag jeg løb alle veje Københavns kommune færdig
Alder: 146+ (det er ingen alder for en drage)
By: København NV, Danmark, Jorden, Mælkevejen.
Baggrund - uddannelse & Job: Datalog, astrofysiker, forfatter, underviser, foredragsholder.
Har du programmeret (bestemt ikke et krav): Ja. Mest PHP og Python, men været inde over lidt af hvert.
Hvorfor Datamatiker: God chef, tilfældige valg.
Forventninger og ønsker til Datamatikeruddannelsen:
Yaw Boateng
Alder: 32
By: kbh, østerbro
Baggrund:
Levet at at fjerne folks smerter i led og muskler.
Har du tidligere programmeret?: ikke rigtigt, men har luret til de seneste måneder.
Forventninger til uddannelsen.
Håber på at blive dygtig nok at kunne få et studiejob inden for 12 måneder.
Glæder mig også til at få en masse gode bekendtskaber
Drømmejob: Selvstændig virksomhed
Ziggy Lange (ZIGMIGHSTER)
Ham uden barmen
Alder: 28 år (sommerbarn - Cancer)
By: København F (Frederiksberg hehe)
Baggrund - uddannelse & Job: Spansk sprog og Kultur, KU Risteassistent i et kaffefirma Tidligere cykelbud i DKs to største byer uha uha
Har du programmeret: Jeg har kun lige dyppet tæerne hos Code Academy og noget python på Udemy.com
Lejlighedsvis nyder af at ændre overskrifter på DR i min browser for at sende forrykte screenshots til fam.
Hvorfor Datamatiker: Fordi det er på KEA, og programmer til automatisering er nice
Forventninger og ønsker til
Malte Mørkeberg Sørensen
Alder: 26
By: København NV, Tingbjerg
Baggrund - uddannelse & job: HF, kundeservice i HBO Max og tidligere cykelbud i Wien, Østrig.
Har du programmeret?: Nope, har lavet lidt opgaver på Python, men intet jeg som sådan kan huske nu.
Hvorfor Datamatiker?: Fordi jeg er interesseret i computere og gerne vil lære at programmere. Og så gør det heller ikke noget at der er gode muligheder for arbejde efter.
Forventninger og ønsker til Datamatikeruddannelsen: Forventer at kunne programmere efter - i hvert fald en smule.
Drømmejob og fremtid: Jeg er lidt for praktisk anlagt til at gå at 'drømme' om fremtiden, men jeg tror


Hvem er I?
eaaa.padlet.org/race/hvem-er-i-qd1ntptla1z8n7h
Hvem er I?
Dat23v2
Rasmus Cederdorff (RACE)
Ham med armen
Alder: Fyldte 30 dagen efter Mette lukkede landet
By: København NV (fra Aarhus)
Baggrund - uddannelse & Job: Datamatiker, PBA Webudvikling & Cand.it - Webarkitektur Freelance Web App Developer & Lektor på EAAA
Har du programmeret (bestemt ikke et krav): Weblesninger og Apps - Masser af JavaScript
Hvorfor Datamatiker: Kombinationen mellem teori og praksis. Praksisnær undervisning.
Forventninger og ønsker til Datamatikeruddannelsen: Klæde jer bedst muligt på til den virkelige verden.
Drømmejob og fremtid: Senior Web Developer at Apple Inc
Interesser: Når jeg ikke tænker i JavaScript, forsøger jeg at lære
Peter Lind (PETL)
Alder: Fylder 50 inden I er halvvejs færdige med uddannelsen
By: Taastrup (fra Kolding a)
Baggrund - uddannelse & Job: BSc.EE. og master i IKT og læring - arbejdet med webudvikling siden 1996, med lidt pauser med et væld af halve uddannelser undervejs. Undervist på KEA i snart 10 år ... chok ...
Har du programmeret (bestemt ikke et krav): BASIC og Commodore 64 assembler - og Pascal, C, Ada, VHDL, C++, Java, PHP, ActionScript, ASP, C#, Python, JavaScript, Rust ... og lidt ekstra
Hvorfor Datamatiker: Jeg vil gerne undervise folk i at blive gode programmører - programmører der har det sjovt med at skabe programmer i stedet for bare at
Lenka (Magdalena Maria Otap / MAGO)
Den dag jeg løb de sidste veje i Københavns kommune.
Alder: 146+ (det er ingen alder for en drage)
By: København NV, Danmark, Jorden, Mælkevejen.
Baggrund - uddannelse & Job: Datalog, astrofysiker, forfatter, underviser, foredragsholder.
Har du programmeret (bestemt ikke et krav): Ja. Mest PHP og Python, men været inde over lidt af hvert.
Hvorfor Datamatiker: God chef, tilfældige valg.
Forventninger og ønsker til Datamatikeruddannelsen:
Lucas Bastin (Slumhunden#2213)
My inside persona
Alder: 100% vædder
By: Waterloose
Baggrund - uddannelse & Job: Birkerød Gymnasium STX fra 2016. Bachelor i International Politik & Økonomi 2020. Tidligere arbejdet som Onboarding & Training Specialist hos 3 Mobil
Har du programmeret (bestemt ikke et krav): Ikke en eneste linje kode.
Hvorfor Datamatiker: Dollars, fleksibilitet ift. arbejde.
Forventninger og ønsker til Datamatikeruddannelsen: At jeg rent faktisk ved hvad jeg laver når jeg er færdig.
Drømmejob og fremtid: Cloud Solution Architect
Interesser: Spiller sygt meget WoW - når min pige på 27 år ikke kræver min opmærksomhed.
Nikolaj Christian Møller
Alder: Fyldte 31 i april 2022
By: København SV (kommer fra Virum)
Baggrund - uddannelse & Job: Har arbejdet i mange år i servicebranchen, læst fys i ét semester og læst på ruc i 2 år.
Har du programmeret (bestemt ikke et krav): Aldrig
Hvorfor Datamatiker: Har altid været ved en computer, og er god til at holde ting i systemer. Vil have en uddannelse som jeg kan blive god til.
Jack Valentin Winther Hansen
Alder: 33
By: Slangerup
Baggrund - uddannelse & Job: Anlægsgartner
Har du programmeret (bestemt ikke et krav): Lidt HTML, CSS, python og C
Hvorfor Datamatiker: Erhvervsorienteret IT udd.
Forventninger og ønsker til Datamatikeruddannelsen: Job

fullName = “Peter Lind”

initials = “petl”

age = 39

city = “Taastrup”

background = “BSc.EE. og master i IKT og læring …”

image = “https://share.cederdorff.com/images/petl.webp”

…

Variables

... are used to store data (values, objects, collections) in the memory

Variables

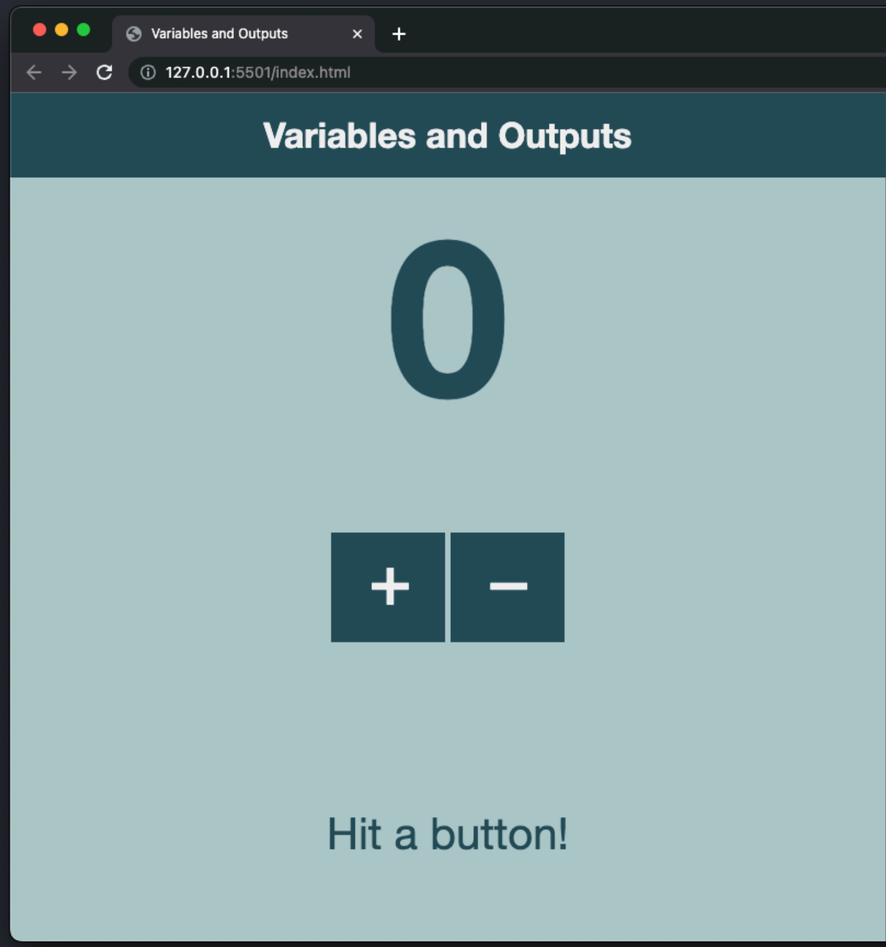
... a simple way to store, get and set data in your code.

Variables

- Hvilke variabler har du anvendt i Clicker-spillet? Og hvor?
- Brug to minutter på at gennemse din kode og noter dig anvendelsen af variabler.
- Forklar dernæst sidemakkeren om din anvendelse af variabler og hvorfor du havde brug for variabler.

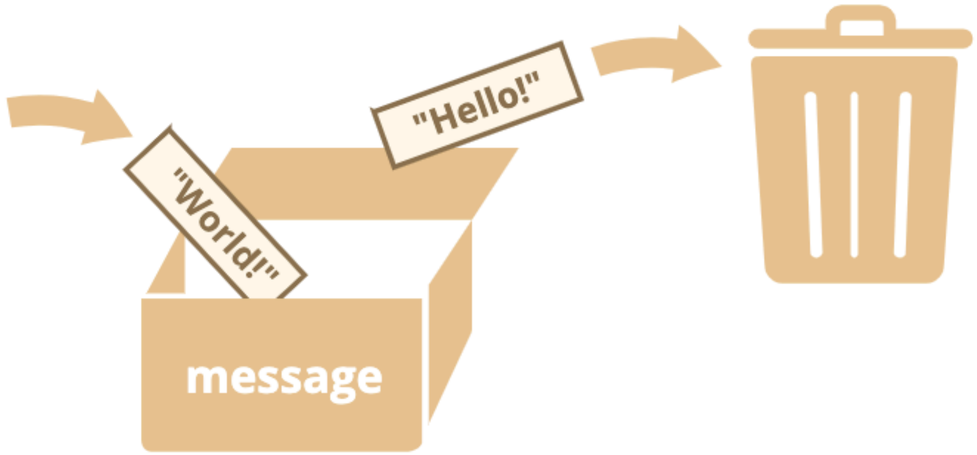
“Tælle” på en variabel

```
let variableName = 0;
variableName++; //lægger 1 til den nuværende værdi
variableName--; //trækker 1 fra den nuværende værdi
variableName += x; //lægger værdien x til den nuværende værdi
variableName -= x; //trækker værdien x fra den nuværende værdi
```



Variable

A variable is a “named storage” and is stored in the memory of the browser.



