Introduction to *Variables*

CODE CHRYSALIS コードクリサリス

By the end of this lesson, you should be able to...

1. Be able to declare variables and assign values
2. Know when to use:
   - `var`
   - `let`
   - `const`

CODE CHRYSALIS コードクリサリス

Remember these data types?

name	examples
number	`1`, `-5`, `1.0001`
string	`"Hello world!"`, `'I love coding!'`
boolean	`true`, `false`

CODE CHRYSALIS コードクリサリス

Remember these data types?

name	examples
number	`1`, `-5`, `1.0001`
string	`"Hello world!"`, `'I love coding!'`
boolean	`true`, `false`

Remember these data types?

name	examples
number	`1`, `-5`, `1.0001`
string	`"Hello world!"`, `'I love coding!'`
boolean	`true`, `false`

What do we do if we want to refer to that data again later?

Use variables!

CODE CHRYSALIS コード クリサリス

Use variables!

Use variables!

Variables are containers that have names. They can be declared and assigned to a value or expression.

Use variables!

Variables are containers that have names. They can be declared and assigned to a value or expression.

This allows you to refer to it in other places of your code.

Use variables!

Variables are containers that have names. They can be declared and assigned to a value or expression.

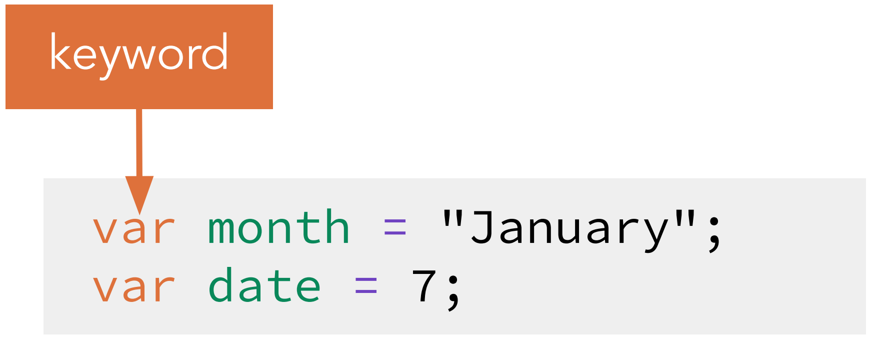
This allows you to refer to it in other places of your code.

```
var month = "January";
var date = 7;
```

Use variables!

Variables are containers that have names. They can be declared and assigned to a value or expression.

This allows you to refer to it in other places of your code.

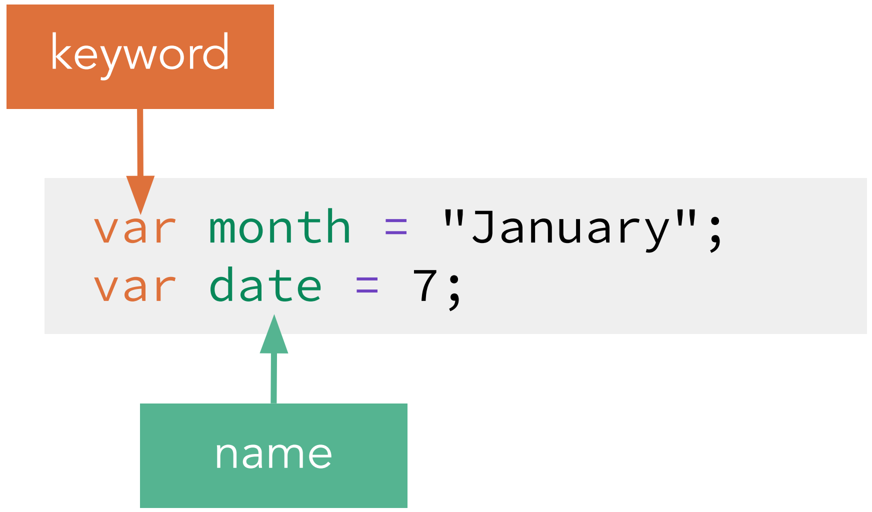


```
var month = "January";
var date = 7;
```

Use variables!

Variables are <u>containers</u> that have names. They can be declared and assigned to a value or expression.

This allows you to refer to it in other places of your code.