We can change the value of the variables as many times as we want.

“use strict”;

Enables all features of modern JavaScript

https://javascript.info/strict-mode

```
"use strict";

// your awesome JavaScript
// code goes here ...
```

“use strict”;

- Add it to the top of your script.
- ALWAYS!
- It helps us
- Gives us more errors and helps us to debug our code.

https://javascript.info/strict-mode

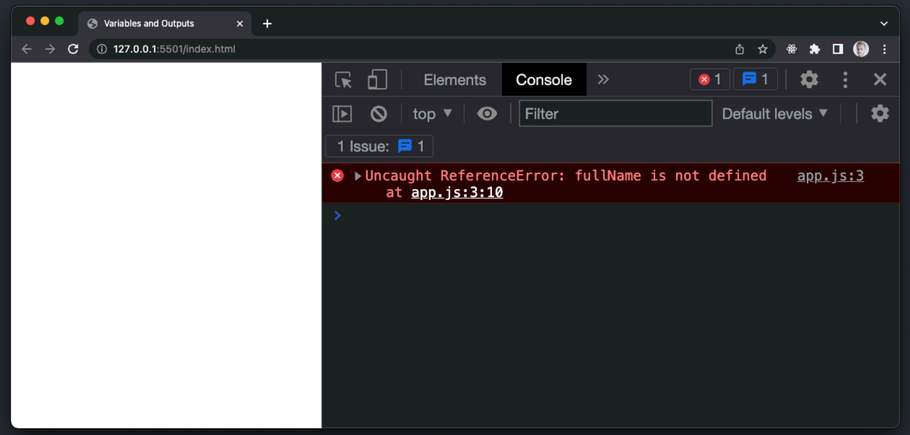
“use strict”;

1. Run the code and see what happens.
2. Remove “use strict”; and see what happens.
3. Insert “use strict”; and see what happens.
4. Remove the `keyword` let and see what happens

```
"use strict";

let fullName = "Peter Lind";

console.log(fullName);
```

https://javascript.info/strict-mode

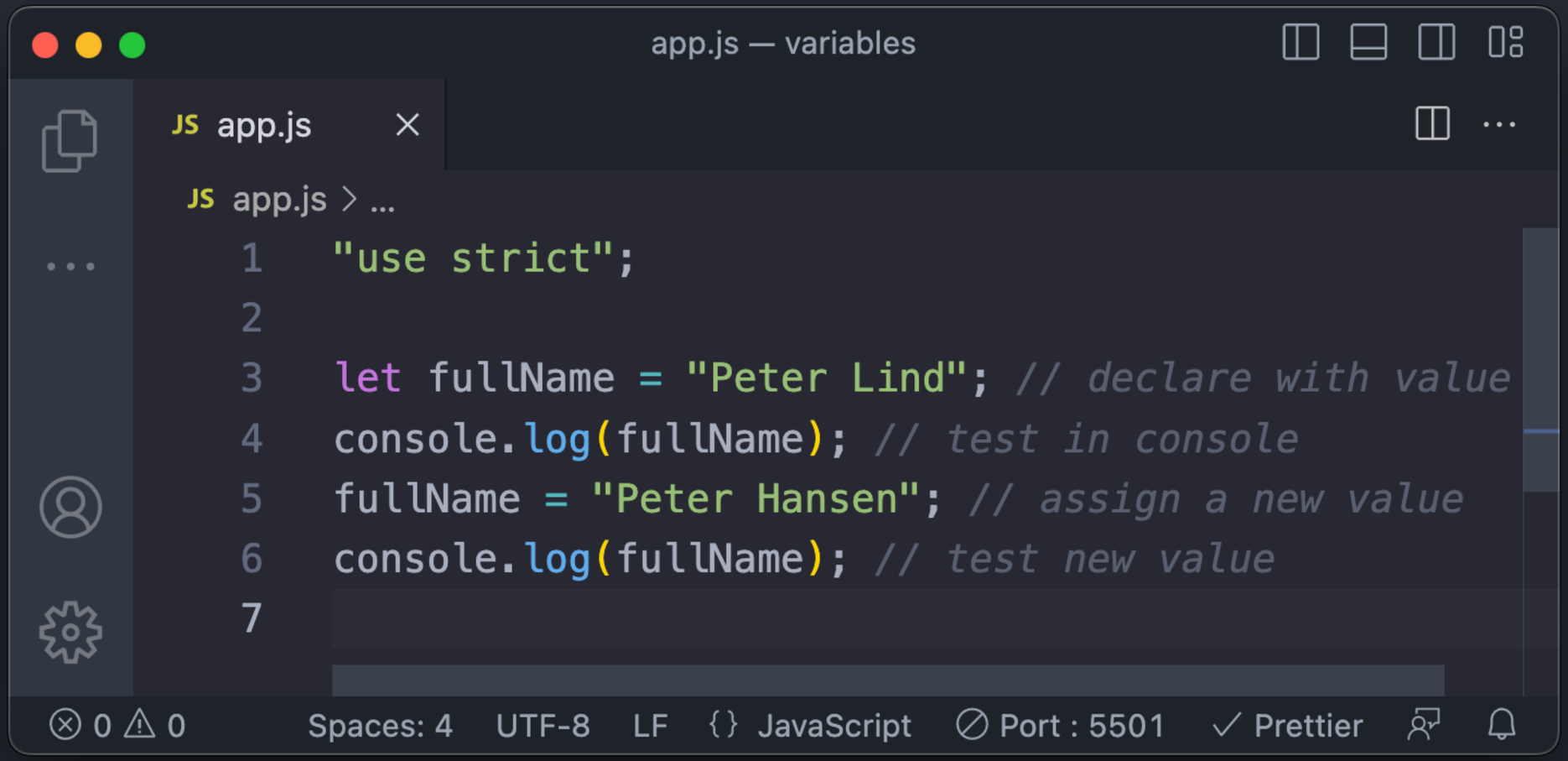
```
"use strict";

// your awesome JavaScript
// code goes here ...
```



We must declare our variables using keyword `let`, `const` or ~~`var`~~

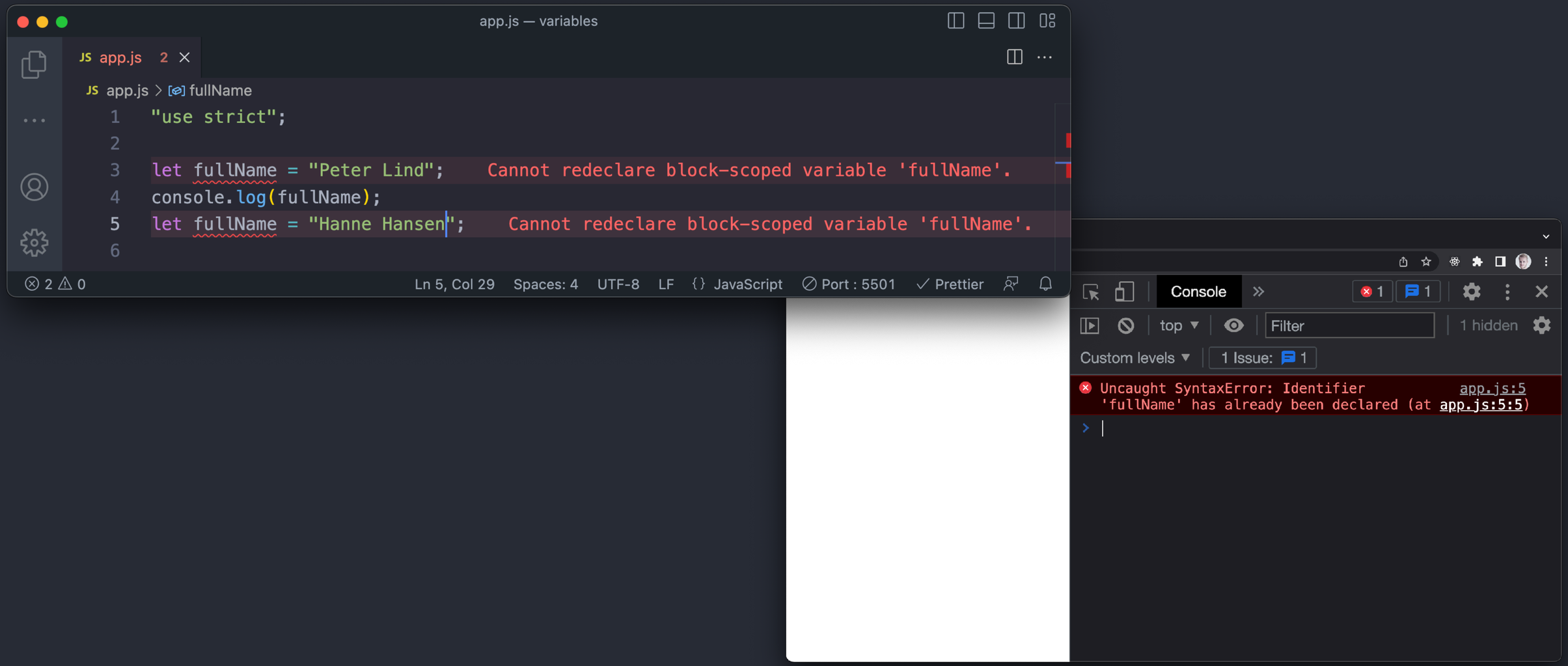
https://javascript.info/strict-mode

The keyword is only used for declaration



The keyword is only used for declaration

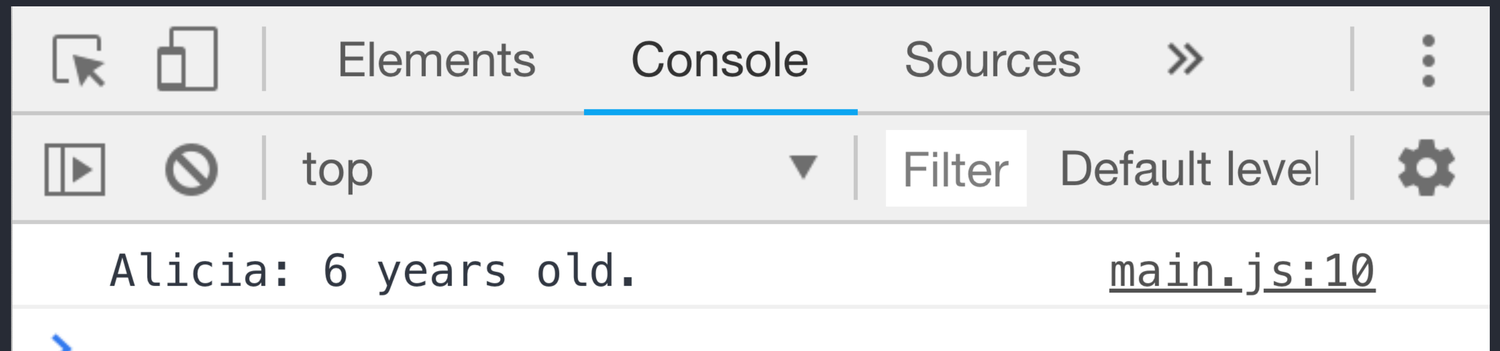
Why let and why “use strict”



Variables

Store data in the memory

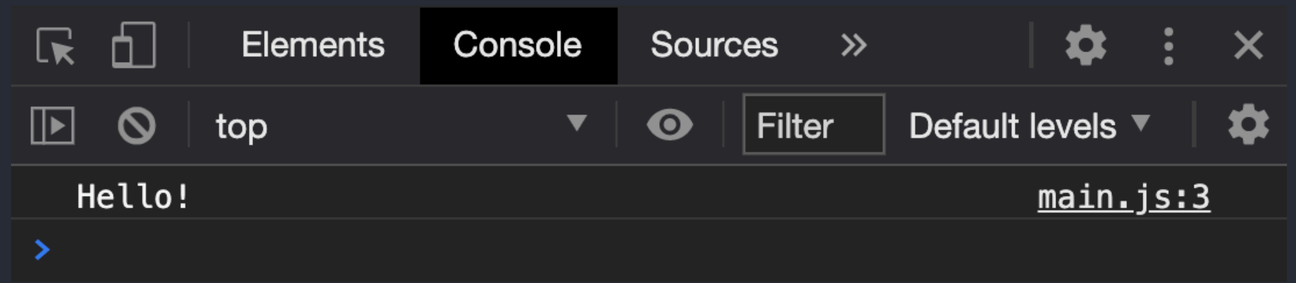
```
let name = "Alicia";
let age = 6;

console.log(name + ": " + age + " years old.");
```



Variables

Store data in the memory

```
let message = "Hello!";

console.log(message);
```



Variables

Store data in the memory

```
let message = "Hello!";
console.log(message);

message = "World!";
console.log(message);
```

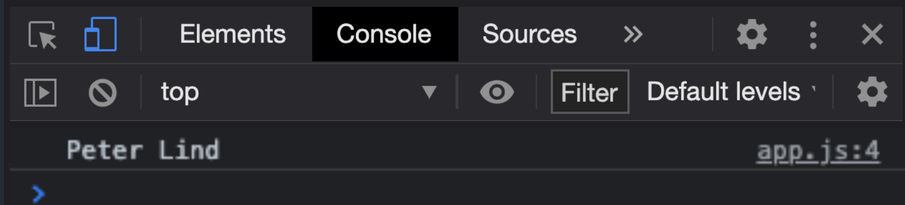
Working with variables

1. Declare two variables: `admin` and `firstName`.
2. Assign the value "John" to `firstName`.
3. Copy the value from `firstName` to `admin`.
4. Show the value of `admin` using `console.log` (must output “John”).

https://javascript.info/variables

Variables & UI

```
let fullName = "Peter Lind";
console.log(fullName);
document.querySelector("#fullName_container").textContent = fullName;
```



Variables & UI

```
let fullName = "Peter Lind";
console.log(fullName);
document.querySelector("#fullName_container").textContent = fullName;
```

```
<body>
    <header>
        <h1 id="fullName_container"></h1>
    </header>
    <script src="app.js"></script>
</body>
```

Variables & UI

```
let fullName = "Peter Lind";
console.log(fullName);
document.querySelector("#fullName_container").textContent = fullName;
```

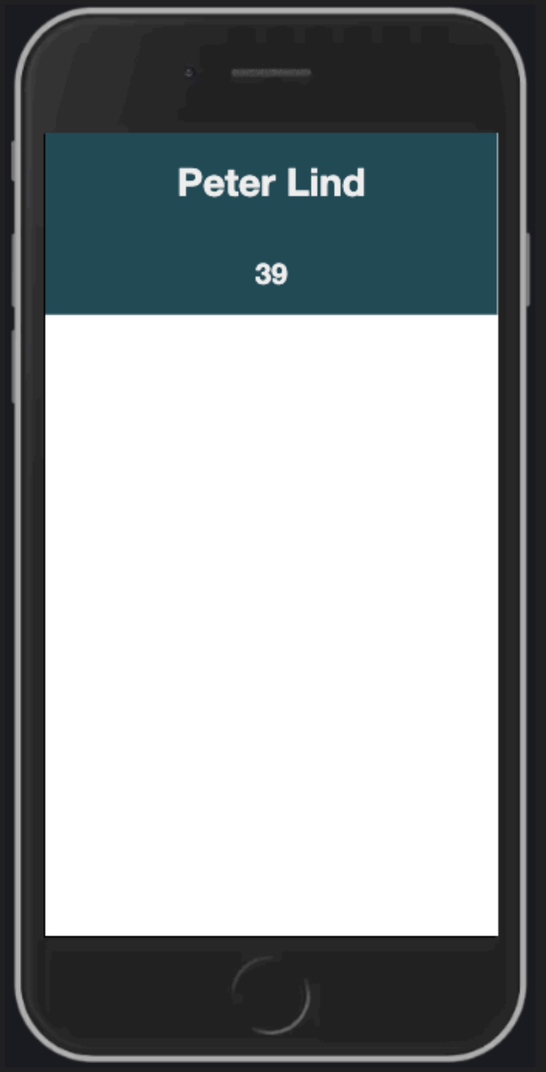
.textContent

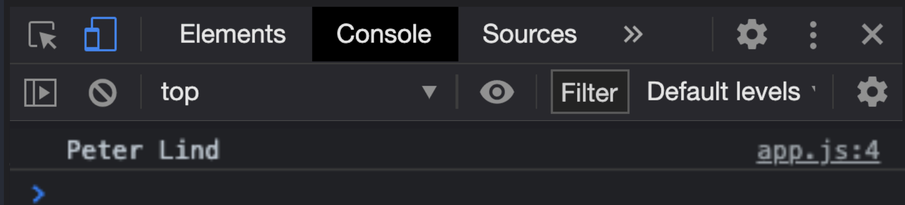
Get or set the text content of a given (HTML) element

https://www.w3schools.com/jsref/prop_node_textcontent.asp

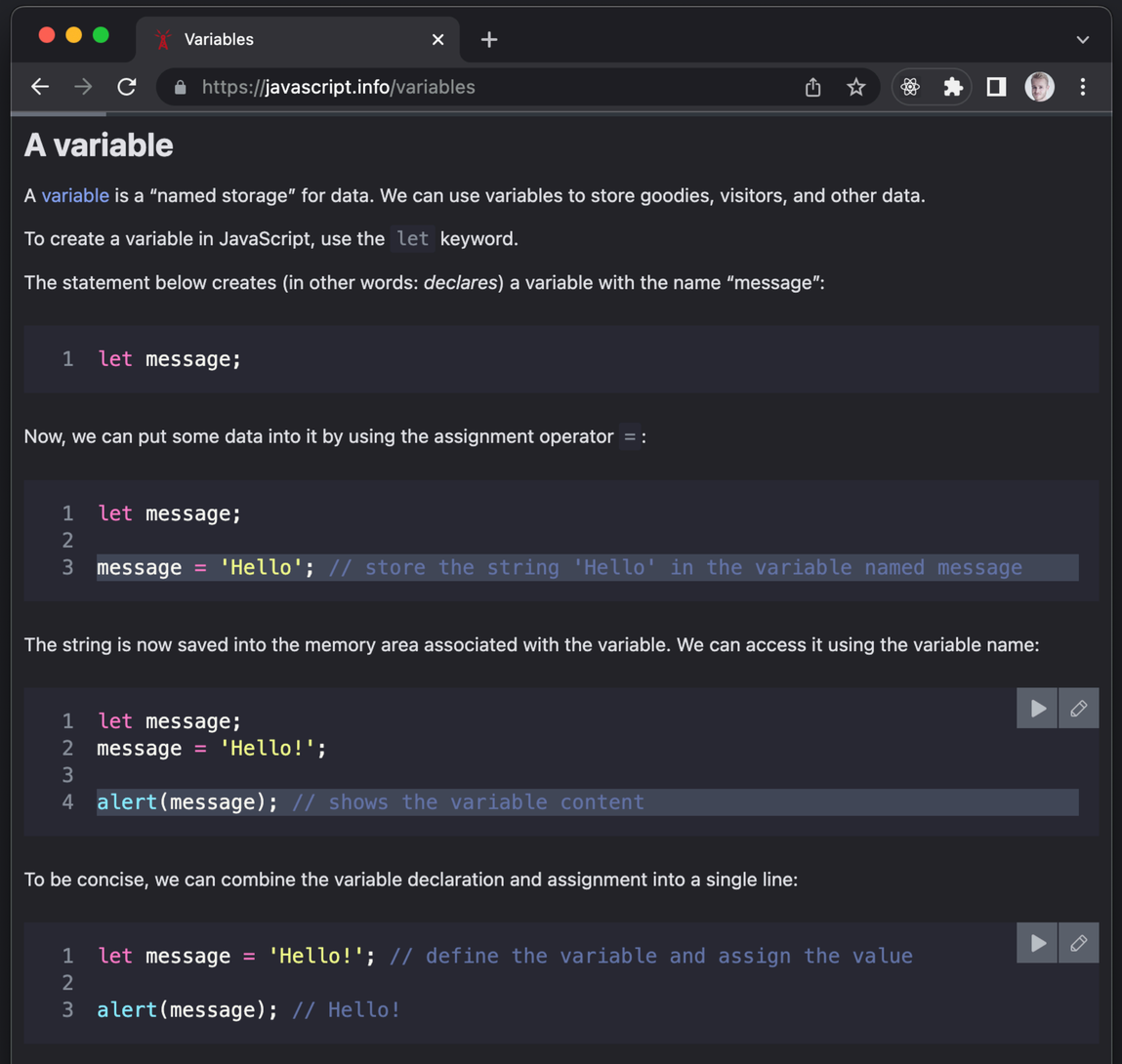
Variables & UI

```
let fullName = "Peter Lind";
let age = 39;
console.log(fullName, age);
document.querySelector("#fullName_container").textContent = fullName;
document.querySelector("#age_container").textContent = age;
```





JavaScript.info/Variables



Read, read,
read,
The docs 🤷

Variables

Template: `project-template` (GitHub) eller egen løsning

1. Definer en variabel

- Definer en variabel af typen `let`.
- Gem dit eller et andet navn i variablen.

```
let name = "Rasmus";
```

- Brug console.log(...) til at udskrive/ logge værdien til konsollen i browser dev tool.

```
console.log(name);
```

- Ændrer name variablen til en et andet navn.

```
name = "Ida";
```

- Log værdien igen.