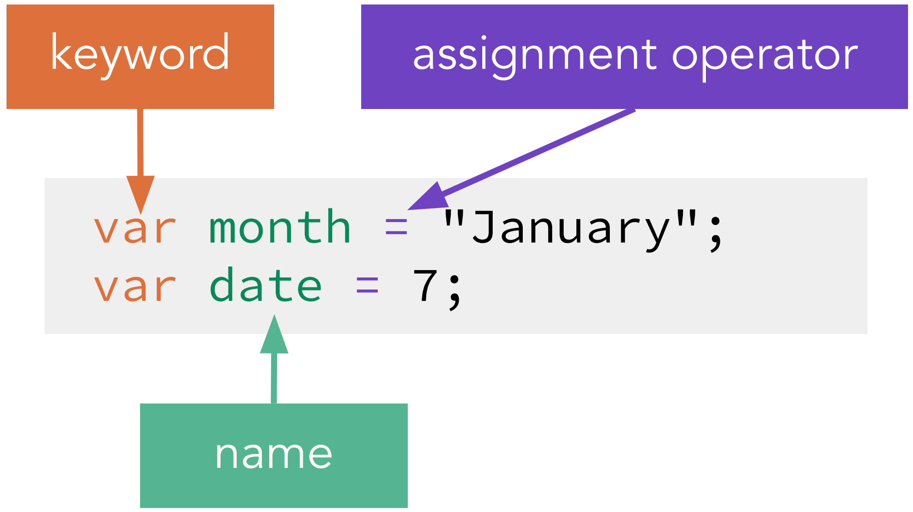


```
var month = "January";
var date = 7;
```

Use variables!

Variables are containers that have names. They can be declared and assigned to a value or expression.

This allows you to refer to it in other places of your code.



```
var month = "January";
var date = 7;
```

Vocabulary

Assignment:

Copying the value of the right side to the left.

```
var meaningOfLife = (6 * 9).toString(13);
```

CODE CHRYSALIS コードクリサリス

var, let, const?

```
var name = "Alice";

let myLocation = "B2";

const anotherName = "Bob";

name = "Carol";

myLocation = "home";

anotherName = "Dave";
```

var, let, const?

```
var name = "Alice";

let myLocation = "B2";

const anotherName = "Bob";

name = "Carol";

myLocation = "home";

anotherName = "Dave";
```

var, let, const?

```
var name = "Alice";

let myLocation = "B2";

const anotherName = "Bob";

name = "Carol";

myLocation = "home";

anotherName = "Dave";
```

var CAN be *reassigned*.

var, let, const?

```
var name = "Alice";

let myLocation = "B2";

const anotherName = "Bob";

name = "Carol";

myLocation = "home";

anotherName = "Dave";
```

var CAN be *reassigned.*

let CAN also be *reassigned.*

var, let, const?

```
var name = "Alice";

let myLocation = "B2";

const anotherName = "Bob";

name = "Carol";

myLocation = "home";

anotherName = "Dave";
```

var CAN be *reassigned.*

let CAN also be *reassigned.*

const CANNOT be *reassigned.*

var, let, const?

```
var name = "Alice";

let myLocation = "B2";

const anotherName = "Bob";

name = "Carol";

myLocation = "home";

anotherName = "Dave";
```

`var` CAN be *reassigned.*

`let` CAN also be *reassigned.*

`const` CANNOT be *reassigned.*

Why use `let` instead of `var`?

var, let, const?

```
var name = "Alice";

let myLocation = "B2";

const anotherName = "Bob";

name = "Carol";

myLocation = "home";

anotherName = "Dave";
```

`var` CAN be *reassigned*.

`let` CAN also be *reassigned*.

`const` CANNOT be *reassigned*.

Why use `let` instead of `var`?

`let` and `const` are newer syntax and have better error handling, so use of `let` and `const` is preferred.

var, let, const?

```
var name = "Alice";

const anotherName = "Bob";

let myLocation = "B2";

myLocation = "home"

name = "Carol";

anotherName = "Dave";
```

This will *throw* a TypeError: Assignment to constant variable.

var CAN be *re* *signed.*

let e re *signed.*

igned.

r **var**?

re new syntax

etter rror handling, so

s le and **const** is preferred.

CODE CHRYSALIS コードクリサリス

Arithmetic Operators (continued)

Variables can be *manipulated* using operators. For example:

```
let myNumber = 0;
const shouldItGetBigger = true;
if (shouldItGetBigger) {
  myNumber = myNumber + 1;
}
```

Arithmetic Operators (continued)

There is another operator, called the *increment operator*, that has a similar effect.

```
let myNumber = 0;
const shouldItGetBigger = true;
if (shouldItGetBigger) {
  myNumber = myNumber + 1;
}
```

```
let myNumber = 0;
const shouldItGetBigger = true;
if (shouldItGetBigger) {
 myNumber++;
}
```

Arithmetic Operators (continued)

++	Increment	3++ => *4*
--	Decrement	9-- => *8*

CODE CHRYSALIS コードクリサリス

REVIEW TIME

REVIEW TIME

REVIEW TIME

- How do you declare a variable?

REVIEW TIME

- How do you declare a variable?
- What types of data can you assign to a variable?

REVIEW TIME

- How do you declare a variable?
- What types of data can you assign to a variable?
- What *is* a variable?

CODE CHRYSALIS コードクリサリス

REVIEW TIME

- How do you declare a variable?
- What types of data can you assign to a variable?
- What *is* a variable?
- What are the differences between the keywords `var`, `let`, and `const`?

CODE CHRYSALIS コードクリサリス

Activity

In *Introduction to Variables*, work with the person next to you on the “Paired Activity” section.

Continue with the exercises after you are done.

CODE CHRYSALIS コード クリサリス