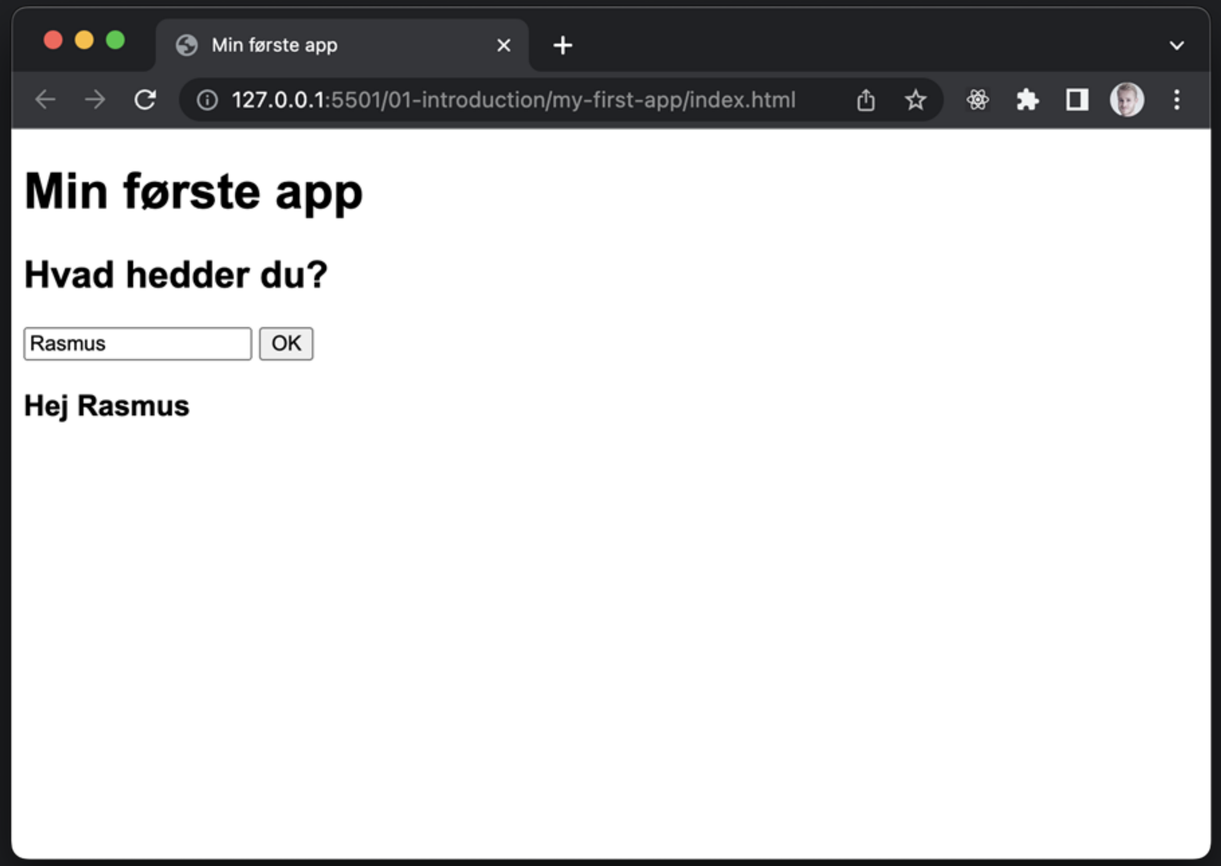
```
console.log(name);
```

- Vis name variablen på websitet (tilføj værdien til DOMen ved hjælp af `document.querySelector`:

```
document.querySelector("#content").innerHTML = name;
```

Hello JavaScript

1. Find øvelsen fra 2. introdag: Hello JavaScript
2. Genbesøg din kode.
3. Overvej hvordan du kan anvende variabler.
4. Implementer ændringerne.
5. Forbedrer, simplificerer eller gør det gin kode lettere at læse? Hvordan?
6. Er der anden ny viden, du kan anvende til at forbedre din løsning?



var **vs** let

THE DIFFERENCE IS THE SCOPING

VAR IS FUNCTION-WIDE OR GLOBAL SCOPE

LET IS BLOCK SCOPED

VAR TOLERATES REDECLARATION

https://javascript.info/variables
https://javascript.info/var

```
// Example 1
// “var” has no block scope
if (true) {
  var test1 = true; // use "var" instead of "let"
}
console.log(test1); // true, the variable lives after if

// Example 2
if (true) {
  let test2 = true; // use "let"
}
console.log(test2); // Error: test is not defined

// Example 3
for (var i = 0; i < 10; i++) {
  // ...
}
console.log(i); // 10, "i" is visible after loop, it's a global variable
```

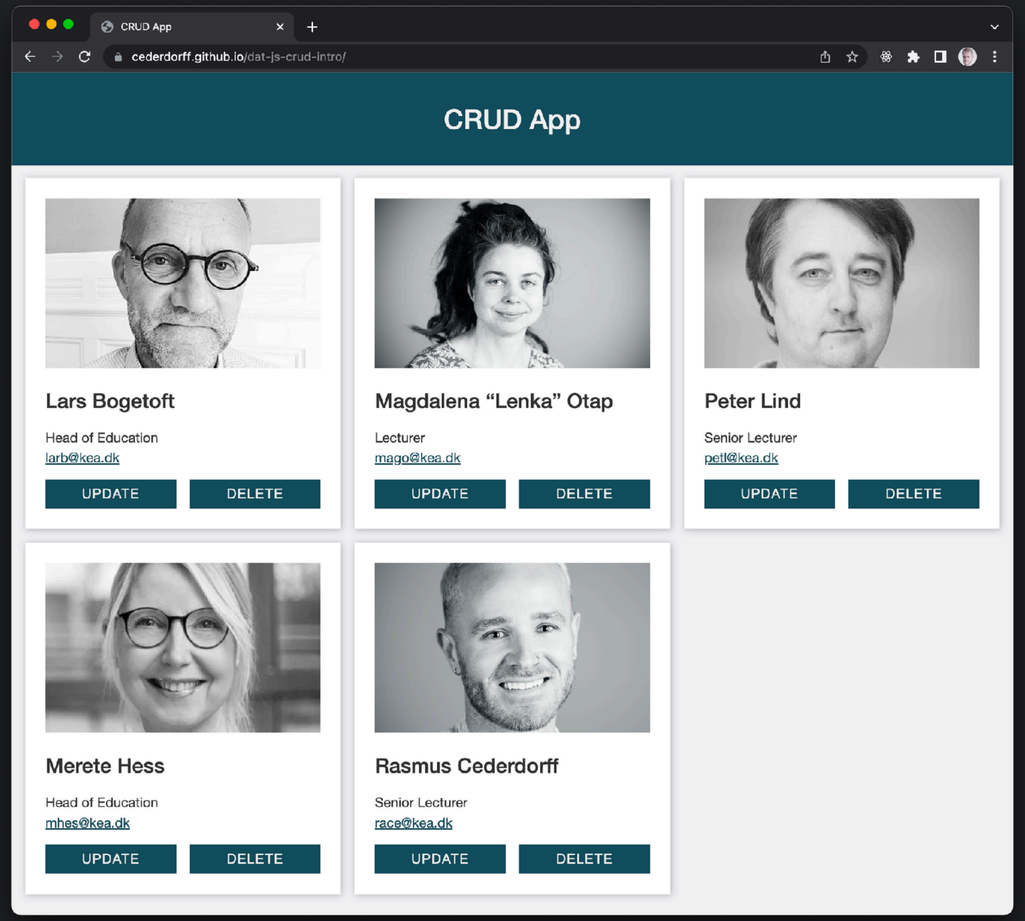
var-vs-let

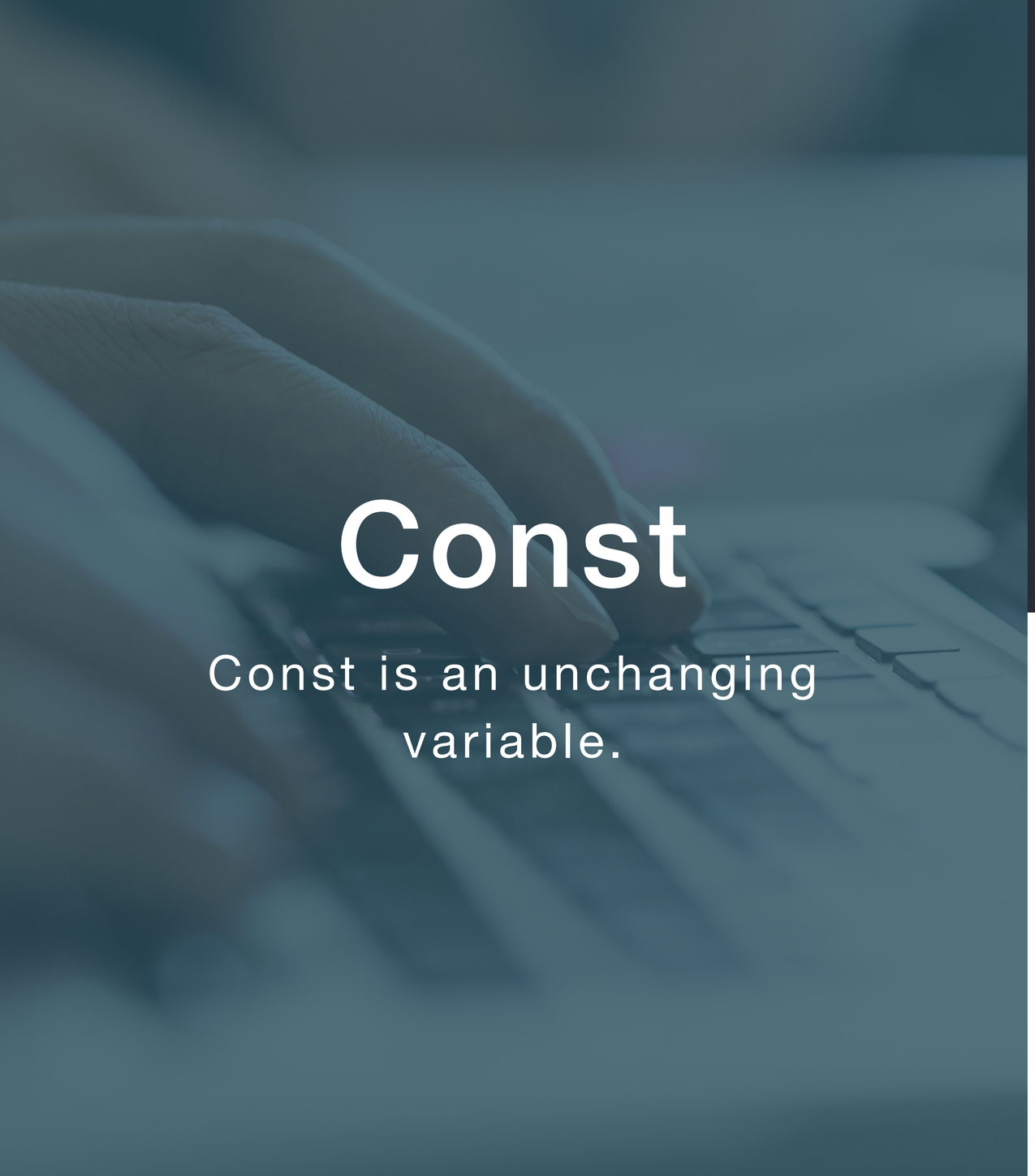
```
// “var” tolerates redeclarations
var user1 = "Pete";
var user1 = "John"; // this "var" does nothing (already declared)
// ...it doesn't trigger an error
console.log(user1); // John

let user2;
let user2; // SyntaxError: 'user' has already been declared
```

var-vs-let

CRUD App

1. Find øvelsen fra Intro til Programmering - CRUD App
2. Genbesøg din kode.
3. Overvej hvordan du anvender variabler.
4. Hvilke variabeltyper anvender du? Hvorfor?





Const

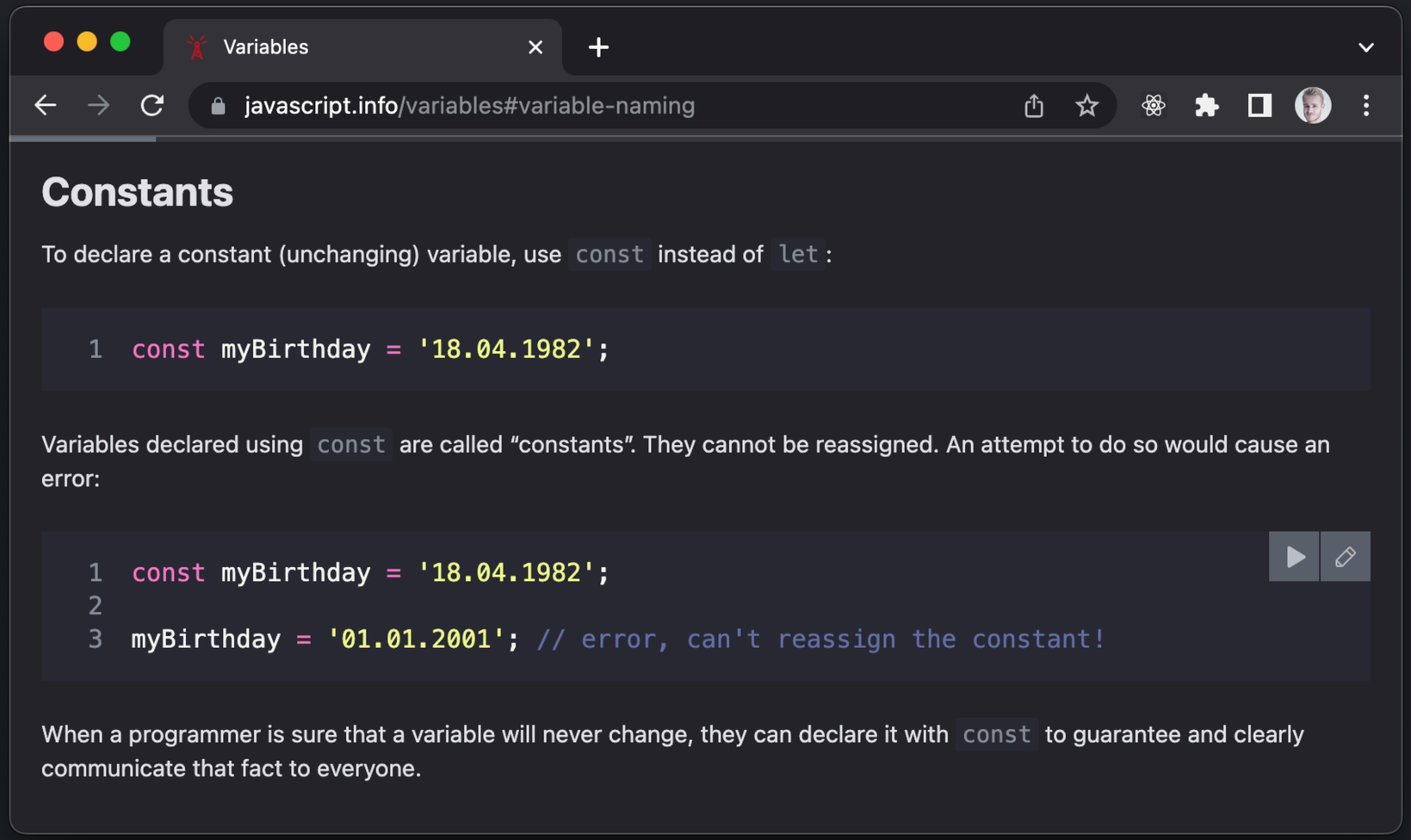
Const is an unchanging variable.

```
const myBirthday = "12-03-1990";
myBirthday = "12-03-1989";
// Uncaught TypeError: can't reassign the constant!
```

`const` cannot be reassigned.

If you try to, an `error` will be thrown.

And the doc says...



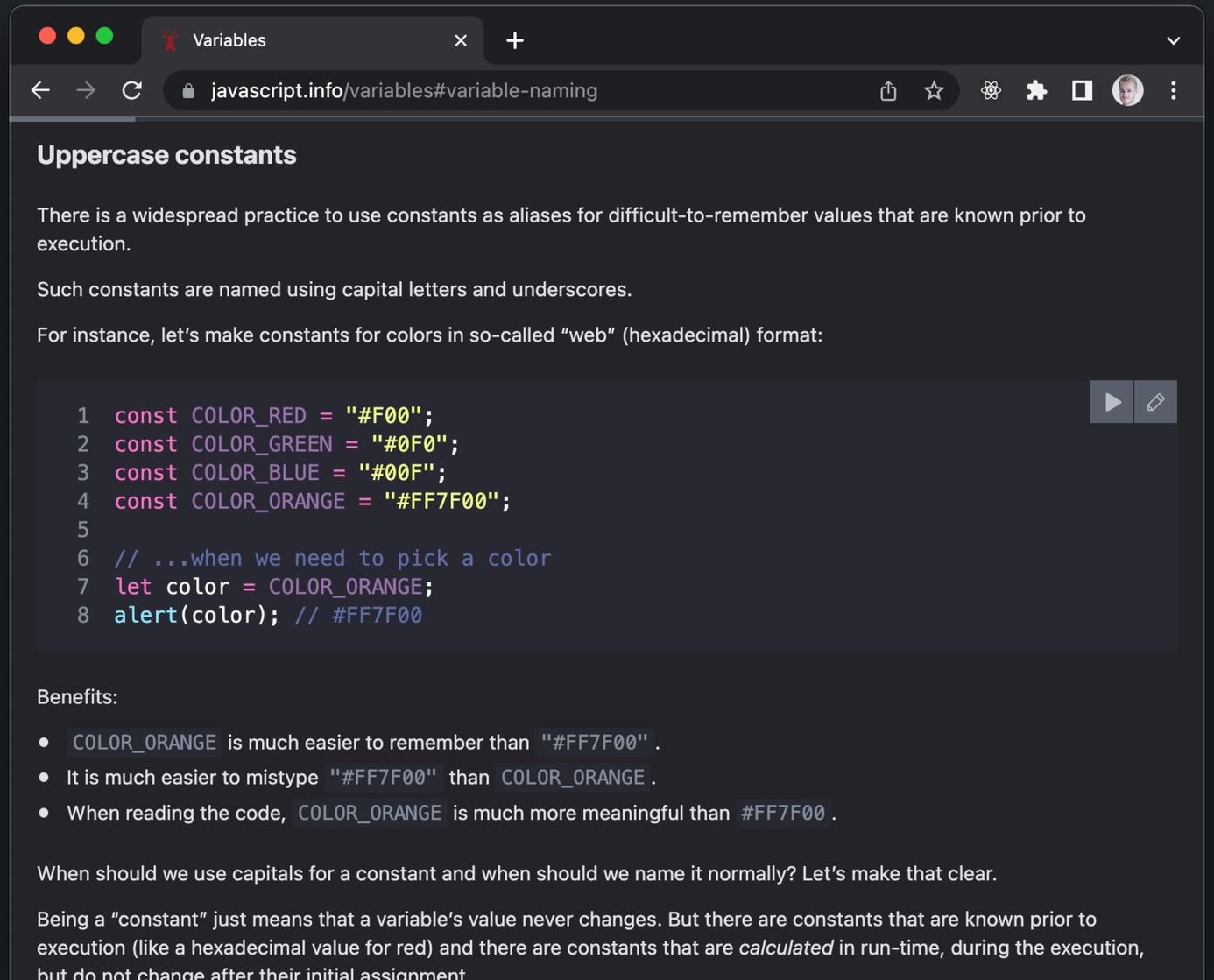
Const can’t be reassigned

```
const myBirthday = "12-03-1990";
myBirthday = "12-03-1989"; // Uncaught TypeError: can't reassign the constant!

const person = {
    name: "Kasper",
    mail: "kato@eaaa.dk",
    age: 32
};

person.age = 33; // no error

person = {
    name: "Rasmus",
    mail: "race@eaaa.dk",
    age: 31
}; // Uncaught TypeError: can't reassign the constant!
```

Const

Hvis du igen tænker på Clicker-Spillet, havde det givet mening at anvende const? Har du anvendt const?

Uppercase?



Uppercase constants

There is a widespread practice to use constants as aliases for difficult-to-remember values that are known prior to execution.

Such constants are named using capital letters and underscores.

For instance, let's make constants for colors in so-called "web" (hexadecimal) format:

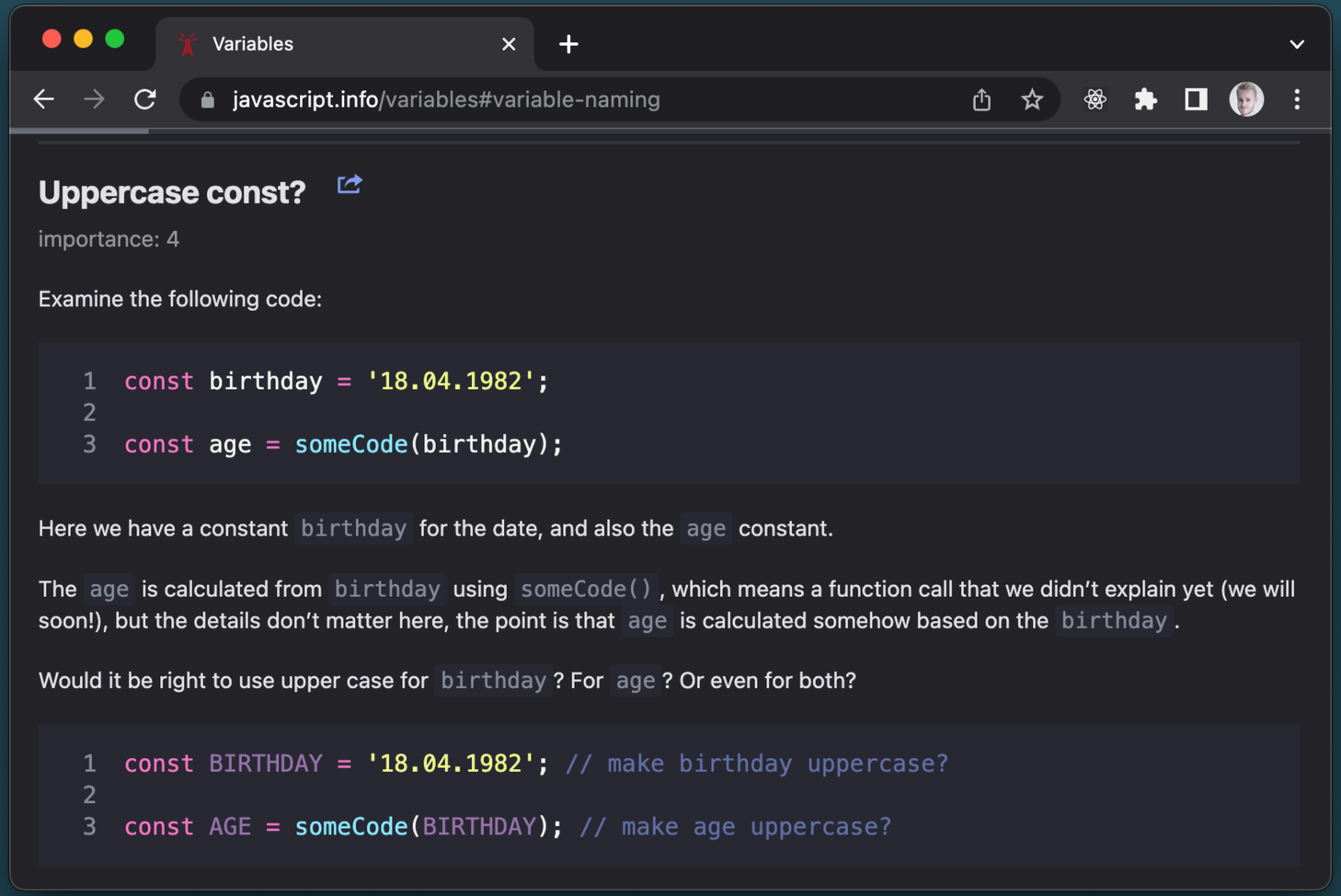
```
const COLOR_RED = "#F00";
const COLOR_GREEN = "#0F0";
const COLOR_BLUE = "#00F";
const COLOR_ORANGE = "#FF7F00";

// ...when we need to pick a color
let color = COLOR_ORANGE;
alert(color); // #FF7F00
```

Benefits:

- `COLOR_ORANGE` is much easier to remember than `"#FF7F00"`.
- It is much easier to mistype `"#FF7F00"` than `COLOR_ORANGE`.
- When reading the code, `COLOR_ORANGE` is much more meaningful than `#FF7F00`.

When should we use capitals for a constant and when should we name it normally? Let's make that clear.

Being a "constant" just means that a variable's value never changes. But there are constants that are known prior to execution (like a hexadecimal value for red) and there are constants that are *calculated* in run-time, during the execution, but do not change after their initial assignment.



https://javascript.info/variables

Use `let` & `const` instead of `var`

https://javascript.info/variables
https://javascript.info/var

But which one? 🤷

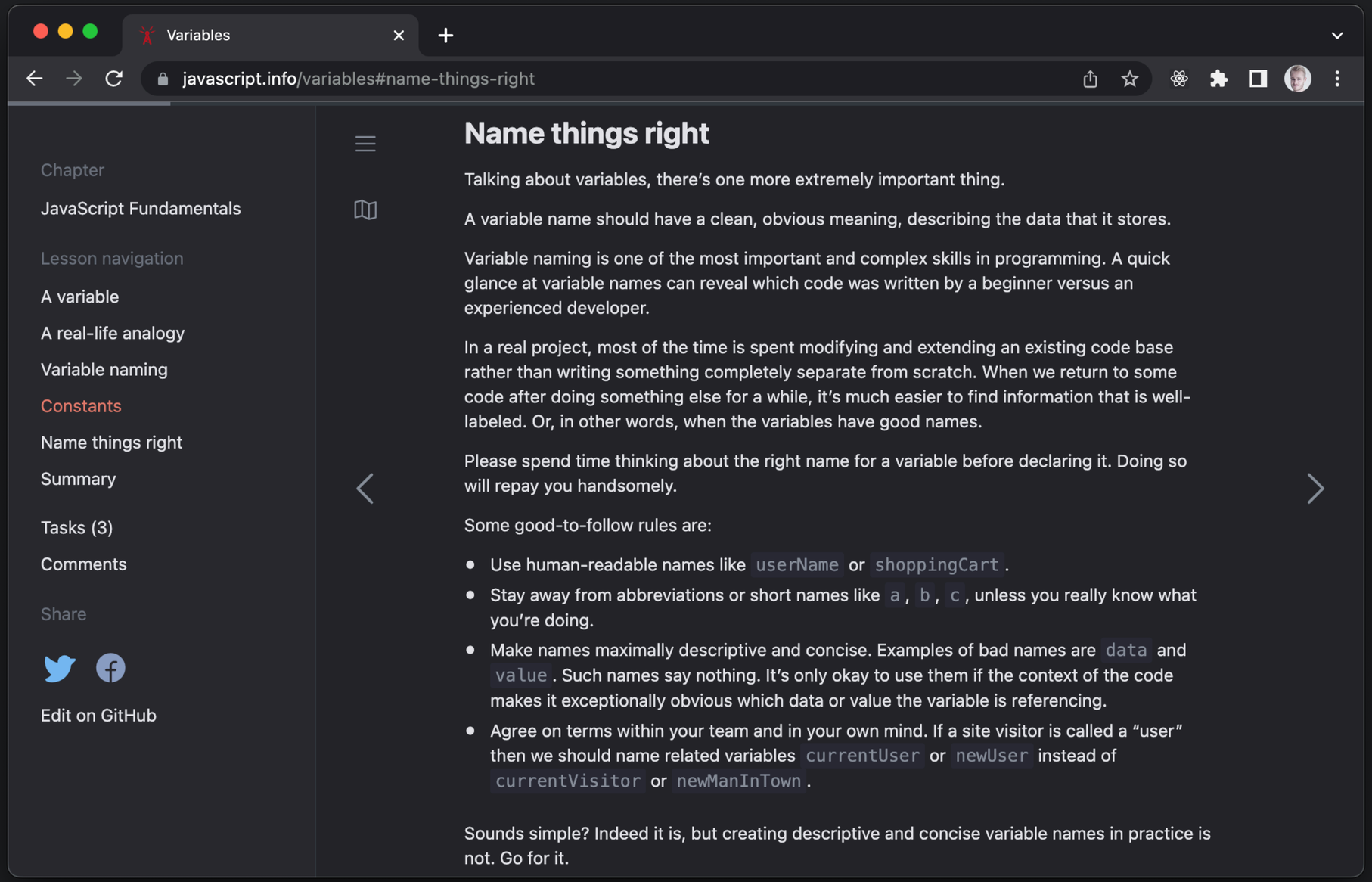
https://javascript.info/variables
https://javascript.info/var

Start with `const`

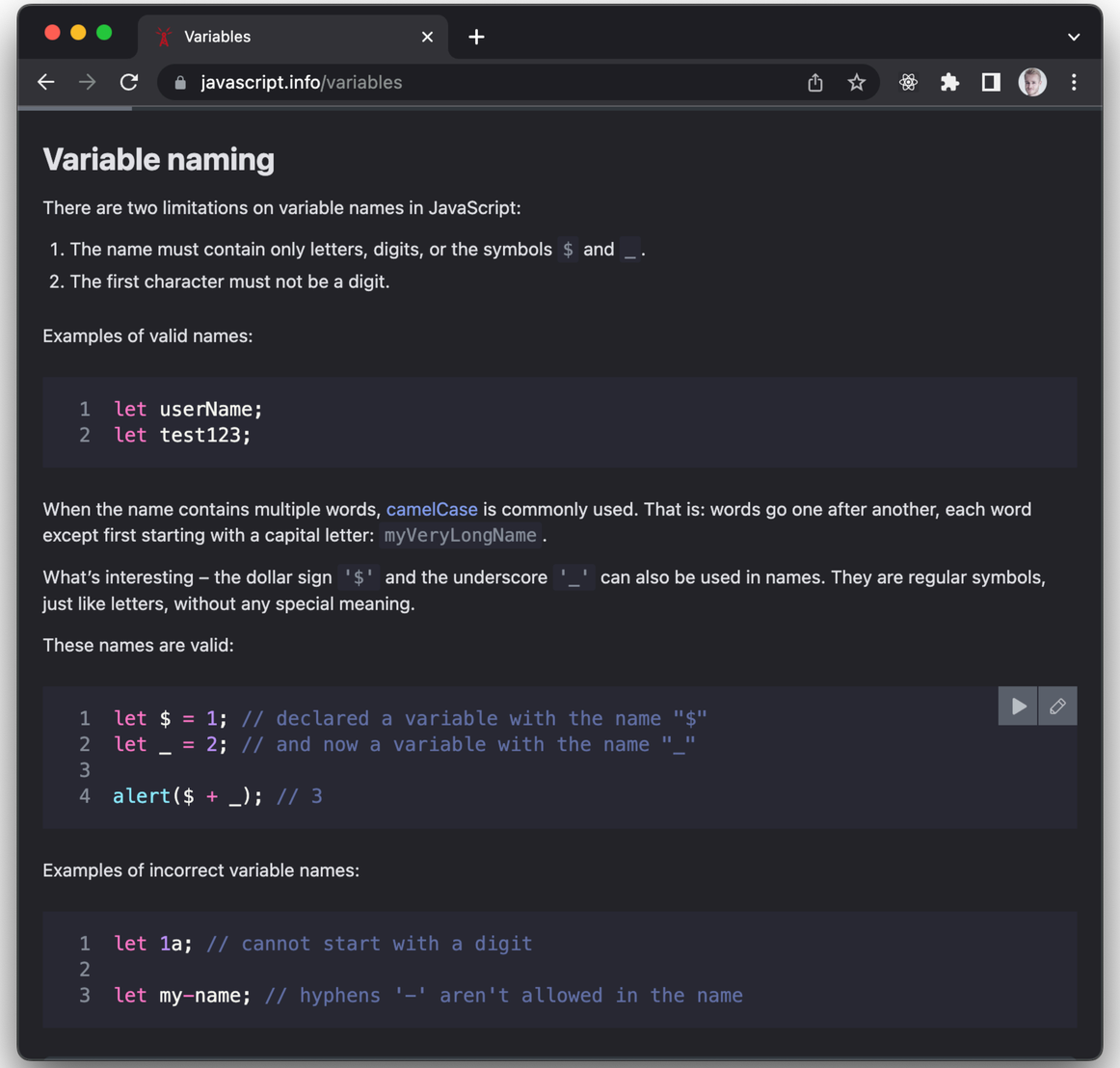
If needed, change to `let`

Never use `var`!

https://javascript.info/variables
https://javascript.info/var



https://javascript.info/variables#name-things-right



https://javascript.info/variables#variable-naming

Name things right (or wrong?)

1. Create a variable with the name of our planet. How would you name such a variable?
2. Create a variable to store the name of a current visitor to a website. How would you name that variable?

https://javascript.info/variables

Global Variables

Variables outside a function (and scopes) are global variables

JavaScript.info/function-basics#local-variables

Scopes:

- Local Variable
- Global Variable

Local variables

A variable declared inside a function is only visible inside that function.

For example:

```
1  function showMessage() {
2    let message = "Hello, I'm JavaScript!"; // local variable
3
4    alert( message );
5  }
6
7  showMessage(); // Hello, I'm JavaScript!
8
9  alert( message ); // <-- Error! The variable is local to the function
```

Outer variables

A function can access an outer variable as well, for example:

```
1  let userName = 'John';
2
3  function showMessage() {
4    let message = 'Hello, ' + userName;
5    alert(message);
6  }
7
8  showMessage(); // Hello, John
```

Global variables

Variables declared outside of any function, such as the outer `userName` in the code above, are called *global*.

Global variables are visible from any function (unless shadowed by locals).

It's a good practice to minimize the use of global variables. Modern code has few or no globals. Most variables reside in their functions. Sometimes though, they can be useful to store project-level data.

https://javascript.info/function-basics#outer-variables

Global & local Variables

Eksempler fra Clicker-Spillet?

Hvorfor er det vigtigt?

Data Types

In JavaScript there are two main data types:

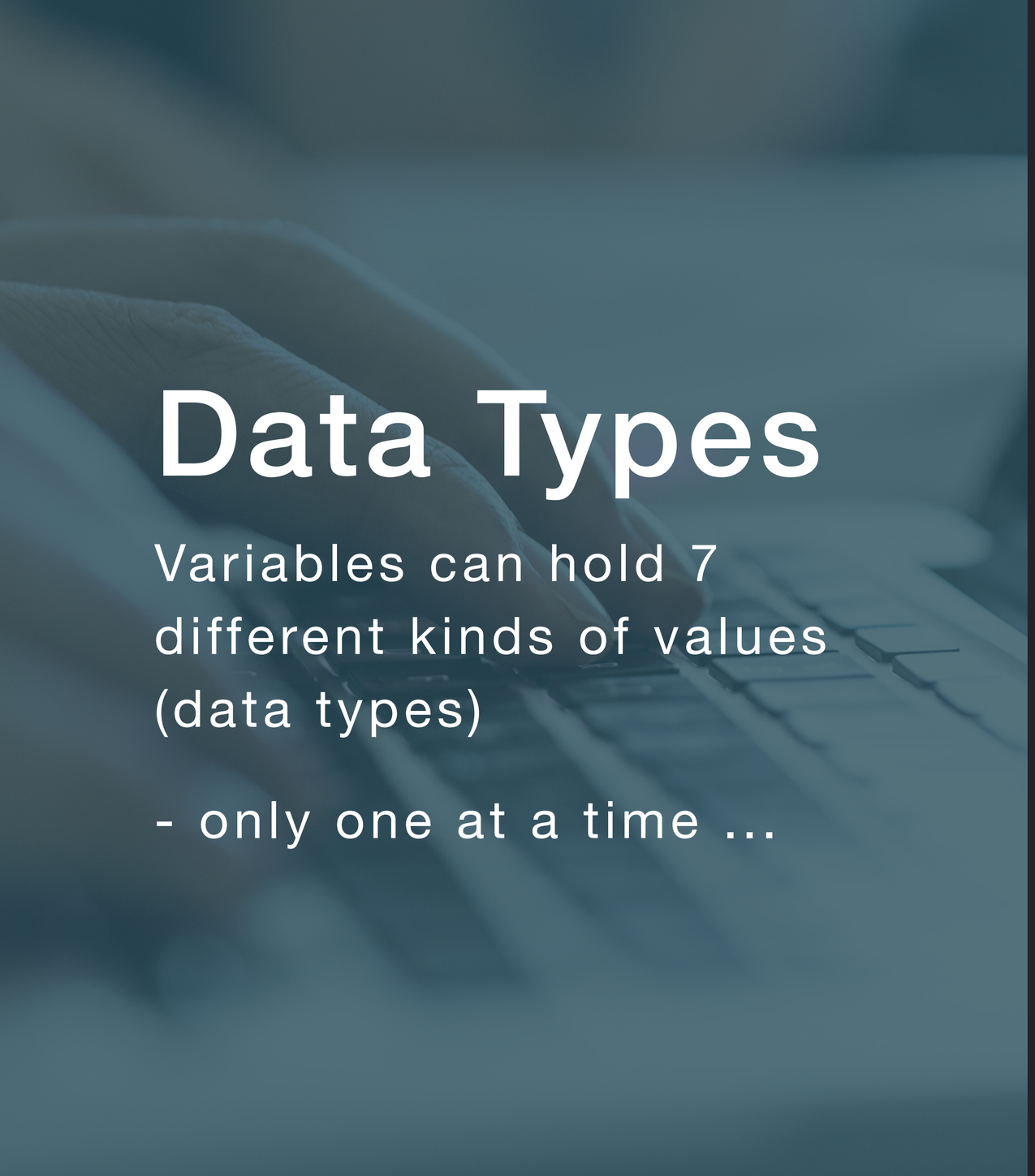
- **Primitive values** like strings, numbers and booleans.
- **Objects** with properties.

```
let str = "Hello";
let str2 = 'Single quotes are ok too';
let phrase = `can embed another ${str}`;
```

```
let n = 123;
n = 12.345;
```

```
let user = new Object(); // "object cons
let user = {};  // "object literal" synt
```

In JavaScript, a value always has a certain type like a string, number, boolean, object, array, etc.



Data Types

Variables can hold 7 different kinds of values (data types)

- only one at a time ...

Boolean

Number

String

Object

Null

Undefined

Symbol

Datatypes - when to use which one?

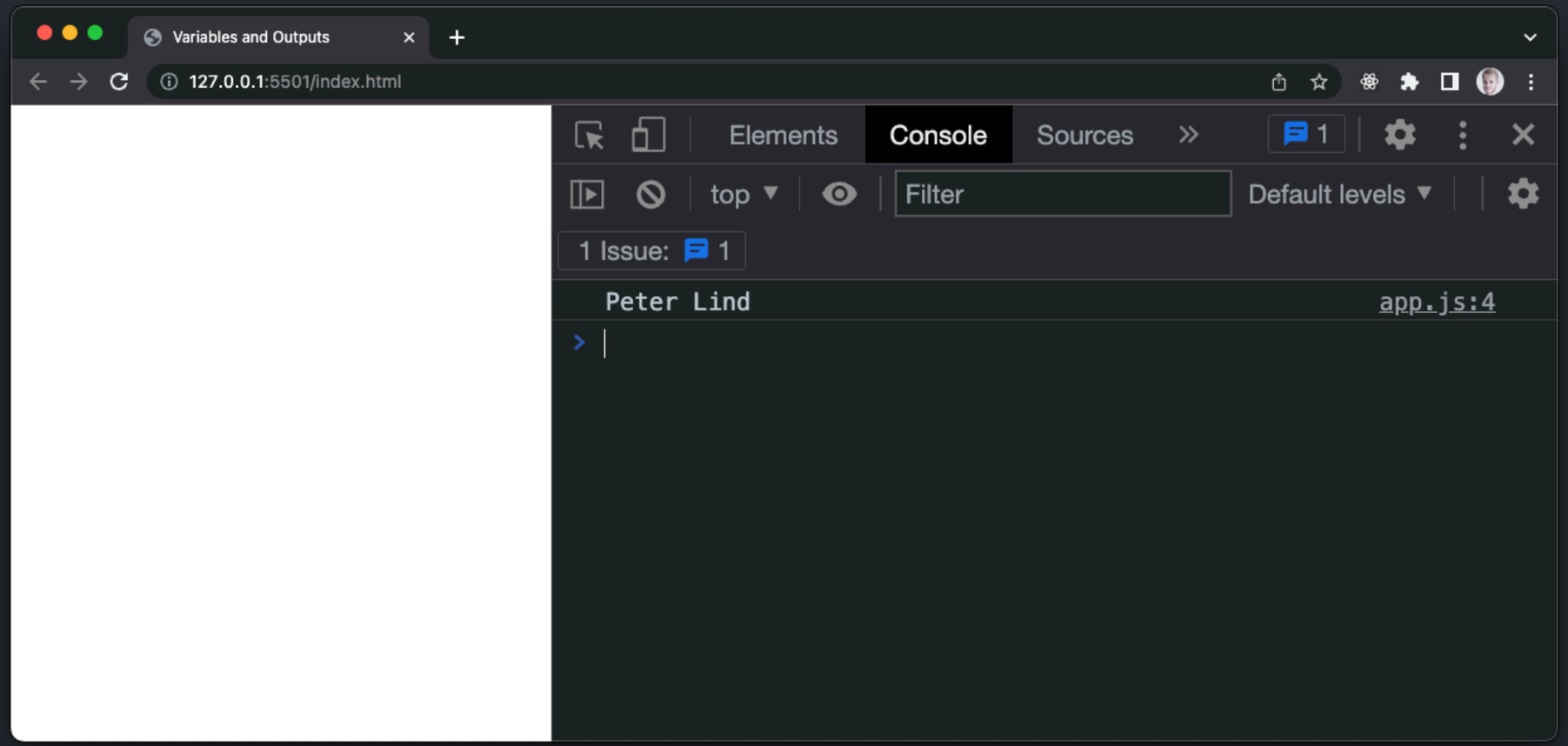
Boolean	Values that are true or false – conditions
Number	Numbers! Points, counting, **math-operations**, measuring …
String	Text! Input-fields – HTML and CSS-attributes … URLs (Phone-number)
Object	Everything (Arrays, Functions) Combined/collected data (of the other types)
Null	No datatype yet – no value!
Undefined	Something that hasn't defined before – don't set things to undefined!
Symbol	Something complicated

Datatyper

Hvilke datatyper har du anvendt i Clicker-spillet?

Hav Console åben - altid!

Så... hvor kan se datatyperne?



Datatyper

1. Prøv disse og test med `console.log`
2. Hvordan ser vi værdien?
3. Hvordan ser vi typen?
4. Hvordan definerer du, hvilken type, du vil gemme i en variabel?

```
const bool = true;
const num = 41;
const str = "Peter";
const obj = {
  cats: 2,
  cars: 1
};
const nothing = null;
let undf;
const symbol = Symbol("symbol");
```

https://javascript.info/variables

Strings

In JavaScript textual data is stored as strings.

Strings are defined with either ‘single quotes’, “double quotes” or `backticks`.

```
let single = 'single-quoted';

let double = "double-quoted";

let backticks = `backticks`;
```

```
let fullName = "Peter Lind";
let age = 39;

let message = fullName + " is " + age + " years old.";
console.log(message); // Peter Lind is 39 years old.
```

Concatenation

We can combine two or more variables in one string (variable).

```
let fullName = "Peter Lind";
let age = 39;

// single
console.log(fullName + ' is ' + age + ' years old.');
// double
console.log(fullName + " is " + age + " years old.");
// backticks
console.log(`${fullName} is ${age} years old.`);
```

Concatenation

Strings are defined with either ‘single quotes’, “double quotes” or `backticks`.

String quotes

1. Test the code to the right.
2. What is the output of the script to the right?

```
"use strict";

let fullName = "Alicia Keys";

console.log(`hello ${1}`); // ?

console.log(`hello ${"name"}`); // ?

console.log(`hello ${fullName}`); // ?
```

Data Type Conversion

In JavaScript, we never specify the type of a variable (String, Number, Boolean, Object, etc.).

We set the value and use the value as the type we expect it to be.

```
const firstName = "Rasmus";
const lastName = "Cederdorff";

let age = 32;

let isSeniorLecture = true;

const obj = {
    cats: 0,
    cars: 1
};

const kids = ["Alicia", "Ida"];
```

It happens when we do:

- Concatenation (+)
- Comparisons (==, !=, <, >, <=, >=)
- Calculations (-, *, /, %)

Automatic conversion

So often JavaScript has to do automatic conversion.

Everything in HTML is strings, so when writing or reading numbers, they have to be converted.

Most of the time this happens automatically.

“Semi-automatic” conversion

`String(`*`value`*`)` converts a value to a string, usually by calling *value*`.toString()`

`Number(`*`value`*`)` converts a value to a number, usually by calling *value*`.valueOf()`

`Boolean(`*`value`*`)` converts a value to a boolean, by unknown magic means!

Note:
Boolean(0) is false, but
Boolean("0") is true

It is always possible to convert something into a string –
but it might not be possible to convert everything into a number!

Numeric conversion

1. How do you convert to Number?

```
let string = "1234";

console.log(string);

// number??
```

String conversion

1. How do you convert to String?

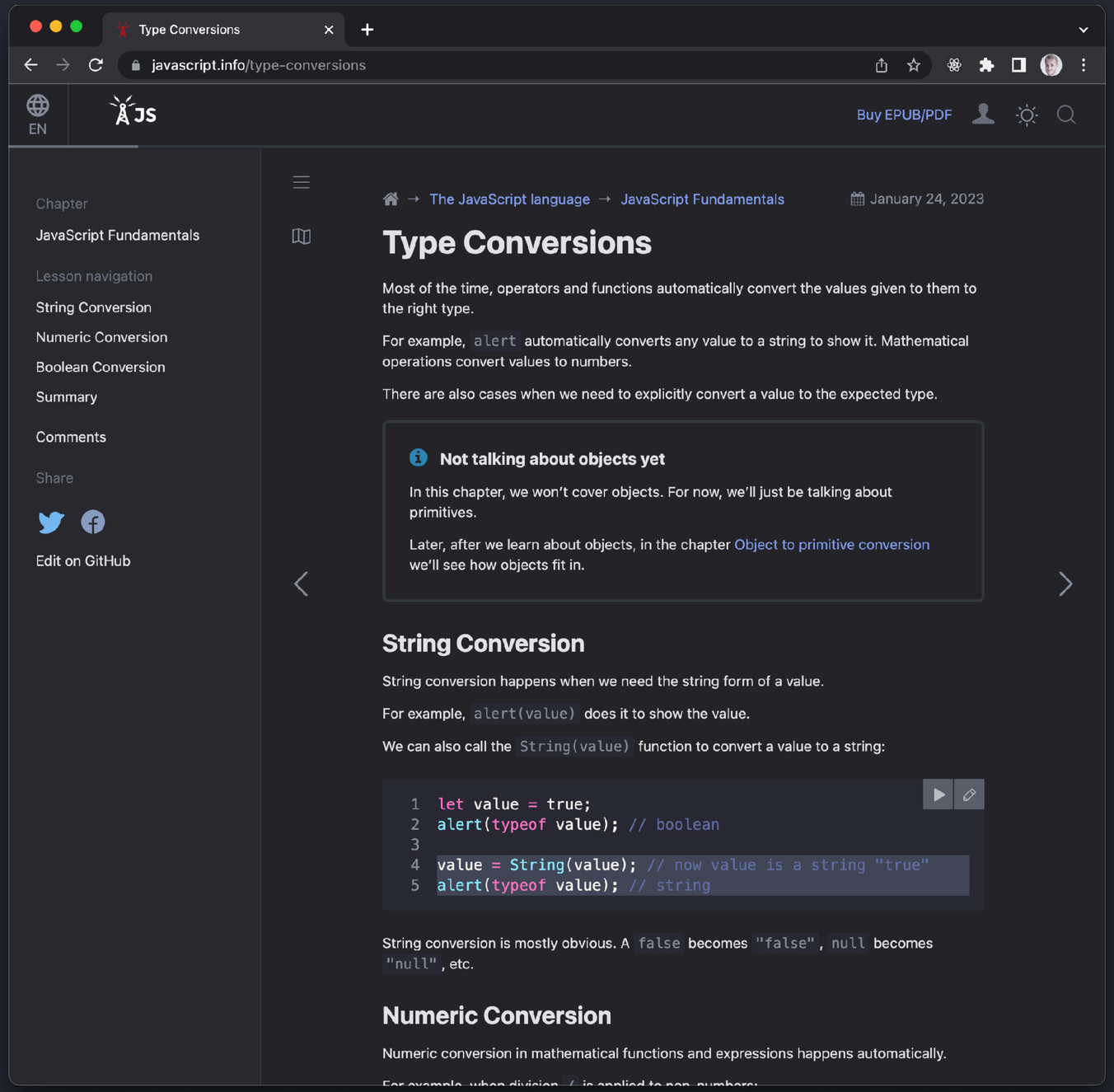
```
let number = 1234;

console.log(number);

// string??
```

Numeric conversion 2

1. Try yourself!
2. What is the output in the console?

```
console.log(Number("   123   ")); // ?
console.log(Number("123z")); // ?
console.log(Number(true)); // ?
console.log(Number(false)); // ?
```

https://javascript.info/strict-mode



Read, read, read, The docs 🤷

Manual conversion

You can ommit the prefix `Number.` if you want

`Number.parseInt( string, radix )`

Converts a string **to** a **number**, using the specified numerical system (MDN)

`Number.prototype.toString( radix )`

Converts a number **to** a **string**, in the specified numerical system (MDN)

There's also: parseFloat, toExponential, toFixed, toPrecision for other kinds of conversion

```
const n1 = 1;
const n2 = 2;

const s1 = "1";
const s2 = "2";

n1 + n2;
s1 + s2;

n1 + s2;
s1 + n2;
```

Concatenation looks like addition

- What will be the result of these “calculations”?
- Test them in the console!

Automatic data type conversion with concatenation

String + string is a **concatenation**

Number + number is a **calculation**

But if the types are different, one gets converted to the other!

The challenge is to figure out which one gets converted to what ...

```
"" + 1 + 0;
"" - 1 + 0;
true + false;
6 / "3";
"2" * "3";
4 + 5 + "px";
"$" + 4 + 5;
"4" - 2;
"4px" - 2;
"   -9   " + 5;
"   -9   " - 5;
null + 1;
undefined + 1;
" \t \n" - 2;
```

Type Conversion

1. Test these expressions.
2. First: Try to guess the result, then check your guess in the console!
3. Explain to each other why your guess was correct/ wrong.

https://javascript.info/type-conversions


Code
Every
Day